6/13/2016

Several data collection / analysis tools are available for processing data recorded in the MiniVIE. This guide will provide an overview of each type and how to process.

* Raw Training Data Files (\*.trainingData)
  + These files contain (1) the EMG waveform (2) the associates features and (3) the class labels in addition to other parameter information and meta data
  + Example fields:

>> s = load('myFile.trainingData','-mat')

s =  
 signalData: [16x250x538 single]

features3D: [16x4x538 double]

classLabelId: [1x538 double]

enableLabel: [1x538 logical]

classNames: {11x1 cell}

featureNames: {'MAV' 'LEN' 'SSC' 'ZC'}

activeChannels: [1 2 3 4 5 6 7 8]

sampleRateHz: 1000

* + The data fields can then be plotted directly
  + However the preferred way to load and manipulate the data is through the TrainingDataAnalysis class:

>> h = TrainingDataAnalysis('myFile.trainingData')

[TrainingData] Creating Empty Training Data Object

[TrainingDataAnalysis] Creating Training Data Analysis Object from file: "myFile.trainingData"

[TrainingData] Loading file: "myFile.trainingData"

[TrainingData] Loading 538 Samples

[TrainingData] Setting window size to 250.

[TrainingData] Sample rate empty. Assuming 1000Hz

h =

TrainingDataAnalysis with properties:

fullFileName: 'myFile.trainingData'

Name: ''

Verbose: 1

SaveInOldFormat: 0

HasUnsavedData: 0

SampleCount: 538

SampleRate: 1000

ActiveChannels: [1 2 3 4 5 6 7 8]

ActiveFeatures: [1 2 3 4]

ClassNames: {11x1 cell}

FeatureNames: {'MAV' 'LEN' 'SSC' 'ZC'}

MaxChannels: 16

WindowSize: 250

NumClasses: 11

NumActiveChannels: 8

NumFeatures: 4

* + The data can then be explored through the plot\_\* functions:  
      
    methods('TrainingDataAnalysis')

Methods for class TrainingDataAnalysis:

TrainingDataAnalysis getClassData getRawSignals plot\_emg\_with\_breaks setClassNames

addTrainingData getClassLabelCount getStichedData plot\_features\_sorted\_class setFeatureNames

allocateMemory getClassLabels hasData recomputeFeatures validate

clearData getContinuousData initialize removeTrainingData

disableDataBySample getDataTable loadTrainingData saveTrainingData

disableLabeledData getEnabledClassLabels plot\_emg\_filtered saveobj

enableAllLabeledData getFeatureData plot\_emg\_per\_class setActiveChannels

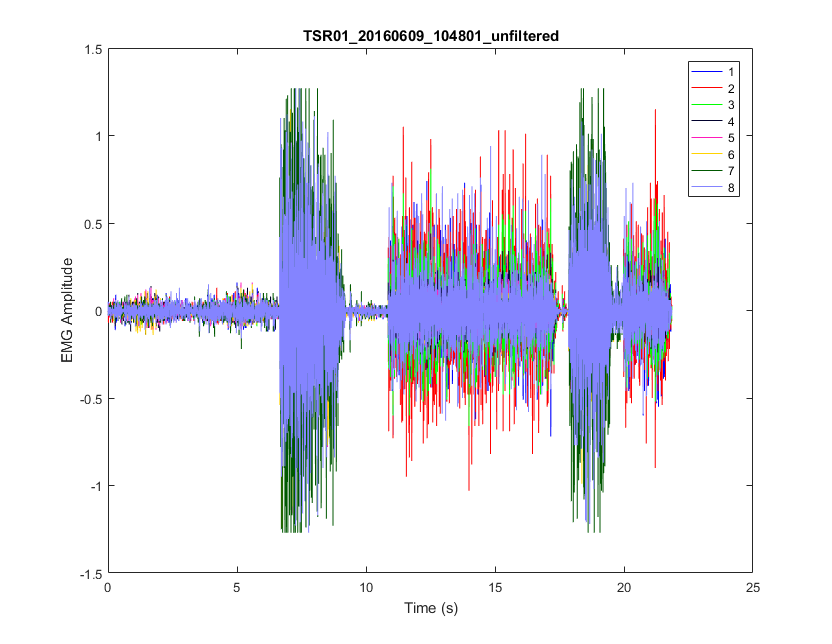
getAllClassLabels getMetaLabels plot\_emg\_unfiltered setActiveFeatures

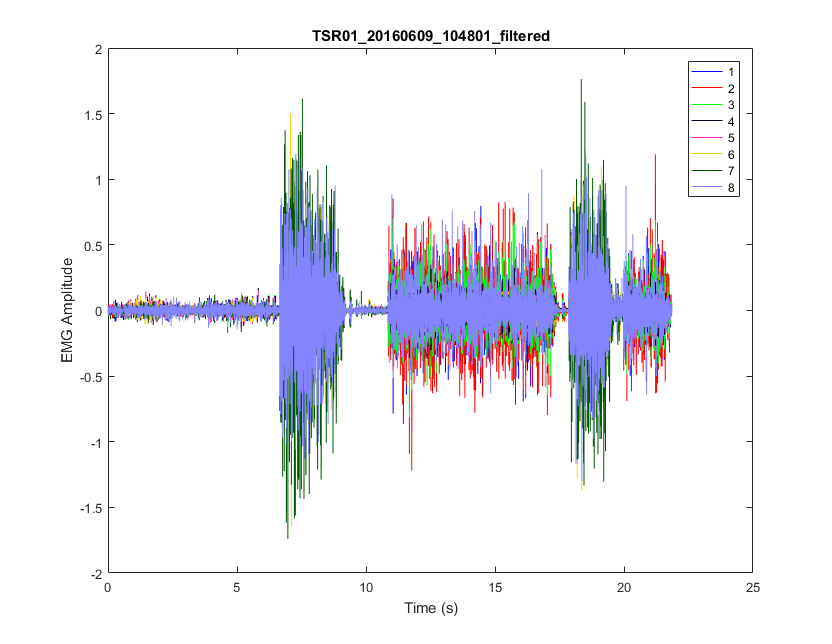
Static methods:

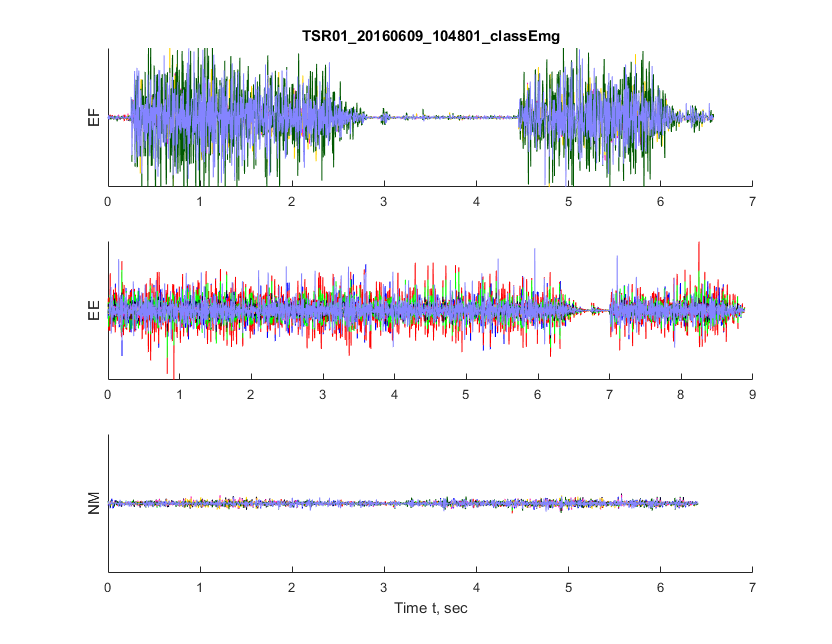
batchRunQuickLook load\_data plot\_mav\_per\_class

filter\_data loadobj plot\_one\_class\_emg

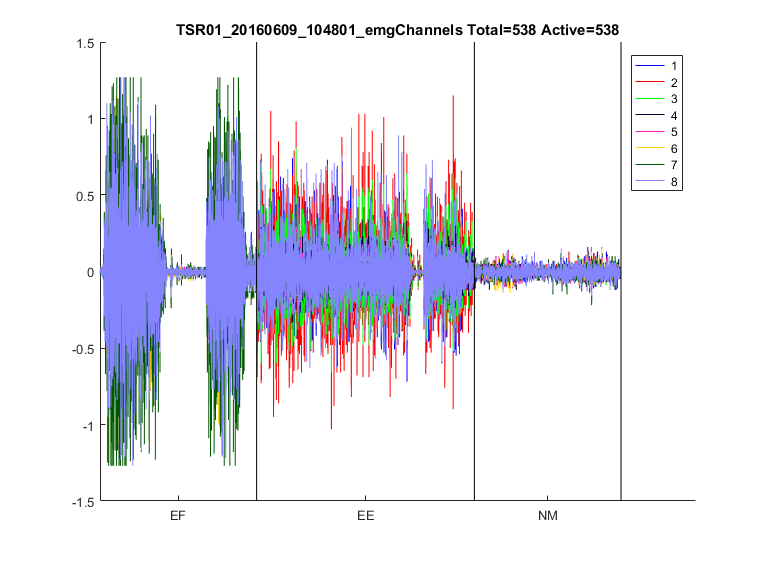
* + plot\_emg\_unfiltered: EMG response before any filtering applied, as recorded.



* + plot\_emg\_filtered: EMG with nominal filtering applied.   
    Note: this doesn’t have to be the filtering used during collection.   
    Note: for Myo bands, no filtering is typically applied, so this looks like the unfiltered  
    
  + plot\_emg\_per\_class: Since the training set may contain segments of contractions with multiple repetitions, find all the EMG data for each motion class (abbreviated). E.g. EF = Elbow Flexion, EE = Elbow Extension, NM = No Movement.



* + plot\_emg\_with\_breaks: Plot the EMG training history with all the repetitions grouped.



* + plot\_features\_sorted\_class: Plot the EMG training history as features with all the repetitions grouped.



* TAC Assessment

The Target Achievement Control (TAC) assessment presents the user with a target arm position and then times how long the user take to achieve that position

TODO: Complete Document