

MATALAB INTERFACE FOR 3DOF MICHELANGELO PROSTHESIS

Quick start guide and reference manual

V 1.1

Table of Contents

About this interface	3
Installation	3
Usage.....	4
<i>Getting started with the class</i>	4
<i>Example</i>	5
Troubleshooting.....	5

About this interface

By default the Michelangelo is velocity controlled (i.e. this is the default OttoBock implementation). The MATLAB wrapper is developed with the aim to overcome this limitation and extend the functionality of the prosthesis in one easy-to-use interface. The main idea is to provide several layers of interaction:

- Velocity control of the prosthesis, as implemented by OttoBock;
- Custom implemented position control of the prosthesis;
- Raw feedback variables such as EMG, force, positions of the actuators;
- Physical feedback variables (i.e., normalized raw feedback data);

The MATLAB wrapper simply encapsulates the functionalities developed in the C# and makes them available to the user. Therefore, it doesn't possess any specific control/processing logic.

Installation

- Download the package from the svn repository (<https://svn.bccn.uni-goettingen.de/usvn/svn/Mikey>) and place it on the desired path but **do not** change the internal folder structure.
- In the drivers' folder locate the setup.exe and start it with admin privileges. Install the drivers and ignore the warning about the digital signature (i.e. certificate). Download from: http://professionals.ottobock.de/cps/rde/xchg/ob_de_de/hs.xsl/10398.html
- Make sure that you have .NET 4.0 or later installed on your machine (if you don't have it install it from <http://www.microsoft.com/en-us/download/details.aspx?id=17851>).
- Open MATLAB with admin privileges. Run the setup script Mikey_Setup.m. This should register the DLL's for MATLAB usage. If it fails do it manually:
 - Locate the windows command prompt (cmd) and open it with admin privileges. Copy and paste the following commands into it:

```
cd C:\Windows\Microsoft.NET\Framework64\v4.0.303191
regasm <path to the provided DLLs>\Michelangelo.dll /tlb2
```
 - The system should notify you of the successful operation.
 - Run MATLAB in Admin mode and add Mikey root folder to the search path (Set Path -> Add Folder); Adding subfolders is not necessary

¹ This is an example path; your path can differ depending on the .NET/Windows version.

² In the <DLL path> the spaces are not supported.

Usage

- Michelangelo must be in the Bluetooth mode (press & hold the power button until the second long beep – the light should be blue).
- Plug in the Bluetooth dongle. The new Michelangelo Hand uses fast, while the old one uses the slow dongle. Make sure you are using the appropriate dongle.

Getting started with the class

1. We need to connect to the hand using the provided class constructor:

```
mike = michelangeloHand('DumpMode', 'val', 'PullFrequency', val);
```

If successful, the matlab command window will display something like this:

```
Mikey is ready to go :)
```

The input parameters name and their values

'DumpMode': can be set as 'SensorsEMG', 'Sensors', or 'EMG';

'DongleSpeed': can be set as 'slow', 'fast';

'PullFrequency': can be set a number in the range (0,100), this is the refresh frequency at which MATLAB reads the data from the DLL (no data is being lost; it is always transmitted in batches).

2. In order to start/stop acquisition simply call:

```
mike.startMonitoring();  
mike.stopMonitoring();
```

The data is logged into internal MATLAB buffer, but the current sensor readouts are also available

3. You can also issue commands to the hand via the following set of methods:

```
mike.reachNeutralPreshape();  
mike.reachPreshape('argument_name', argument_val);  
mike.reachPosition('argument_name', argument_val);
```

4. When finished with your work make sure you properly disconnect from the hand by calling:

```
mike.delete();
```

Example

Obtain the EMG from the AC electrodes, calculate and plot the corresponding RMS:

```
mike = michelangeloHand('DumpMode', 'EMG', 'PullFrequency', 20);

fh=figure; clf; set(gcf,'color','w');
lh = addlistener(mike,'NewDataAvaliable', @(src,
event)processTheData(src, event, fh));

function processTheData( src, event, fh )
%   Detailed explanation goes here
RMSwindow = 200; % in ms
RMSwindow = min(RMSwindow,src.logCountEMG);
rawEMG=src.logSensors.EMG(src.logCountEMG-
RMSwindow+1:src.logCountEMG,:);
EMGmean=mean(rawEMG,1);
EMG=rawEMG-repmat(EMGmean,RMSwindow,1);
clf;

subplot(2,1,1)
plot(EMG(:,6))
ylim([-4000,4000])

subplot(2,1,2)
plot(EMG(:,7))
ylim([-4000,4000])
end
```

Troubleshooting

- Q: The prosthesis cannot connect!
A: Take out the Bluetooth dongle and switch off/on the prosthesis. If the problem persists restart MATLAB also.
- Q: I have got BSOD!
A: Turn off the power save/sleep mode while the prosthesis is connected.
- Q: Prosthesis is not responding properly to the commands or shows unusual feedback values.
A: Connection has been lost. You will need to reconnect.
- Q: MATLAB crashed!
A: That is normal ...