On the use of Myo as an extension for digital musical instruments

Mattia Paterna
Sound and Music Computing
A.C. Meyers Vænge 15
København, Denmark
mpater15@student.aau.dk

Paolo A. Mesiano
Sound and Music Computing
A.C. Meyers Vænge 15
København, Denmark
pmesia15@student.aau.dk

ABSTRACT

A system is presented that lets the user to affect in real-time a granular synthesis engine built in Max/MSP. To do so, the Myo™ gesture control armband is used, that provides a series of pre-defined gestures and spatial values. That data is then processed within a C++ script built using Thalmic Labs' proprietary SDK and sent to Max/MSP via UDP protocol. Such a system aims to improve the possibilities of expressiveness for musicians, especially in improvisations and group performances.

Author Keywords

Myo™ armband, gesture control, Max, granular synthesis

1. INTRODUCTION

In an era of technological development, computers have taken a fundamental role in the musical field. Such a large computational power allows for whatever comes in our mind: we could even start with a blank sheet in the design of musical instruments with endless potential [4]. New programming environments, such as Max/MSP, allows for more extensive experimentations but often lack of expressivity that the musician could aim [3]. Indeed, new problems definition and difficulties arise when performing using laptops and their solely controls, e.g. keyboard and mouse. Therefore, novel methods of controls have to be sought to improve the possibilities of expressiveness and provide the musician with some *human-feel* control gestures.

In this paper, a system is presented which lets the user to affect in real-time some parameters within a granular synthesis engine. Doing so, the Myo™ gesture control armband is used. Section 2 presents the implementation strategies, as well as some technical specifications. Section 3 gives an overall evaluation of the system based on a selection of dedicated framework. Finally, section 4 provides a general conclusion.

2. IMPLEMENTATION

2.1 Myo™ gesture control armband

The Myo gesture control armband, by Thalmic Labs[1], is a device that classifies different user gestures by measuring muscle activity, together with the detection of the user arm motion and position in space to control different applications through hand gestures.

The Myo armband is provided with eight proprietary medical grade stainless steel EMG sensors that measure the electric potential of the user forearm muscles. Furthermore, the armband is provided with a nine-axis IMU containing a three-axis gyroscope, a three-axis accelerometer and a three-axis magnetometer that allow the measurement of position and motion of the arm in space.

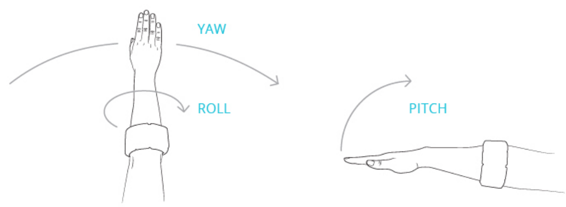

Figure 1: Orientation controls provided by Myo™ armband. Courtesy: `developerblog.myo.com`

The armband provides two kind of data. Firstly, the EMG sensors provide gestural data by classifying the muscle activity into predetermined hand poses (*fist, fingerspread, wave in, wave out, rest*). Moreover, the IMU provide spatial data, further divided into orientation data and an acceleration vector giving information about the motion of the arm. Orientation data are separated in the three angles roll, pitch and yaw as shown in 1.

All that data can be used for a wide range of purposes by means of the Thalmic Labs SDK which allows to develop C++ applications with personalised armband behaviour. Furthermore, the Myo armband gives the possibility to design a control interface made of gestures that are very close to standard gestures commonly used for touch-based interfaces[1], such as analog knob rotation, sliders, tapping, etc. It is indeed possible to transfer the physical interaction of these controls to mid-air gestures by exploiting as much as possible their existent intuitiveness. Hence, we think that such many features make it a valid aid to improve musical meaningless.

2.2 The script

We embedded the Myo SDK into a C++ script which receives both spatial and gestural data from the armband and sends specific commands to a Cycling 74's Max/MSP[2] patch via UDP packets. The overall framework of the system is shown in fig 2.

[1] `https://www.myo.com`

[2] `https://cycling74.com/products/max`

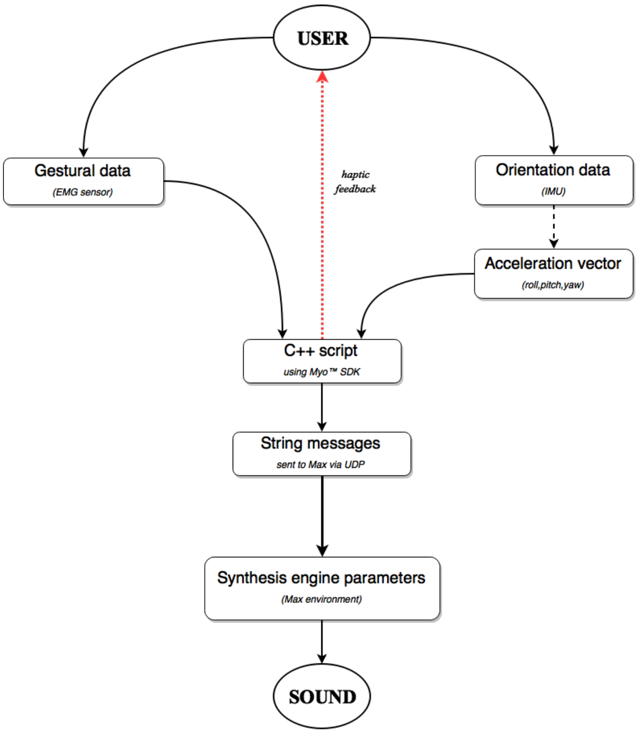

Figure 2: Overall framework of the system.

The SDK comes with a set of virtual functions that govern the armband behavior and that can be overloaded by the user. In our case, we set up the Myo behavior onto two layers of control. We call them *top* and *bottom* layer. The former is an always-active control level that receives pitch and yaw angle data continuously and uses them to control volume and density respectively. The latter is accessed through the double tap and gives access either to triggers (wave-in, wave out gestures) or held gestures combined with roll angle data. These are then used to control cross-fade, speed, and on/off random grain duration and amplitude. Table 1 shows all the gestures involved, including the corresponding affected sound parameter, the control type and the scaled range on the Max/MSP side.

When opening the application, the top layer starts sending spatial data to Max/MSP. Once the bottom layer is enabled via double tap, the top layer is excluded. The bottom layer keeps its activity until another double tap gesture is performed. Spatial and gestural data is continuously measured by the device and is read at frequency defined by the user within the script. We chose an update time of $50ms$ that aims to provide the user with a real-time change feeling. Nevertheless, this feature can be easily changed depending on the performer needs of motion tracking resolution, which can be set up to a maximum of 1ms refresh time according to the Myo™ specifics.

Every time data is updated, the script performs different tasks:

- filter spatial data to a range that fits the performer comfort and scale them to integer values from 0 to 99. This makes possible to restrict the parameter space to the solid angle that the performer arm can sweep;
- provide haptic feedback, a *short-time* vibration, every time a different gesture is performed or when a spatial range limit is reached to let the user experience the boundaries within it could affects the synthesis engine parameters;
- translate spatial and gestural data to proper string messages, create UDP packets and send them through a UDP network.

2.3 Max/MSP and the user interface

The granular synthesis engine has been built in Max/MSP. We can think of granular synthesis as thousands of grains over time creating an animated sonic atmosphere. The main building block, the *grain*, is a brief microacoustic event shaped by an amplitude envelope[7].

In our system, the asynchronous granular synthesis is achieved combining two *phasor* objects. The former reads a selected portion of the sample, while the latter is reading the enveloped grain. They both can vary their speed, yielding both to a pitch shifting effect and to a "freezing" effect. We decided to add some peculiar characteristics to the standard engine, such as some random controls over grain amplitude and panning, a *scattering* option that allows for flexible start/end sample reading points and, most important, two distinct reading buffers that allows for a granulation over two buffer at a time. Both the buffers have separated controls and come with a cross-fade command.

The whole system has then been put in a compact form, so to show only essential controls to the user. For further information about the interface, see 3.2.

3. EVALUATION

As said in [2], more and more emphasis is given to the evaluation of a new digital musical instrument. There may be many different perspectives which to view the effectiveness of such instruments with, as well as some concepts as *playability*, *enjoyment* and *design*[5]. This section is intended to provide a detailed evaluation of the system described herein, particularly according to different framework for the evaluation of interactive digital musical instruments. Emphasis is given to the *parameter space*, to the design interface and to the *mapping algorithm*, following its definition in [6].

3.1 The parameter space

The Myo™ armband can track the position of the user arm within a sphere centered on the user body, since both pitch and yaw angles have a range of 2π. We can therefore define a parameter space as the portion of this sphere that can be swept by the performer arm: it is a square-based cone with vertex on the performer body, with axis oriented in front of the body and with base sizes that depends on how much the user can move its arm horizontally (yaw) and vertically (pitch). The parameter space is thus the set of all possible pairs of integers and scaled values within and these pairs are then sent to the Max/MSP patch to control *volume* and *density* respectively. In our case, the parameter space corresponds to the *top* layer, since no gesture is required to change these values, but only the natural arm movement.

Pitch and yaw ranges can of course be set not only depending on the user physical mobility but also on the way it decides to visualize the parameter space. Indeed, the user can decide to shrink yaw and pitch ranges to more than the limits imposed by physical mobility, if this provides a better visualization of the parameter space and a more comfortable experience of the device. It is also possible for the user to decide in which direction to increment the values (left-right or right-left for yaw, bottom-top or top-bottom for pitch). Hence the performer can design the parameter space to best fit its expressive needs and comfort, without affecting the musical output.

Briefly, the parameter space is depicted as virtual quadrilateral that spans all-over the space in front of the user

Detected gesture	Sound parameter	Type (Myo data)	Range
Fist	Speed	Continuous (roll)	0/1
Fingerspread	A/B crossfade	Continuous (roll)	0/1
Wave in	Random grain dur	Trigger	-
Wave out	Scattering	Trigger	-
-	Density	Continuous (yaw)	0/1
-	Grain delay	Continuous (pitch)	0/2 (sec)

Table 1: Gesture-to-parameter mapping

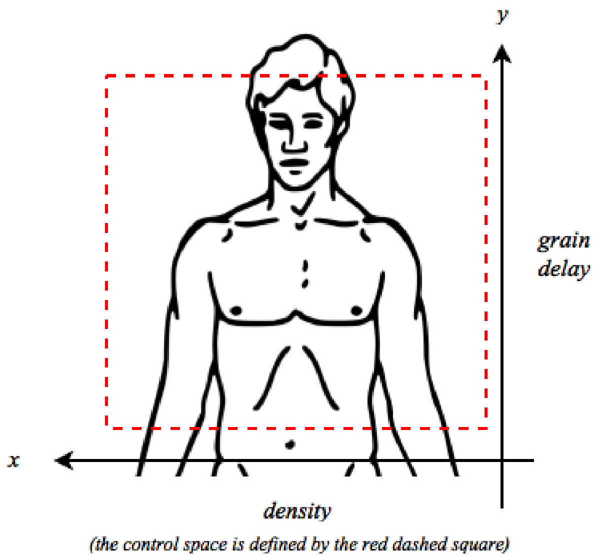

Figure 3: User gesture parameter space

upper-body. We have also decided not to include roll values inside the parameter space, so to obtain the closest representation to a Cartesian coordinate system. Figure 3 shows such defined parameter space.

Doing so, the user could experience and control its movements even with the sight. Moreover, that can be also considered a *focal* space during a performance. If the performer is able to concentrate the audience attention in that space through wide and easily recognizable gestures, it could yield to a definition of a new paradigm of interaction between the musician and the audience, especially in the field of electroacoustic music when the lack of a "classic" instrumentalist can somewhat damage the whole meaning of the execution.

3.2 The design interface

In designing the interface, some concepts of enjoyability and friendliness have been taken into consideration. We aimed to build an interface as simple as possible, with some dedicated visual feedback that make use of everyday metaphors (e.g. slide for continuous parameters, on/off toggles for triggers and some "swipe" controls in, for instance, the choice of the sample portion to be processed). The essence of Van Der Rohe's architectural philosophy here has found an application, avoiding useless controls and rub them out from a plethora of possible solutions, and even exciting at a first glance. Figure 4 depicts the digital interface built in Max/MSP.

3.3 The mapping algorithm

A mapping algorithm for real-time expressive controls fulfils the criteria for the building of a complete musical instrument and connects physical interfaces and synthesis tech-

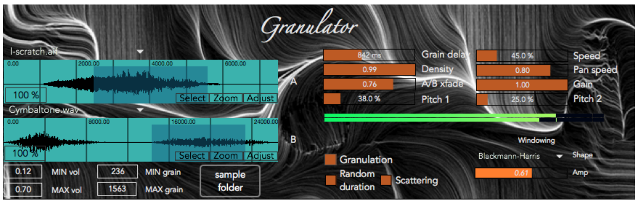

Figure 4: Max/MSP interface

nique[6]. The MITDS framework proposed by Overholt tries to consider all these concerns all at once. We could therefore apply the evaluation measure provided by [6] and draw some consideration.

First of all, we aim to close the gap between complex synthesis method (e.g. sample-based granular synthesis) and expressive human gestures and create a match between spatial/visual control and synthesis parameters. Many revision of the system have been made since the first draft has been depicted, that might improve a true expressive performance and widen the user range even to a non-professional figures.

We have not been interested in capture gestures that could be related with traditional instrumental techniques. Indeed, a new gestural grammar and new performance techniques are proposed. One of the possible drawbacks in defining a complete new grammar is the time needed for a user to learn and practice it. Even though this grammar lies in the predefined Myo™ gestures, we think that it can be accessible to a broad range thanks to the simplicity of the gesture, mostly based on everyday life actions. For instance, the *fist and rotation* replicates the turning of an analog knob and could be considered indeed easier than scrolling a digital knob using the mouse. We should also consider the learning curve for a user who is not experienced in affecting musical parameter with laptop controls, such as a mouse.

Since granular synthesis often produces complex timbres with mostly subtle changes, we have decided to match the gestures with the most prominent changes in the sound synthesis. It is evident that a listener could recognize with ease any modification in either density or volume rather than the level of polyphony, i.e. the number of voices within the synthesis. We have also decided not to map some synthesis parameters that do not have any correspondence with "analog" gestures, for instance the *grain length* parameter. That is because we have been stick to the choice of match the most analog and musical controls possible, so to offer digitalized control a human counterpart and a *action-sound* coupling [6] as clear as possible. Doing so, we are of course aware of the gap between "digital-born"[3] parameters and

[3] With *digital-born*, we intend all those parameters that come with the advent of the computer music and were not existing in traditional instruments before. A intonation parameter, for instance, is as much familiar both to a soprano and a electroacoustic musician who performs with sinusoidal oscillators. The carrier frequency in FM synthesis could be a good example of that family.

the way of providing controls over them in our system.

The grain length could therefore only be controlled in detail by hand, otherwise an input trigger allows for a random length option. The choice of a trigger can be also seen as a digital counterpart that reflects any electrical button task. Furthermore, no interaction choice to the user is given for sample selection. Nevertheless, the user could always choose which samples and whose portions to be processed.

More stress has been put in giving the user controls over some specific parameters added to the standard granular engine, as a continuous cross-fade between the two processed samples and trigger controls for random amplitudes and grains length, in order to enhance the identity and the uniqueness of our system. Moreover, some kind of correspondence between the nature of the parameters and that of the gestures is applied. Whereas the synthesis engine allows for continuous parameters (e.g. density), the user is provided with a large scale continuous gesture.

We also aimed extra-human effort into the interaction design. Granular synthesis could be in fact thought as an orchestra in which each of its musician is playing a determinate fragment, or grain. In designing such mapping, it has be taken into account the figure of the orchestral conductor, role deeply important both for musicians and audience as vehicle for musical expression. Usually, a conductor provides the orchestra with general wide gestures that vehicle dramatic changes in the piece structure (e.g. *crescendo*, *accelerando* and all information about phrasing), and other subtle ones that represent detailed information written down into the score (e.g. specific instrumental techniques). According to this methodology, the controls, as explained in 2.2 have been divided into coarse and fine-grained depending on the affected parameters, so that the mapping structure can be seen as composed of a top layer continuous controls and a bottom layer controls triggering some subtle parameters and providing as well continuous controls for the synthesis specific parameters.

To sum up, with this mapping choice the coupling between performers actions and produced sound could be clear enough both to the musician and to the audience. Finally, the choice of natural and wide gestures could help the audience in a better understanding of the musical meaning.

3.4 Further considerations

According to what stated in [5], we could investigate and explain some strong and weak points in our system. Since our system is mainly oriented to performers (and composers as well), we can assess a possible lack of usability for non musical-trained people. That is the result of a process where a musician think over an extension of its expressive potential and is itself both designer and user. Its choice will naturally reflect its way of facing musical troubles and possibilities, creating some constraints for all the other possible users.

Such a constraint can be recognised in our system. The parameter space could require way more time for a general user to get experienced with, albeit the user itself can take decision on it. On the other hand, it gives a skilled player (and the designer itself) a wide palette of experimentation. The choice of limiting the parameter space on a 2-D coordinate system representing the main controls involved in the granular synthesis could be considered as a will of ours to attempt on a good trade-off.

It also comes to our mind that the methodological approach to the evaluation of such systems plays an important role in the evaluation itself. We chose, for instance, the MITDS framework since

> it focuses on human approaches to real-time control of the multidimensional parameter spaces in a musical composition or performance[6].

Accordingly, we have defined our parameter space first, and then refined it according to the proposed measures. Much of the instrument reshaping has been based on feedback arising from several trials. However, these trials have been performed by the designers themselves, who are both performers though, and audience feedback is missing. Thus, most of our considerations about perceptibility should be proved through an open performance. We have also to take into consideration some probable bias introduced by the designers' skills and deep knowledge of computer music and electroacoustic field. Nevertheless, computer music pratictioners are invaluable in informing the design of new digital musical instruments[5].

We have also chosen to maintain a clear mapping between input parameters from the Myo™ and the parameter for synthesis as suggested by several authors. A more complex and over-structured model is not useful for our purpose and, indeed, could widen the gap between the system model and the audience understanding of it. Putting some constraints onto the space of mapping can be favourable to guide the two sides of the performance toward the same back end.

4. CONCLUSIONS

In this paper, a system that lets the user to affect in real-time a granular synthesis engine built in Max/MSP has been presented and reviewed according to different frameworks proposed in the literature. Several efforts have been made by the Authors to provide a good trade-off between the amount of controlled parameters and the intelligibility of the mapping model for both the performer and the audience side. We have explored the possibilities given by a hardware device such as Thalmic Labs' Myo™ . We would like finally to stress the many advantages in using it within the DMI field to extend expressiveness and improve the building of a true coupling between human gestures and digital synthesis parameters.

5. REFERENCES

[1] A. T. Cabreira and F. Hwang. An analysis of mid-air gestures used across three platforms. In *Proceedings of the 2015 British HCI Conference*, pages 257–258. ACM, 2015.

[2] S. Jordà and S. Mealla. A methodological framework for teaching, evaluating and informing nime design with a focus on expressiveness and mapping. In *Proceedings of the International Conference on New Interfaces for Musical Expression*, volume 30, pages 233–238, 2014.

[3] T. Magnusson. Designing constraints: Composing and performing with digital musical systems. *Computer Music Journal*, 34(4):62–73, 2010.

[4] E. R. Miranda and M. M. Wanderley. *New digital musical instruments: control and interaction beyond the keyboard*, volume 21. AR Editions, Inc., 2006.

[5] S. O'modhrain. A framework for the evaluation of digital musical instruments. *Computer Music Journal*, 35(1):28–42, 2011.

[6] D. Overholt. The musical interface technology design space. *Organised Sound*, 14(2):217–226, 2009.

[7] C. Roads. *Microsound*. MIT press, 2004.