DOCUMENTATION

Contents

Acknowledgements

Myo Mapper is a software developed by Balandino Di Donato and supported by Integra Lab , Birmingham Conservatoire, and Birmingham City University.

License

1 Introduction

Myo Mapper is an open source application to map Thalmic Labs's Myo armband[1] into OSC[2] and MIDI[3] parameters. Myo Mapper can be downloaded here `http://www.balandinodidonato.com/myomapper/` and its source code here: `https://github.com/balandinodidonato/MyoMapper`.

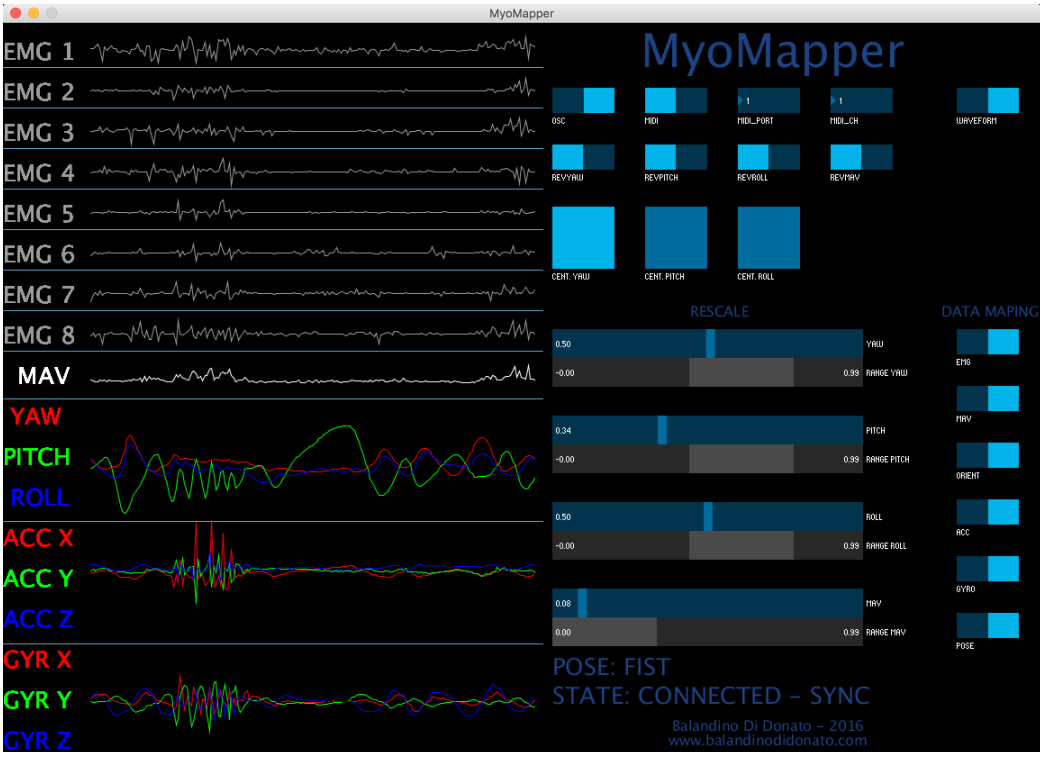

Figure 1: MyoMapper screenshot

[1]The Thalmic Labs's Myo armband is a wearable gesture control and motion control device that lets you take control of your phone, computer, and so much more, touch-free. Find out more here: `https://www.myo.com/`.

[2]Open Sound Control (OSC) is a protocol for communication among computers, sound synthesizers, and other multimedia devices that is optimized for modern networking technology. Find out more, `http://opensoundcontrol.org/introduction-osc`

[3]MIDI is an industry standard music technology protocol that connects products from many different companies including digital musical instruments, computers, tablets and smartphones. `https://www.midi.org/`

2 Install Myo Mapper

To get Myo Mapper working you have to follow just few steps listed below.

- Download Myo Connect for Windows or Mac[4].
- Launch Myo Connect
- Install the unlock.myo connector.
- Open the Application Manager
  - +Add
  - <MyoMapper folder>/unlock.myo
- It is not mandatory, yet it is better if you disable off all others scrips.
- Connect your Myo armband from the Armband Manager
- Verify Myo Connect detect your gestures
- Launch Myo Mapper

For more support about adding new connectors please visit the Myo support page[5].

[4]Myo Connect, `https://developer.thalmic.com/downloads`

[5]Adding new connectors, `https://support.getmyo.com/hc/en-us/articles/204156049-Adding-new-Connectors`

3 GUI

Myo Mapper's GUI is split in two. On the left side of the widow a visual representation of all Myo's data and on the right side the controls to manage them.

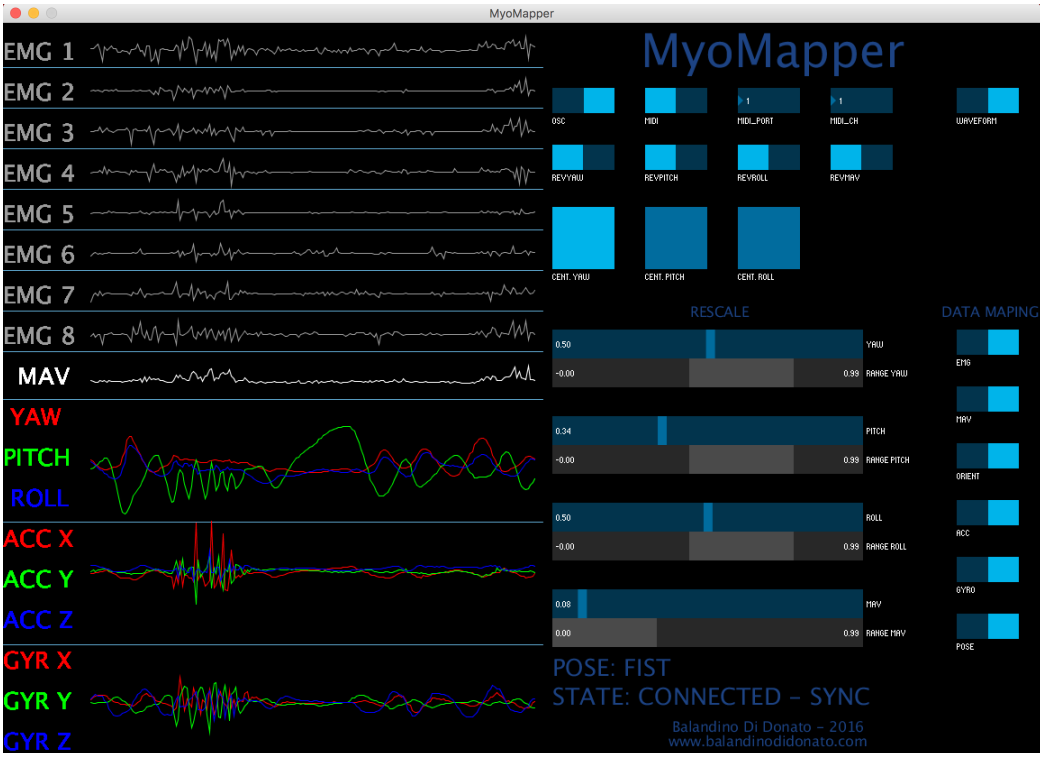

Figure 2: Myo Mapper screenshot

3.1 Myo data visualisation

The Myo data visualisation can be enabled or disabled by the toggle on the top right of the window. By disabling the Myo data visual representation you can save CPU, which in some cases may be useful for other process.

Figure 3: Visual representation toggle

3.2 EMGs

The first portion of graph on the left side desctibes the 8 EMG signals.

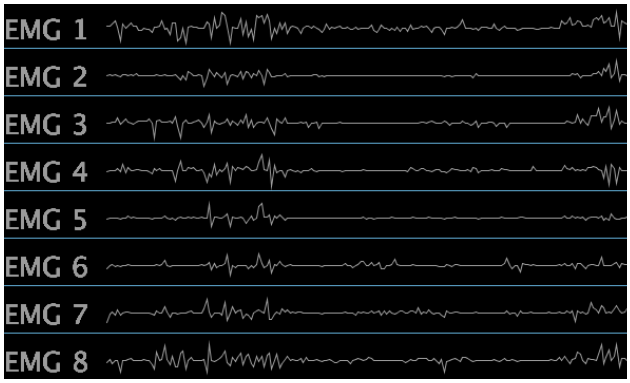

Figure 4: Visual representation toggle

EMG signals have been numbered following the enumeration by Thalmic Lab[6].

Figure 5: Source immage: Arief, Z., Sulistijono, I. A., & Ardiansyah, R. A. (2015) Comparison of five time series EMG features extractions using Myo Armband. *International Electronics Symposium (IES)*, (pp. 11-14). IEEE. Chicago.

[6]EMG pad enumeration, `https://developer.thalmic.com/forums/topic/255/`

3.3 MAV

MAV stand for Mean Absolute Value, which in this case refers to the EMGs' MAV. In Myo Mapper the EMG's MAV is calculated through the following formula from (Arief et al. 2015)[7].

$$MAV = \frac{1}{N}\sum_{k-1}^{N}|X_k|$$

Figure 6: Mean Absolute Value (MAV)

3.4 Orientation

The orientation data: yaw, pitch and roll are displayed in a merged graph, and they appear respectively in green, red and blue.

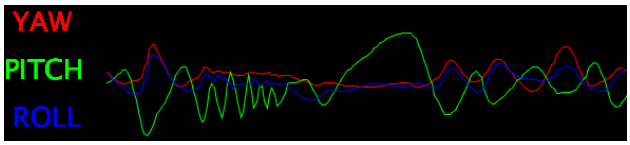

Figure 7: Orientation data

3.5 Acceleration

The acceleration data have been represented following the same method adopted for the orientation data. Here, acceleration x, y and z are respectively red, green and blue.

[7] Arief Z., Sulistijono I. A., Ardiansyah R. (2015) A. Comparison of five time series EMG features extractions using Myo Armband, *International Electronics Symposium (IES)*, 11 - 14, Surabaya.

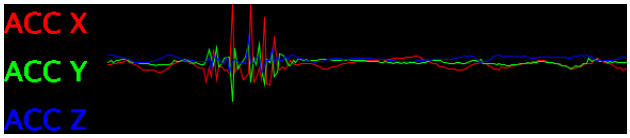

Figure 8: Acceleration data

3.6 Gyro

As for the acceleration data the gyro data have been represented following the same method adopted for the orientation data. Here, gyro x, y and z are respectively red, green and blue.

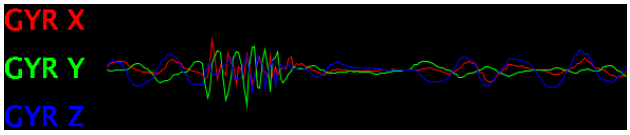

Figure 9: Gyro data

3.7 OSC and MIDI control

On the right side of the Myo Mapper, all controls to manage the Myo's data.

The first line of controls is composed by a OSC and MIDI toggle to respectively allow or not Myo Mapper to send OSC and MIDI messages to third applications. The number boxes next to the MIDI toggle are to set up the MIDI port and channel (Fig. 10).

Moreover, it is possible to select the data to send to the third application by enabling or disabling respective the toggles on the right side (Fig. 11).

Figure 10: OSC and MIDI control

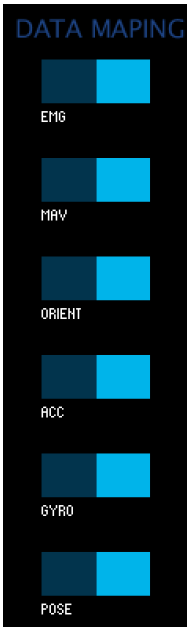

Figure 11: OSC and MIDI data mapping

3.8 Myo data elaboration

Myo data reference point, specifically the Yaw data is established once the Myo is turned on. Thus, incoming Myo data may be different despite orienting the Myo in the same position. Moreover, it has been experienced by many Myo users a data drift of the yaw parameter. For this reasons, in Myo Mapper are implemented a functions to reverse, rescale and centre Myo data.

Reverse Data

Reverse data toggles allow the user to reverse the yaw, pitch, roll and MAV value.

Figure 12: Reverse

Centre Data

Centre data bangs allow to set the current yaw, pitch, roll values at 0.5 and so rescaling the data accordingly.

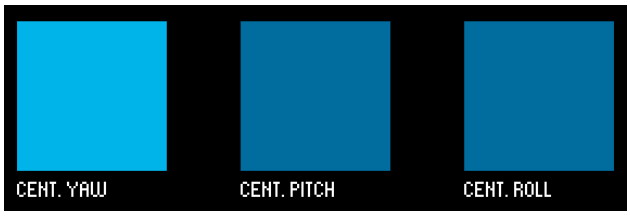

Figure 13: Centre data

Rescale Data

Rescale data sliders, grey and are placed just under a blue sliders, which function is to monitor the data (Fig. 14). Rescale data sliders allow to rescale the Myo data within a certain range. It is possible to set a minimum value with the left edge of the slider, the maximum with the right edge and the whole range can be transposed by moving the slider from its centre.

Figure 14: Rescale data

3.9 Pose and Myo status

The recognised hand posed and the status are displayed by two labels underneath the rescale data sliders.

Figure 15: Pose and Myo status

4 OSC and MIDI

4.1 OSC connection

Myo Mapper sends OSC messages at the port number 5432. To change it:

- Open the osc.pde file within the MyoMapper folder

  ```
  open <path>/<to>/MyoMapper/MyoMapper/osc.pde
  ```

- Edit the number port (latest number) at the 10th code line

  ```
  myRemoteLocation = new NetAddress("127.0.0.1",5432);
  ```

- Save the osc.pde file
- Build Myo Mapper or compile it from Processing.

4.2 OSC Mapping

Myo parameter	**OSC tag**	**n. values**	**Range Values**
EMG 1, EMG 2, ..., EMG 8	/emg	8	-128 , 127
MAV	/emgMav	1	0. , 1.
Yaw, Pitch, Roll	/orientation	3	0. , 1.
Acc X, Y, Z	/acc	3	0 , 1000
Gyro X, Y, Z	/gyro	3	0. , 2PI
Pose	Pose	2	”pose”, 0 - 5

Following the hand pose mapping.

Pose	String	Int
Rest	REST	0
Fist	FIST	1
Fingers Spread	FINGERS_SPREAD	2
Wave In	WAVE_IN	3
Wave Out	WAVE_OUT	4
Double Tap	DOUBLE_TAP	5

4.3 MIDI connection

The MIDI port can be changed through the user interface once you have built the application. In order to change MIDI channel to which send MIDI data it is also doable through the GUI. To change cc value of the single Myo value, you need to open the relative pde file and edit the parameters of the functions which send MIDI data. For Example, to change cc values relative to Myo acceleration values you have to open the myoAcceleration.pde file

```
open <path>/<to>/MyoMapper/MyoMapper/myoAcceleration.pde
```

Then, edit the function to send MIDI data.

```
myBus.sendControllerChange(chMIDI, 4, rollMIDI);
```

where *chMIDI* is the midi channel, *4* is the cc value and *rollMIDI* is the velocity value.

4.4 MIDI mapping

Myo parameter	cc value	Velocity
EMG1	1	0, 127
EMG2	2	0, 127
EMG3	3	0, 127
EMG4	4	0, 127
EMG5	5	0, 127
EMG6	6	0, 127
EMG7	7	0, 127
EMG8	8	0, 127
MAV	9	0, 127
Yaw	10	0, 127
Pitch	11	0, 127
Roll	12	0, 127
Acc X	13	0, 127
Acc X	14	0, 127
Acc X	15	0, 127
Gyro X	16	0, 127
Gyro X	17	0, 127
Gyro X	18	0, 127
Pose: Rest	19	0
Pose: Fist	19	25
Pose: Fingers Spread	19	50
Pose: Wave In	19	76
Pose: Wave Out	19	101
Pose: Double Tap	19	127

5 Build Myo Mapper

1. Download:
   - Processing 2.2.1[8]
   - Libraries for Processing:
     - The midibus[9]
     - ControlP5[10]
     - oscP5[11]
     - Myo For Processing[12]
2. Create a folder called Myo Mapper
3. Open a Terminal and change your directory to the Myo Mapper Folder

   ```
   cd <path>/<to>/MyoMapper
   ```

4. Clone the repository

   ```
   git clone https://github.com/balandinodidonato/MyoMapper
   ```

5. Open the MyoMapper.pde file
6. open MyoMaper/MyoMapper.pde
7. Navigate to the top bar and select:
   - File
   - Export Application
8. Select the Platform which you will be working on
9. Click on Export

[8]Processing, `https://processing.org/download/`
[9]midibus, `http://www.smallbutdigital.com/themidibus.php`
[10]ControlP5, `http://www.sojamo.de/libraries/controlP5/`
[11]oscP5, `http://www.sojamo.de/libraries/oscP5/`
[12]Myo for Processing, `https://github.com/nok/myo-processing`

6 Interactive performance with Myo and Myo Mapper

Myo Mapper is a software which enable the user to drive audio and visual elaborations, within third software able to receive OSC and MIDI messages, using the Myo armband.

To get started with your interactive performance you can download examples for Integra Live[13], Pd[14] and Max[15] trough the following link.

`http://www.balandinodidonato.com/myomapper/`

[13]Integra Live, `http://integralive.org/`
[14]Pure Data (aka Pd), `https://puredata.info`
[15]Max, `https://cycling74.com/products/max/`