UNIVERSITÀ DEGLI STUDI DI MILANO
BICOCCA

# Advanced Human-System Interfaces

## Third assignment the opptional part

ANASS NASSIRI

```
% Load walking tasks data

data_emg1=load("/MATLAB Drive/Third Assignment Material/Third Assignment
Material/Walking tasks/Session1_Shimmer_F0AD.mat")
```

```
data_emg1 = struct with fields:
    Shimmer_F0AD_EMG_CH1_BandPass_Filter_CAL: [231380x1 double]
    Shimmer_F0AD_EMG_CH2_BandPass_Filter_CAL: [231380x1 double]
               Shimmer_F0AD_Event_Marker_CAL: [231380x1 double]
```

```
data_emg2=load("/MATLAB Drive/Third Assignment Material/Third Assignment
Material/Walking tasks/Session2_Shimmer_F0AD.mat")
```

```
data_emg2 = struct with fields:
    Shimmer_F0AD_EMG_CH1_BandPass_Filter_CAL: [242926x1 double]
    Shimmer_F0AD_EMG_CH2_BandPass_Filter_CAL: [242926x1 double]
               Shimmer_F0AD_Event_Marker_CAL: [242926x1 double]
```

```
data_emg3=load("/MATLAB Drive/Third Assignment Material/Third Assignment
Material/Walking tasks/Session2_Shimmer_F0AD.mat")
```

```
data_emg3 = struct with fields:
    Shimmer_F0AD_EMG_CH1_BandPass_Filter_CAL: [242926x1 double]
    Shimmer_F0AD_EMG_CH2_BandPass_Filter_CAL: [242926x1 double]
               Shimmer_F0AD_Event_Marker_CAL: [242926x1 double]
```

```
% Extract EMG signals from the identified fields
emg_signal1_ch1 = data_emg1.Shimmer_F0AD_EMG_CH1_BandPass_Filter_CAL;
emg_signal1_ch2 = data_emg1.Shimmer_F0AD_EMG_CH2_BandPass_Filter_CAL;

emg_signal2_ch1 = data_emg2.Shimmer_F0AD_EMG_CH1_BandPass_Filter_CAL;
emg_signal2_ch2 = data_emg2.Shimmer_F0AD_EMG_CH2_BandPass_Filter_CAL;

emg_signal3_ch1 = data_emg3.Shimmer_F0AD_EMG_CH1_BandPass_Filter_CAL;
emg_signal3_ch2 = data_emg3.Shimmer_F0AD_EMG_CH2_BandPass_Filter_CAL;
```

### Step 2: Signal Denoising

```
% Display fields to debug  where EMG signals are stored
disp('Fields in data_emg1:');
```

```
Fields in data_emg1:
```

```
disp(fieldnames(data_emg1));
```

```
    {'Shimmer_F0AD_EMG_CH1_BandPass_Filter_CAL'}
    {'Shimmer_F0AD_EMG_CH2_BandPass_Filter_CAL'}
    {'Shimmer_F0AD_Event_Marker_CAL'           }
```

```matlab

disp('Fields in data_emg2:');
```

```
Fields in data_emg2:
```

```matlab
disp(fieldnames(data_emg2));
```

```
    {'Shimmer_F0AD_EMG_CH1_BandPass_Filter_CAL'}
    {'Shimmer_F0AD_EMG_CH2_BandPass_Filter_CAL'}
    {'Shimmer_F0AD_Event_Marker_CAL'           }
```

```matlab

disp('Fields in data_emg3:');
```

```
Fields in data_emg3:
```

```matlab
disp(fieldnames(data_emg3));
```

```
    {'Shimmer_F0AD_EMG_CH1_BandPass_Filter_CAL'}
    {'Shimmer_F0AD_EMG_CH2_BandPass_Filter_CAL'}
    {'Shimmer_F0AD_Event_Marker_CAL'           }
```

### Step 3: Envelope Evaluation

```matlab
function envelope = computeEnvelope(signal, window_size)
    envelope = sqrt(movmean(signal.^2, window_size));
end

window_size = 100; % Example window size for RMS computation

envelope_emg1_ch1 = computeEnvelope(emg_signal1_ch1, window_size);
envelope_emg1_ch2 = computeEnvelope(emg_signal1_ch2, window_size);

envelope_emg2_ch1 = computeEnvelope(emg_signal2_ch1, window_size);
envelope_emg2_ch2 = computeEnvelope(emg_signal2_ch2, window_size);

envelope_emg3_ch1 = computeEnvelope(emg_signal3_ch1, window_size);
envelope_emg3_ch2 = computeEnvelope(emg_signal3_ch2, window_size);
```

### Step 4: Gait Cadence Computation

```matlab
function cadence = computeGaitCadence(envelope, fs)
    [peaks, ~] = findpeaks(envelope, 'MinPeakDistance', fs/2); % assuming
minimum step interval of 0.5 sec
    cadence = length(peaks) / (length(envelope) / fs) * 60; % steps per minute
end
```

```
fs_emg = 512; % Sampling frequency for EMG

gait_cadence1_ch1 = computeGaitCadence(envelope_emg1_ch1, fs_emg);
gait_cadence1_ch2 = computeGaitCadence(envelope_emg1_ch2, fs_emg);

gait_cadence2_ch1 = computeGaitCadence(envelope_emg2_ch1, fs_emg);
gait_cadence2_ch2 = computeGaitCadence(envelope_emg2_ch2, fs_emg);

gait_cadence3_ch1 = computeGaitCadence(envelope_emg3_ch1, fs_emg);
gait_cadence3_ch2 = computeGaitCadence(envelope_emg3_ch2, fs_emg);
```

### Compiling results into a table for both channels

```
% Compile results into a table for both channels
subject_names = {'Subject1', 'Subject2', 'Subject3'};
gait_cadences_ch1 = [gait_cadence1_ch1; gait_cadence2_ch1; gait_cadence3_ch1];
gait_cadences_ch2 = [gait_cadence1_ch2; gait_cadence2_ch2; gait_cadence3_ch2];

results_table_ch1 = table(subject_names', gait_cadences_ch1, 'VariableNames',
{'Subject', 'GaitCadence_CH1'});
results_table_ch2 = table(subject_names', gait_cadences_ch2, 'VariableNames',
{'Subject', 'GaitCadence_CH2'});

% Display results tables
disp('Gait Cadence (steps per minute) for each subject (Channel 1):');
```

```
Gait Cadence (steps per minute) for each subject (Channel 1):
```

```
disp(results_table_ch1);
```

```
      Subject       GaitCadence_CH1
    ____________    _______________

    {'Subject1'}        92.938
    {'Subject2'}        92.947
    {'Subject3'}        92.947
```

```
disp('Gait Cadence (steps per minute) for each subject (Channel 2):');
```

```
Gait Cadence (steps per minute) for each subject (Channel 2):
```

```
disp(results_table_ch2);
```

```
      Subject       GaitCadence_CH2
    ____________    _______________

    {'Subject1'}        90.017
    {'Subject2'}        93.832
    {'Subject3'}        93.832
```

```
%Third Assignmnet optional part
%created by anass nassiri
```