

```

clear
close all

% We're here to remove "off" sections from data

load Filtered_EMG.mat
Sum1 = zeros(8,167761);
% Field names index

% Pulling the sampling rate and setting up a time vector
Fs = EMGfilt.HO.posture.Fs(1,1);
t = EMGfilt.HO.posture.Time(1,:);

Names = fieldnames(EMGfilt);

for j=1:11
    Sum1(j,:) = sum(EMGfilt.(Names{j}).posture.Data,1);
    % Summing up all the signals for each posture
end

Sum2 = sum(Sum1,1);
% Summing up the posture signals

ABS = abs(Sum2);
% Making all of the signal positive

SmoothBINsize = 800; % bin size in ms for smoothing the data
Smoothed = smoothdata(ABS,'movmean',(SmoothBINsize*1000/Fs));

%OnOffIndex = zeros(1,167761);
%
% for i = 1: 167761
% setting a threshold for when the combined signal is "on" vs. "off"
%     if Smoothed(1,i) > 0.00021
%         OnOffIndex(1,i)=1;
%     else
%         OnOffIndex(1,i)=0;
%     end
% end
% I really only need to do this the one time, so I saved the resulting
% matrix as "On_Off_index_Training"

load("On_Off_index_Training.mat")

figure(1)
plot(t,Smoothed,t,(OnOffIndex/1000))

EMG_on=EMGfilt;
% Making a copy of the filtered data to remove the "off" sections for
% feature extraction

for j=1:11
    for i=1:8
        EMG_on.(Names{j}).posture.Data(i,:)=EMG_on.(Names{j}).posture.Data(i,:).*OnOffIndex;
        % Multiplying everything in the "off" sections by zero
    end
end

```

```
end
    EMG_on.(Names{j}).posture.Data(:,all(EMG_on.(Names{j}).posture.Data == 0)) = [];
    % Removing columns of data that are all zeros
end
% Removing the "off" sections of data

figure(2)
plot(EMG_on.HC.posture.Data(1,:))
% Verifying that the "off" sections are removed

% With the exception of some trimming, it looks like I've removed the
% "off sections, I'm saving the structure as "EMG_Off_Removed.mat"
```

