MyoMex Quickstart

Table of Contents

Before you begin, please read through README.txt and follow all steps for setting up the Myo Connect application, Myo SDK, and building the MEX function `myo_mex`.

Mark Tomaszewski, mark@mark-toma.com, marktoma@buffalo.edu

Before Using `MyoMex`

If you decided not to read through `README.txt`, let's at least show you the quickest possible way to get started.

```
install_myo_mex; % adds directories to MATLAB search path
% install_myo_mex save % additionally saves the path

sdk_path = 'C:\myo-sdk-win-0.9.0'; % root path to Myo SDK
build_myo_mex(sdk_path); % builds myo_mex
```

```
Evaluating mex command:
 'mex -O -I"C:\myo-sdk-win-0.9.0\include" -L"C:\myo-sdk-win-0.9.0\lib" -lm

MEX-file 'myo_mex' built successfully!
```

MyoMex Usage

Before using `MyoMex`, you must decide how many Myos wou'd like to use in this session. The MyoMex constructor argument `countMyos` specifies this value. Make sure that exactly `countMyos` devices are connected in the Myo Connect application or else MyoMex construction will fail. The Myo device(s) should also be worn on a human arm to avoid unexpected disconnection of the device from Myo Connect. If Myo Connect loses a device at any time, MyoMex will terminate and invalidate itself.

```
countMyos = 1;
```

Instantiate MyoMex

After constructing a new `MyoMex` instance, we'll inspect its properties.

```
mm = MyoMex(countMyos)
```

```
Warning: The class file for 'MyoData' has been changed, but the change can
```

```
applied because objects based on the old class file still exist. If you us
those objects, you might get unexpected results. You can use the 'clear' c
to remove those objects. See 'help clear' for information on how to remove
objects.

mm =

  MyoMex with properties:

    myoData: [1x1 MyoData]
```

Notice that the only property of `mm` is a `1xcountMyos` MyoData object. There is no device data stored by `mm`. The data from each physical Myo device is passed through `mm` to each element of `mm.myoData`.

Inspect `MyoData`

Since `MyoData` objects inherit from `handle`, we can get handles to the `MyoData` objects representing each physical device and use them directly.

```
m1 = mm.myoData(1);
if countMyos == 2, m2 = mm.myoData(2); end
```

Now, we'll just continue this exercise with `m1`, but the exact same demonstration applies for `m2` as well (if `countMyos == 2`.

The most recent data from Myo will be stored in the relevant properties of the `MyoData` object (i.e. `quat`, `gyro`, `accel`, `emg`, `pose`, etc.). The following is a list of all properties in `MyoData`.

```
pause(0.1); % wait briefly for the first data frame to come in

% data properties sampled on the IMU time base
m1.timeIMU
m1.quat
m1.rot
m1.gyro
m1.gyro_fixed
m1.accel
m1.accel_fixed

% data properties sampled on the EMG time base
m1.timeEMG
m1.emg
m1.pose
m1.pose_rest
m1.pose_fist
m1.pose_wave_in
m1.pose_wave_out
m1.pose_fingers_spread
m1.pose_double_tap
m1.arm
m1.arm_right
m1.arm_left
m1.arm_unknown
```

```
m1.xDir
m1.xDir_wrist
m1.xDir_elbow
m1.xDir_unknown


        ans =

            0.1330


        ans =

           -0.7107   -0.4577   -0.3240   -0.4248


        ans =

            0.4290   -0.3072    0.8494
            0.9004    0.2201   -0.3752
           -0.0716    0.9258    0.3711


        ans =

           35.7500   14.1875   23.8125


        ans =

           31.2058   26.3787   19.4100


        ans =

           -0.0718    0.7300    0.3325


        ans =

            0.0274   -0.0288    0.8044


        ans =

            0.1730


        ans =

          Columns 1 through 7

           -0.0625   -0.1406   -0.2422         0    0.0156    0.2656   -0.1953
```

```
  Column 8

   -0.1641

ans =

       65535

ans =

     0

ans =

     0

ans =

     0

ans =

     0

ans =

     0

ans =

     0

ans =

     2

ans =

     0

ans =

     0
```

```
ans =

     1


ans =

     2


ans =

     0


ans =

     0


ans =

     1
```

Using Logged Data

As `MyoData` receives data from `MyoMex`, it's automatically accumulated in so-called `data_log` properties, i.e. `quat_log`, `accel_log`, etc. We refer to this as the streaming mode of a `MyoData` object. This status is indicated by the `isStreaming` property.

```
m1.isStreaming
```

```
ans =

     1
```

We can inspect the accumulation of the logs for example,

```
fprintf('%10s%10s\n','time','samples')
for ii = 1:5
  fprintf('% 8.2f%10d\n',...
    m1.timeIMU,size(m1.quat_log,1));
  pause(0.2);
end
fprintf('\n\n');
```

```
      time   samples
      0.17         9
      0.37        19
      0.57        29
```

```
0.77        39
0.97        49
```

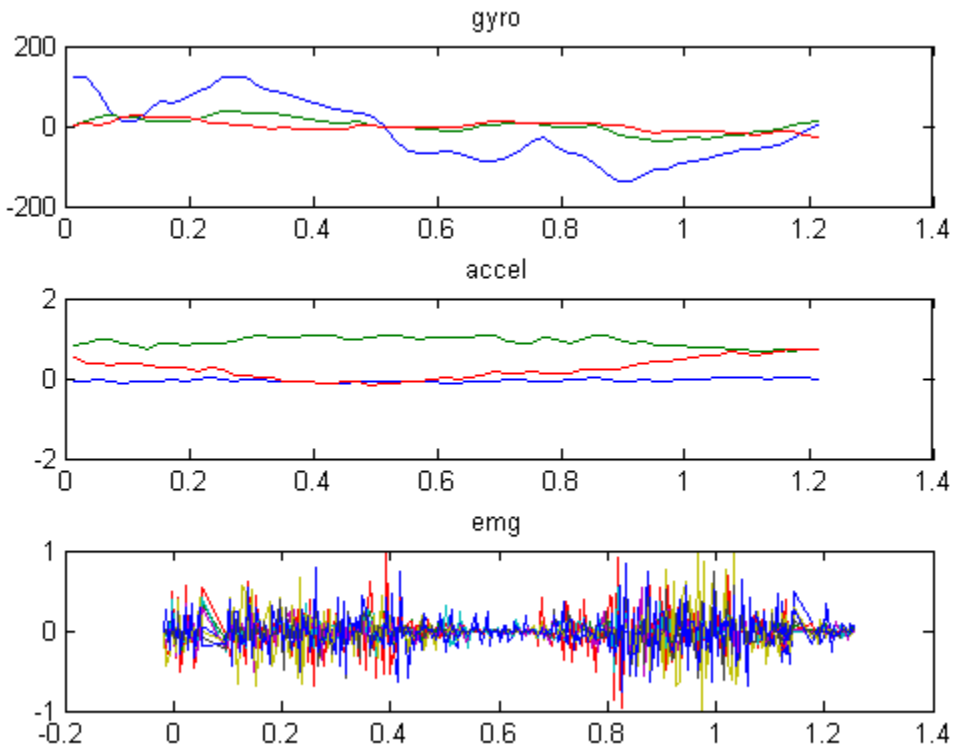
Although we can't stop the data from being passed to in `MyoData`, we can toggle streaming mode by using the methods `stopStreaming()` and `startStreaming()`.

```
m1.stopStreaming();
fprintf('Number of samples:                \t%d\n',length(m1.timeIMU_log));
pause(1);
fprintf('Number of samples after pause(1):\t%d\n',length(m1.timeIMU_log));
```

```
Number of samples:                 61
Number of samples after pause(1): 61
```

Now we can plot some data taking care to use the correct time vectors.

```
figure;
subplot(3,1,1); plot(m1.timeIMU_log,m1.gyro_log);  title('gyro');
subplot(3,1,2); plot(m1.timeIMU_log,m1.accel_log); title('accel');
subplot(3,1,3); plot(m1.timeEMG_log,m1.emg_log);   title('emg');
```



If you'd like to clear the `data_log` properties to start a new logging trial, then you may use the `clearLogs()` method,

```
% collect about T seconds of data
T = 5; % seconds
```

```
m1.clearLogs()
m1.startStreaming();
pause(T);
m1.stopStreaming();
fprintf('Logged data for %d seconds,\n\t',T);
fprintf('IMU samples: %10d\tApprox. IMU sample rate: %5.2f\n\t',...
  length(m1.timeIMU_log),length(m1.timeIMU_log)/T);
fprintf('EMG samples: %10d\tApprox. EMG sample rate: %5.2f\n\t',...
  length(m1.timeEMG_log),length(m1.timeEMG_log)/T);
```

```
Logged data for 5 seconds,
 IMU samples:        251 Approx. IMU sample rate: 50.20
 EMG samples:       1000 Approx. EMG sample rate: 200.00
```

Finally, when you're done with `MyoMex`, don't forget to clean up!

```
mm.delete;
clear mm
```

Finally, take advantages of the following resources for additional information about `MyoData`!

```
% MyoMexGUI_Monitor
% properties MyoData
% methods MyoData
% help MyoData
```

Published with MATLAB® R2013a