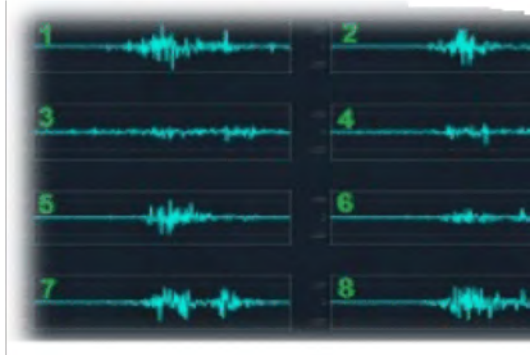


Lecture of Myo



Reading data from the hand

Mitsuhiro Hayashibe

Dai Owaki, Tohoku Univ.

Robotics Department, Biomechanical Eng. Department

Do you know Myo?

Made by Thalmic Labs

Wearable sensor for control with the arm and hand

Mouse & Keyboard

Presentation Control

Robot Control



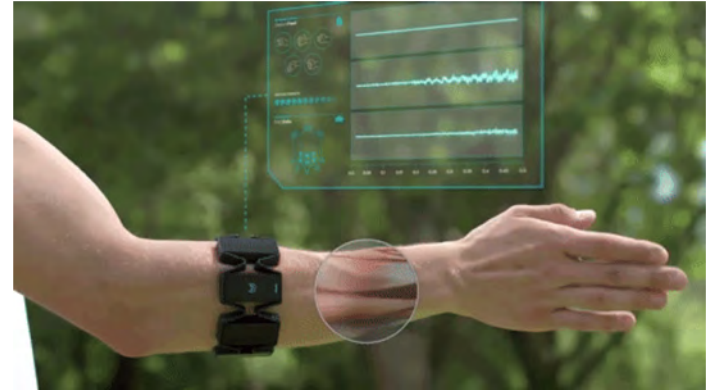
Myo production has officially ended as of Oct 12, 2018 and is no longer available for purchase.

- The Myo team

Electromyography

Electrical signals from Central Nervous System activate the contraction of the muscles

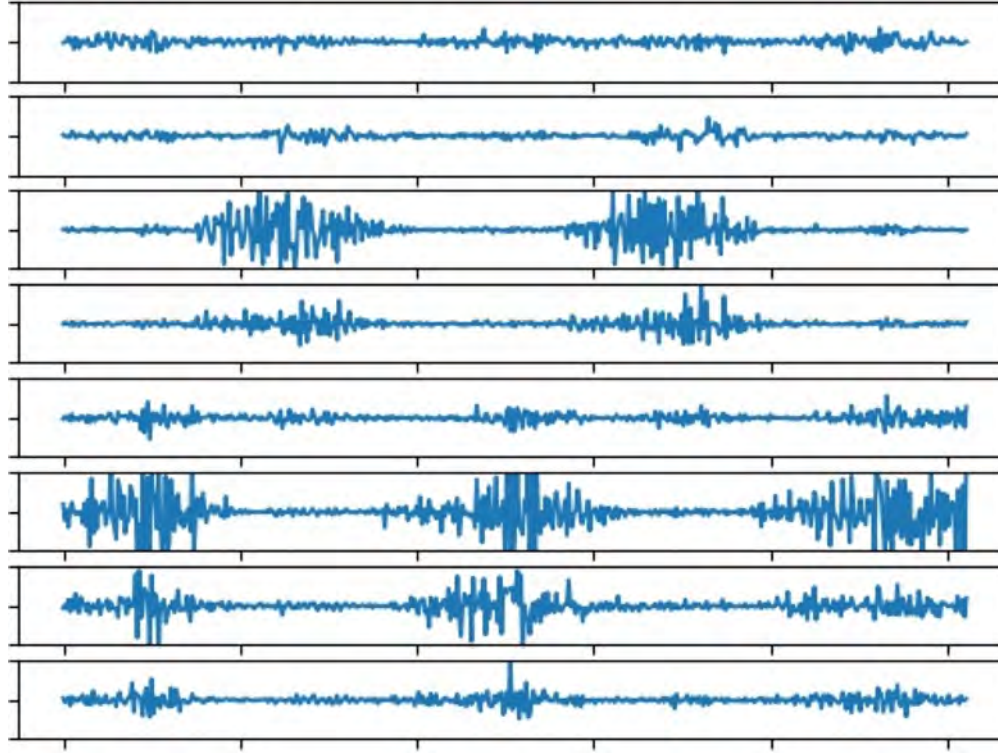
By using two electrodes, it is possible to read this electrical signal



Position is important!

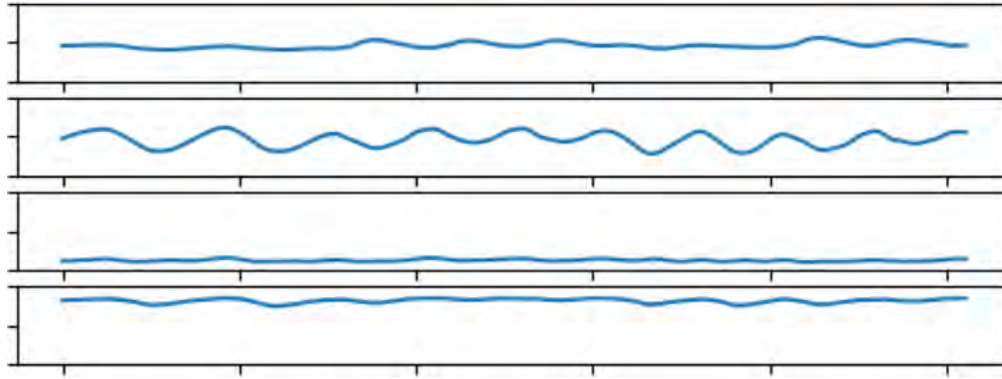
The placement of electrodes and skin impedance might affect the signal

Get data from muscles!



Get data from the arm!

.Orientation



Get data from the hand!



OPEN



WAVE OUT



WAVE IN



FIST



PINCH

Let's use Myo with python!

.Open-myo: <https://github.com/Alvipe/Open-Myo>

.Dependencies: `bluepy` (Linux only)

.Scripts

- MyoClient.py
- MyoSample.py
- Poweroff.py



MyoClient

- Listen Myo and raise **events**
- Start a new **thread** to acquire Myo data
- It does not choose the device!
 - Must be selected manually

Event handlers

add_**emg**_event_handler(on_emg)

.emg[8]

add_**imu**_event_handler(on_imu)

.quat[4], acc[3], gyro[3]

add_**sync**_event_handler(on_sync)

.arm = {right, left}, x_direction = {wrist, elbow}

add **classifier** event hanlder(on_classifier)

Let's try some examples

```
1 import MyoClient as myo
2 import numpy as np
3
4 def on_emg(emg):
5
6
7
8
9
10
11 myo_mac_addr = myo.get_myo()
12 print("MAC address: %s" % myo_mac_addr)
13 myo_device = myo.Device(myo_mac_addr)
14 myo_device.services.sleep_mode(1) # never sleep
15 myo_device.services.set_leds([128, 128, 255], [128, 128, 255])
16 myo_device.services.vibrate(1) # short vibration
17
18 myo_device.services.emg_filt_notifications()
19 myo_device.services.set_mode(myo.EmgMode.FILT,
20                               myo.ImuMode.DATA, myo.ClassifierMode.ON)
21
22 myo_device.add_emg_event_handler(on_emg)
23
24 myo_device.run()
```

Set leds green if sum of emg > 200
Otherwise red

#4 handler

#14-16 set properties

#18 use filtered EMG

#22 set the handler

#24 start acquisition

Now with poses

```
1 import MyoClient as myo
2 import numpy as np
3
4 def on_classifier(pose):
5     rest = 0x00, fist = 0x01, wave in = 0x02, wave out = 0x03,
6     fingers spread = 0x04, double tap = 0x05, unknown = 0xff
7
8
9
10 myo_mac_addr = myo.get_myo()
11 print("MAC address: %s" % myo_mac_addr)
12 myo_device = myo.Device(myo_mac_addr)
13 myo_device.services.sleep_mode(1) # never sleep
14 myo_device.services.set_leds([128, 128, 255], [128, 128, 255])
15 myo_device.services.vibrate(1) # short vibration
16
17 myo_device.services.classifier_notifications()
18 myo_device.services.set_mode(myo.EmgMode.FILT,
19                             myo.ImuMode.DATA, myo.ClassifierMode.ON)
20
21 myo_device.add_classifier_event_handler(on_classifier)
22
23
24 myo_device.run()
```

#4 handler

#14-16 set properties

#18 use classifier

#22 set the handler

#24 start acquisition