Developing a Biosignal-Specific Processing Pipeline in MATLAB

Athena Nouhi, Sarah Ostadabbas

Abstract— **Electrocardiogram (ECG), electrodermal activity (EDA) and electromyogram (EMG) are among physiological signals widely used in various biomedical applications including health tracking, sleep quality assessment, early disease detection/diagnosis, and human affective state recognition. This project centers around the development of a biosignal-specific processing pipeline written in MATLAB in order to analyze these physiological signals in a modular fashion based on the state-of-the-art studies reported in scientific literature. In this report, we start with providing design description of our graphical user interface (GUI) specialized for each signal. At first, each physiological signal goes through preprocessing steps including noise removal and segmentation. Then, the signal attributes (features) that have been shown highly relevant to a better category discrimination in an intended application are extracted. Lastly, these extracted features are fed to the MathWorks Classification Learner app for classification. This app allows the model training specification, cross-validation scheme farming, and classification result computation. For every algorithm in our GUI, the citations to the latest studies in the field are also provided.**

I. Design of the Graphical User Interface (GUI)

Our GUI consists of three different pages with a user-friendly flow to perform signal processing on different physiological signals. First page is for selection of the signal type, which can be ECG, EDA, or EMG as shown in Fig. 1. The second page performs preprocessing and works slightly different for different signal types. The third page performs feature extraction from preprocessed data obtained in second page and can be customized based on the intended application.

A. Signal-Specific Preprocessing Page

When the type of signal is selected, preprocessing page pops up. On this page, first you need to upload your data and assign its related sampling rate. This data can be a single vector signal or a dataset. Dataset must be in a matrix format with rows of data stack on top of each other. In this page, you can visualize the time series signal at any stage in the preprocessing pipeline. If a dataset is upload, a panel will show up and you can select a particular row number to be plotted. This row number will be used for all other plots on this page. You change this number at any time during preprocessing.

To increase the signal-to-noise ratio (SNR) of your uploaded signal, a filter panel is provided in this page. On filter panel, you can choose the filter type as well as other parameters of the filter such as low and high cutoff frequencies. You have option to select from three types of filters that are most common in physiological signal processing: Elliptic, Butterworth and Gaussian. By pressing the *Apply* button, the filter will be applied to all of the data and *Plot filtered data* button will be activated, so you can plot the filtered signal or the specified data row of the dataset matrix.

The descriptions mentioned above are mainly common for all signal types. There are a few variations in preprocessing steps of each signal types which will be described in details in the related signal section. An example of preprocessing page for ECG signal is shown in Fig. 2.

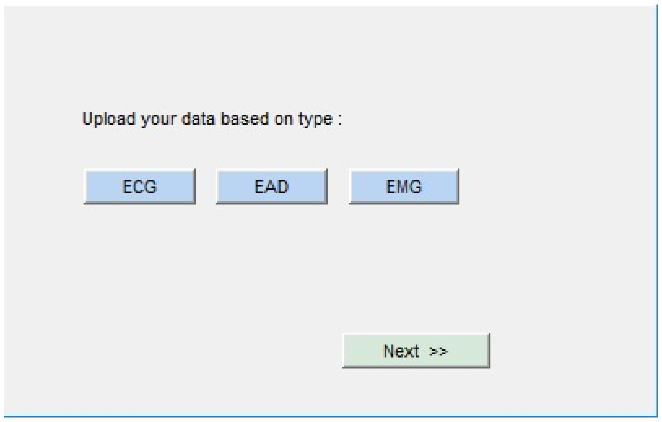
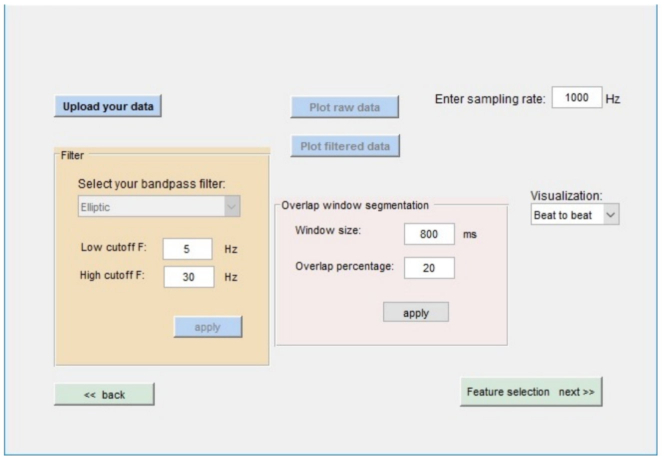


Fig. 1. First page of GUI to select physiologocal signal type



Enter sampling rate: 1000 Hz

Select your bandpass filter: Elliptic

Low cutoff F: 5 Hz

High cutoff F: 30 Hz

Window size: 800 ms

Overlap percentage: 20

Visualization: Beat to beat

Fig. 2. Example of second page of GUI specialized for preprocessing of ECG signal.

Athena Nouhi and Sarah Ostadabbas are with the Augmented Cognition laboratory, Electrical and Computer Engineering Department, Northeastern University, MA, USA (corresponding author's e-mail: ostadabbas@ece.neu.edu).

This work was supported by funding from MathWorks.

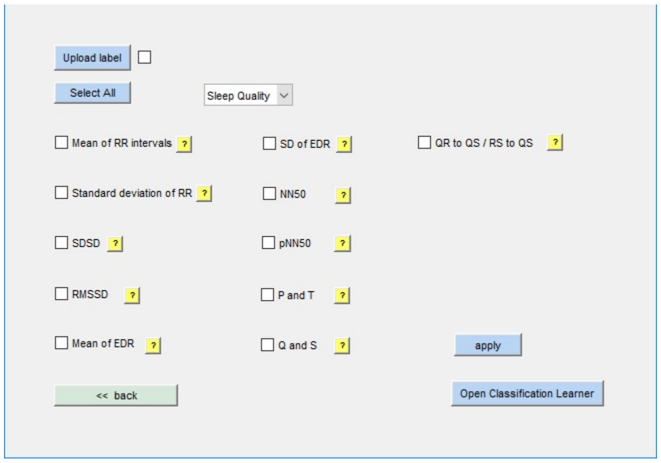


Fig. 3. Example of third page of GUI specialized for feature extraction from ECG signal.

B. Feature Extraction Page

In the third page, you can choose features among the variety of options which are provided with explanation and citation. You can also select a specific application and settle on the default options for that particular application (which the features are chosen based on the recent high impact studies in the field). For classification purposes, you can upload the class labels of your dataset by clicking on the *Upload Labels* button.

By clicking *Apply* button the feature matrix with the name of ”Feature-Matrix” will be created in your MATLAB workspace. If the box near *Upload Labels* button is checked, the feature matrix includes the label as its last column. You can continue to the classification stage by clicking *Classification Learner App* button which opens up the MathWorks learner app and you will be able to use your created feature matrix and perform multiple classification tasks. The feature extraction page for ECG signal is shown in Fig. 3.

II. Biosignal #1: Electrocardiogram (ECG)

ECG is recording of electrical activity of the heart and contains many information such as human health situation, sleep quality and emotional states. ECG cycle created by each heart beat contains P, QRS and T waves shown in Fig. 4. One of the most important challenges in processing ECG signals is robust detection of different waves and intervals. During the code development, a portion of the ECG dataset was collected using BITalino data acquisition board with sampling frequency of 1000Hz. BITalino is a hardware and software toolkit that has been specifically designed to collect body signals [1]. We also employed a couple of datasets from PhisyoBank ATM website for algorithm testing and verification purposes [2].

A. ECG Preprocessing

1) Noise Removal: As mentioned in Section I-A, we have provided three filter type options in the filtering menu. With a thorough analysis of the signal power spectral density (PSD), we found out that a band-pass Elliptic filter with 5-30Hz

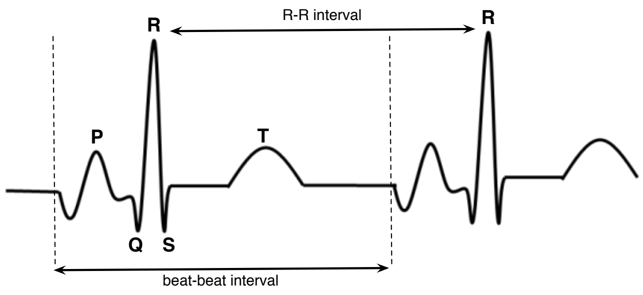


Fig. 4. P, QRS and T waves and intervals in one heartbeat cycle.

cutoffs has the best performance on ECG signal to remove baseline and high frequency noise. So we assigned this filter as the default option in the filter menu. These default filter parameters are different from the filter characteristics used in the well-known Pan-Tompkins algorithm which has been our main reference for the QRS detection [3]. We noticed that the filters used in [3] lower the amplitude of R peaks and as a result reduces its recognition apparency from the T and P peaks. Consequently, they had to apply more steps in order to recognize R peaks from P and T peaks. By trying a couple of different filters and cutoff frequencies, we figured out that Elliptic and Gaussian filters are performing very promising on ECG without decreasing amplitude of R wave significantly. Butterworth filter has also been used in many literature; As a result it is one of the options provided in the filter menu. In Fig. 5, both raw ECG signal and band-pass filtered signal using default Elliptic filter are displayed.

2) QRS detection: Pan-Tompkins algorithm is one of the most popular methods to detect R peaks in ECG signal [3]. Inspired by the Pan-Tompkins algorithm, we have modified the detection algorithm to reflect a more robust QRS detection. QRS detection based on our modified algorithm is implemented by execution of the following function:

```
function [qrs_amp_raw,qrs_i_raw,delay]=
        pan_tompkins_revised(ecg,fs,gr)
```

This function detects QRS peaks in four steps described as follows:

- *Step i*:

  ```
  function [peakI,peakAmp]=
          findQRSpeaks(ecg,win)
  ```

  After filtering the signal, a moving average window is applied to find local maxima with distance of at least 200ms. According to [3] and considering that maximum human heart rate at 300 per minute, two adjacent QRSs cannot happen more closely than 200ms physiologically. The length of window is 400 ms and the center of window is aligned on each sample. Then, the maximum value under this window is found. If this point is in the center of the window, we will pick it as desired local maximum, otherwise the window moves to the next

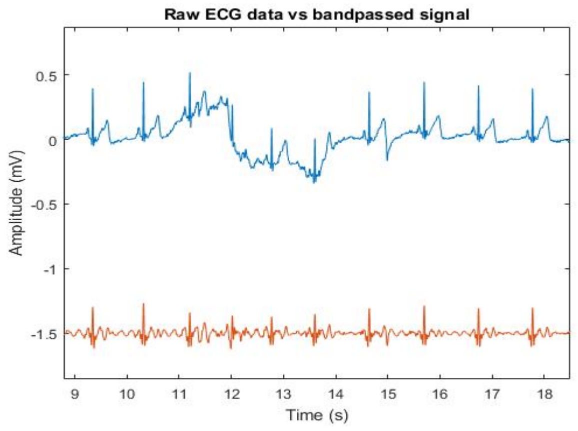


Fig. 5. (upper) Raw ECG signal; (lower) bandpass filtered ECG signal.

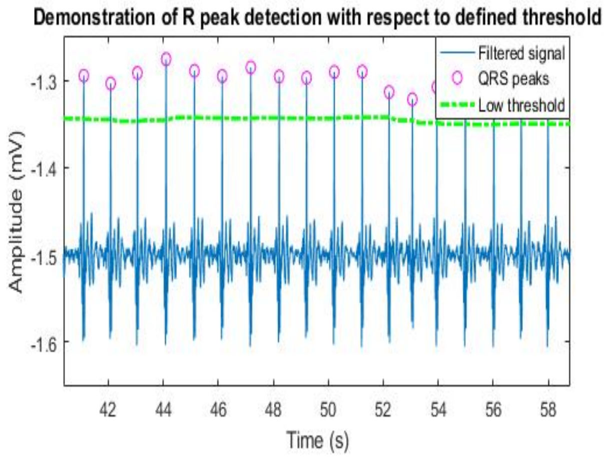


Fig. 6. R peak detection on bandpass filtered signal.

point. In this way we have selected the peaks which are potential to be R peaks and denote ith peak by P_i. This method is also different from the way [3] finds local maxima. You can see detected R peaks in Fig. 6.

- *Step ii*: There are two thresholds that we used to detect actual R peaks from detected R peaks in *step i*. One is amplitude threshold (Amp-Thr) which is obtained in a different way than [3] and the other one is RR interval threshold (RR-Thr) applied in [1]. Amp-Thr is initialized by 1/3 times maximum value of the first 2 seconds of the signal. We did not use other thresholds such as noise threshold applied in [3]. When an R wave is found, Amp-Thr equals to 0.75 times mean of amplitude of last 8 determined R peaks. If the number of determined R peaks up to that point was less than 8, the threshold is mean of amplitude of all determined R peaks. RR-Thr equals to mean of last 8 RR-intervals. We calculate current RR-interval. $RR_i = P_i - P_{i-1}$;
- *Step iii*: If current RR_i is greater than 1.66 times Amp-Thr, then an R peak is missing. We will look for the maximum value between $R_i + 0.2 \times f_s$ and $P_i - 0.2 \times f_s$ and pick it as R peak [3].
- *Step iv*: If P_i is greater than Amp-Thr, we will pick it as an R peak.

3) Q and S point detection: To detect Q, S, P, and T we need to have a smoother signal. Therefore, we used a different filtering method for these detections. To remove baseline drift, we applied a high-pass Elliptic filter with cutoff frequency of 5Hz to the raw signal. Then, we applied a low-pass Gaussian filter. The coefficient for this filter is driven by this formula:

$$w(n) = e^{-1/2[(\alpha n/((N-1)/2))]^2} \quad (1)$$

in which $N = 35$ (number of points) and $\alpha = 6$ (inversely proportional to reciprocal standard deviation). Q_i and S_i are detected for each R_i as:

```
function [Q_i , S_i]=
    find_Q_S(signal,QRS_array_i,wq,ws)
```

where wq and ws are two window sizes before and after R_i which are considered to be 70ms corresponding to 70 samples for $f_s = 1000Hz$.

Search for Q_i is performed in the the interval of R_i and $R_i - wq$. Q_i is the first closest local minimum before R_i in the interval of R_i and $R_i - wq$. If no local minimum was found, we consider the absolute minimum in that interval as Q_i.

Search for S_i is performed in the the interval of R_i and $R_i + ws$. S_i is the first closest local minimum after R_i in the interval of R_i and $R_i + ws$. If no local minimum was found, we consider the absolute minimum in that interval as S_i.

```
function [P_i, T_i] =
find_P_T(signal,Q_i,S_i,PLength, TLength)
```

$PLength$ is maximum length for P wave before Q and $TLength$ is maximum length for T wave after S. We considered $PLength$ and $TLength$ 120ms and 300ms respectively which with $f_s = 1000Hz$ would be 120 and 300 samples. T_i is the absolute maximum found in the interval of R_i to $R_i - PLength$. Since minimum length of ST segment is 80ms, T_i is the absolute maximum found in the interval of R_i to $R_i + TLength$. The detected locations of R, P, T, Q, and S in a few heartbeats are shown in Fig. 7.

5) Segmentation: Segmentation is an optional step in ECG preprocessing which is performed by defining a fixed size window with overlap. You can select the size of window (in ms) and overlap percentage. This segmentation is applied on each dataset row. The last segments in each row will be discarded to avoid segment size inequality. If you choose to not do the segmentation, each row of your dataset will be considered as one segment. Later, feature extraction will be performed on each segment.

6) Segmentation Visualization: The second page in the GUI for ECG signals gives you the option of visualizing the single signal or selected dataset row, segmented based on two methods: *beat to beat* and *R to R* . In *beat to beat* visualization, signal is partitioned beat by beat and each

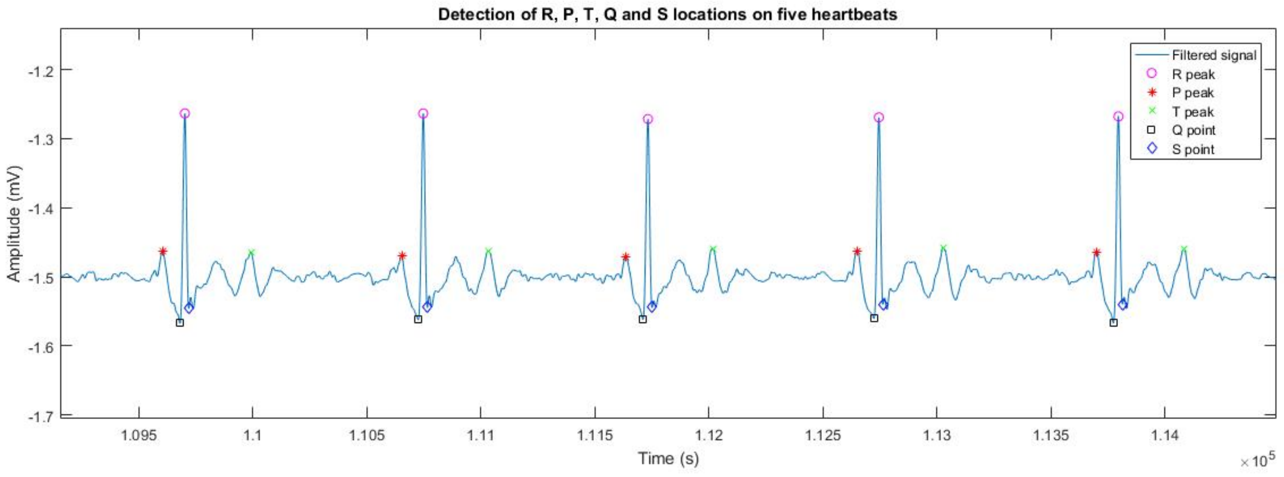


Fig. 7. Detection of R, P, T, Q and S locations in five heartbeats.

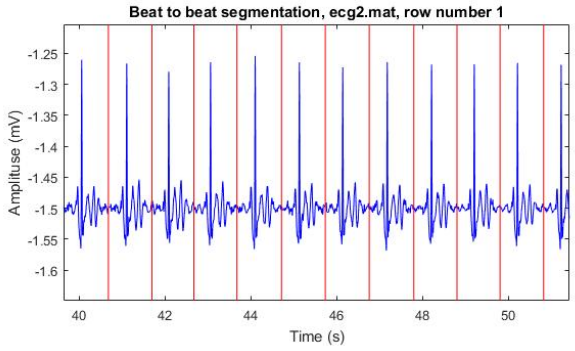


Fig. 8. Beat to beat segmentation visualization of the ECG signal based on P and T waves detection.

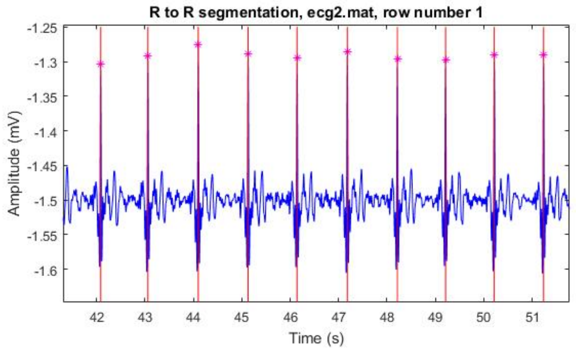


Fig. 9. R to R segmentation visualization of the ECG signal based on QRS wave detection.

partition contains a whole heartbeat, containing P, QRS, and T waves. For this purpose, P and T points are detected for each beat by functions described in Section II-A.4. Beginning of each partition is the middle point between previous T and current P. End of each partition is the middle point between current T and next P point. You can see beat to beat visualization in Fig. 8. In *R to R* visualization, each partition is from one R to the next R. R point detection is described in Section II-A.2. Fig. 9 shows the *R to R* visualization.

B. ECG Feature Extraction

```
function RR_mean =
         find_mean_RR(QRS_array_i)
```

This feature is calculation of mean of all detected RR intervals in a segment [4].

```
function SD_RR =
         find_SD_RR(QRS_array_i)
```

This feature is calculation of standard deviation of all detected RR intervals in a segment [4].

```
function [SDSD_RR_array]=
         find_SDSD(QRS_array_i)
```

SDSD measure is defined as the standard deviation of the differences between adjacent RR intervals in a segment [4].

$$SDSD = 1/N\sqrt{\sum_{i=2}^{N}(RR_i - RR_{i-1} - \mu)^2} \quad (2)$$

where

$$\mu = (\sum_{i=2}^{N}(RR_i - RR_{i-1}))/(N-1) \quad (3)$$

and N = number of RR intervals.

```
function RMSSD =
         find_RMSSD(QRS_array_i)
```

RMSSD measure is defined as the square root of the mean of the sum of the squares of differences between adjacent RR intervals in a segment [5].

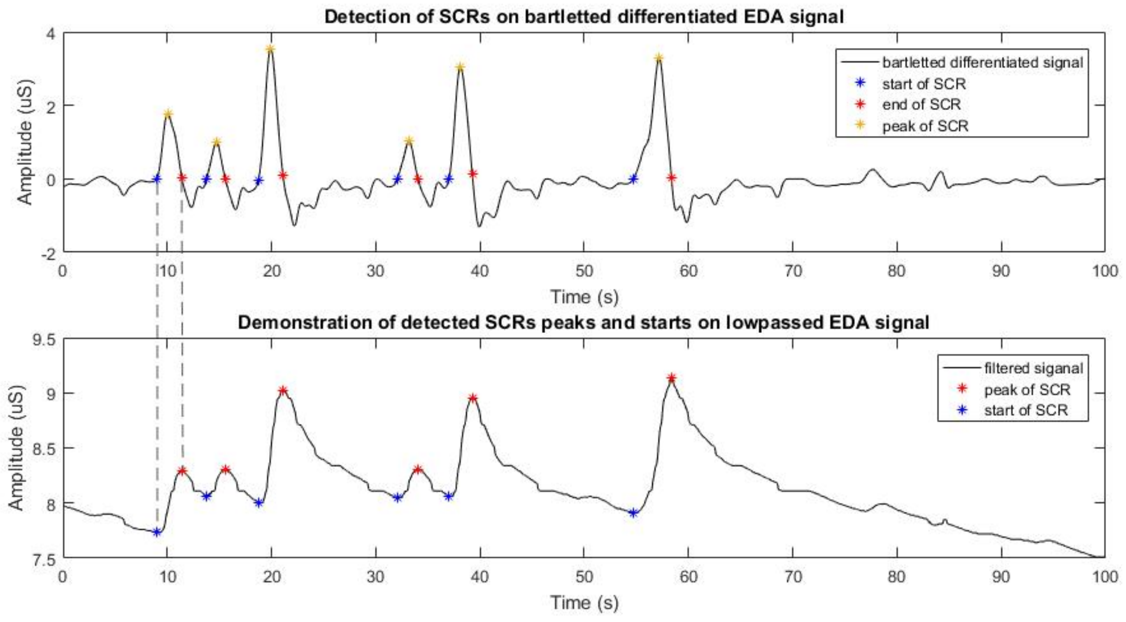


Fig. 10. Detection of SCRs on bartletted differentiated EDA signal (top) and demonstration of detected SCRs' peaks and starts on low-passed EDA signal (bottom).

$$RMSSD = \sqrt{(\sum_{i=2}^{N}(RR_i - RR_{i-1})^2)/(N-1)} \quad (4)$$

```
function NN50 =
find_NN50(QRS_array_i, criterion_ms,fs)
```

NN50 measure is defined as the number of pairs of adjacent RR intervals where the first RR interval exceeds the second RR interval by more than 50ms [4].

```
function pNN50 =
find_pNN50(QRS_array_i, criterion_ms,fs)
```

pNN50 measure is defined as the number of pairs of adjacent RR intervals where the second RR interval exceeds the first RR interval by more than 50ms [4].

```
function EDR =
find_EDR(QRS_array_i, signal,w,fs)
```

EDR is calculated based on the area of each normal QRS complex measured over a fixed window width (which is determined by the interval from the PQ junction to the J-point of a normal QRS). w is the width of moving window. We estimated w with two times sum of QR interval and RS interval ($w = 2 \times (QR + RS)$). The center of the window is aligned over each R peak to calculate the area under it. This feature is calculation of mean of EDR.

8) SD of EDR: This features is calculation of standard deviation of EDR.

9) P and T peak locations: The methods to extract these two features are explained in Section II-A.4.

10) Q and S point locations: The methods to extract these two features are described in Section II-A.3.

```
function [QRtoQS,RStoQS] =
         find_QRtoQS_RStoQS(Q,R,S)
```

To find these two features, we calculate the ratio of QR interval to QS interval and also the ratio of RS interval to QS interval for each R peak.

III. Biosignal #2: Electrodermal Activity (EDA)

Electrodermal activity (EDA) is about momentary or long term changes in skin electrical conductivity because of various internal or external stimuli. Skin conductivity changes with sweat gland activity and sweat glands are activated with sympathetic nervous system. The EDA signal is composed of two activities, tonic and phasic. The slowly varying base signal is the tonic part, also called the skin conductance level (SCL). The faster changing part is called phasic activity or skin conductance response (SCR) which is related to exterior stimuli or non-specific activation and is the bumps that appear in the signal [6]. Because of the relation of skin conductance variations to the sympathetic nervous system, EDA signal has been widely studies in emotion detection research. Many important features for this purpose are extracted from SCRs. As a result, it is important to detect SCRs appropriately.

A. *EDA preprocessing*

Preprocessing of EDA consists of two important stages: noise removal and SCR detection.

1) Noise removal: Filter options of filtering menu for EDA is the same as ECG preprocessing. During our code development phase, we used a couple of datasets shared by MIT Effective Computing group [7]. Performing several trials revealed that Gaussian filter have the best performance among other options. As a result, we assigned Gaussian as the default option in EDA filter menu.

```
function find_SCR(filtered,fs,rowNumber)
```

In this function, inputs are the low-pass filtered signal, sampling rate and dataset row number (in case our data is a dataset), respectively. To detect SCRs, we used the method presented in [8]. After differentiation, the filtered signal is convolved by a 20 point Bartlett window. Bartlett window is a triangle function represented in (5). The occurrence of the SCR is detected by finding two consecutive zero-crossings, from negative to positive and positive to negative. We considered negative to positive as the beginning and positive to negative as the end of an SCR. The amplitude of the SCR is obtained by finding the maximum value between these two zero-crossings. Detected SCRs with an amplitude smaller that 10 percent of the maximum SCR amplitudes already detected in that signal are excluded [8]. Fig. 10 demonstrates the detection of SCRs on a differentiated signal. On the top plot, you can see detected SCR's beginnings, ends, and peaks. In the bottom plot, you can see the original low-pass filtered signal and the location of the previously detected beginning and end points which on the original signal are the beginning and peak locations of the SCRs, respectively.

$$w(n) = \begin{cases} \frac{2n}{N}, & 0 \leq n \leq N/2 \\ 2 - \frac{2n}{N}, & N/2 < n \leq N \end{cases} \tag{5}$$

B. *EDA Feature Extraction*

1) SCR duration mean: This feature is the average of SRC durations. Duration of an SCR is defined as distance from location of the beginning to the end of an SCR.

2) SCR amplitude mean: This feature is the average of the amplitude of the detected SCRs.

3) SCR rise-time mean: This feature is the average of the SCR rise-times. Rise-time of an SCR is defined as time distance between the beginning to the peak of an SCR.

4) Signal mean: This feature is average of the low-pass filtered signal.

5) Number of detected SCRs: This feature is number of the detected SCRs in the EDA signal.

IV. Biosignal #3: Electromyography (EMG)

Electromyography (EMG) is a diagnostic procedure to determine the health of muscles and the nerve cells that control them (motor neurons) by recording and evaluating the electrical activity produced by skeletal muscles. EMG signals are also widely used in the emotion classification studies as a means to create a taxonomy of facial expression []. EMG measures minute changes in the electrical activity of muscles, which reflects minute muscle movements [9].

A. *EMG preprocessing*

1) Noise removal: The type of filter options in filtering menu for EMG is the same as ECG and EDA preprocessing. However, the filters here are not band-pass, but low-pass. Due to the importance of the baseline in the EMG signal, we did not include any high-pass filter option to keep the DC baseline of the signal. After the PSD inspection of the EMG signals, an Elliptic filter with cutoff frequency of 10Hz was considered as the default filter in the filter menu.

2) Segmentation: Segmentation is an optional step for EMG preprocessing which is performed by defining a fixed size window with overlap. The details of this step are similar to the ECG signal segmentation which is explained in Section II-A.5.

B. *EMG Feature Extraction*

We selected the feature options for EDA based on the features introduced in [10].

```
function MAV=find_MAV(signal,L)
```

This feature is the mean absolute value of the EMG signal in an analysis time window with N samples. x_k is the kth sample in this analysis window [10].

$$MAV = \frac{1}{N}\sum_{k=1}^{N}|x_k| \tag{6}$$

```
function count
   = find_zerocerossing(signal,epsilon)
```

Zero crossing count is the numbers of time the EDA signal crosses zero within an analysis time window; it is a simple measure associated with the frequency of the signal. To avoid signal crossing counts due to low-level noise, a threshold ϵ is included the counting ($\epsilon = 0.015V$) [10]. The zero crossing count increases by one if $|x_k - x_{k+1}| \geq \epsilon$, and:
($x_k > 0$ and $x_{k+1} < 0$)
or
($x_k < 0$ and $x_{k+1} > 0$).

```
function count
              = slopSign(signal,epsilon)
```

Slope sign change is related to the signal frequency and is defined as the number of times that the slope of the EMG waveform changes sign within an analysis window. A count threshold ϵ is used to reduce noise-induced counts ($\epsilon = 0.015V$) [10]. The slope sign count increases by one if $|x_k - x_{k+1}| \geq \epsilon$ or$|x_k - x_{k-1}| \geq \epsilon$, and:

$(x_k > x_{k-1}$ and $x_k > x_{k+1})$
or
$(x_k < x_{k-1}$ and $x_k < x_{k+1})$.

```
function find_waveLen
         = waveform_length(signal)
```

This feature provides a measure of the complexity of the signal. It is defined as the cumulative length of the EMG signal within the analysis window [10]:

$$waveLen = \sum_{k=1}^{N} |\Delta x_k| \tag{7}$$

where $\Delta x_k = x_k - x_{k-1}$.

```
function find_logDetect
            = log_detector(signal)
```

This feature is an estimate of the exerted muscle force. The nonlinear detector is characterized as $\log(|x_k|)$ and the log detect feature is defined as:

$$logDetect = e^{\frac{1}{N}\sum_{k=1}^{N} \log |x_k|} \tag{8}$$

6) Standard deviation: This feature is standard deviation of the EMG low-passed signal using the filter selected in the EMG preprocessing page.

7) RMS: This feature is calculation of root mean square of the EMG low-passed signal using the filter selected in the EMG preprocessing page.

V. MathWorks Classification Learner app

In the main page of MATLAB software, on the Apps tab, under Math, Statistics and Optimization, you can find the icon of the Classification Learner app. When classification learner app is opened, you need to create a new session. You can find *New Session* tab on top left of the first page. By clicking on *New Session* you can select your feature matrix from the Workspace or a file. The feature matrix created by our GUI exists in the Workspace. In the next page, you can see three steps. In *Step 1*, you select the ”Feature-Matrix”. In *Step 2* if your ”Feature-Matrix” contains labels, you need to select the last column as *Response*. In *Step 3*, you can select the validation method. By clicking on the *Start Session* you will be directed to the next page that enables several classification algorithms.

VI. Conclusion

In this project, we have created a biosignal-specific processing software in MATLAB by providing the state-of-the-art preprocessing algorithms for each type of signals and then extracting application-relevant features to be selected for creating feature matrix. The resultant feature matrix can be used in *Classification Learner App*.

References

[1] BITalino, *http://www.bitalino.com*, 2016.

[2] P. ATM, *https://physionet.org/cgi-bin/atm/ATM*, 2016.

[3] J. Pan and W. J. Tompkins, “A real-time qrs detection algorithm,” *Biomedical Engineering, IEEE Transactions on*, no. 3, pp. 230–236, 1985.

[4] M. Bsoul, H. Minn, M. Nourani, G. Gupta, and L. Tamil, “Real-time sleep quality assessment using single-lead ecg and multi-stage svm classifier,” in *Engineering in Medicine and Biology Society (EMBC), 2010 Annual International Conference of the IEEE*. IEEE, 2010, pp. 1178–1181.

[5] P. De Chazal, C. Heneghan, E. Sheridan, R. Reilly, P. Nolan, and M. O’Malley, “Automated processing of the single-lead electrocardiogram for the detection of obstructive sleep apnoea,” *Biomedical Engineering, IEEE Transactions on*, vol. 50, no. 6, pp. 686–696, 2003.

[6] H. Gamboa and A. Fred, “An electrodermal activity psychophysiologic model,” 2005.

[7] MIT Effective Computing Group, *http://affect.media.mit.edu*, 2016.

[8] K. H. Kim, S. Bang, and S. Kim, “Emotion recognition system using short-term monitoring of physiological signals,” *Medical and biological engineering and computing*, vol. 42, no. 3, pp. 419–427, 2004.

[9] J. Benedek and R. Hazlett, “Incorporating facial emg emotion measures as feedback in the software design process,” *Proc. Human Computer Interaction Consortium*, 2005.

[10] D. Tkach, H. Huang, and T. A. Kuiken, “Study of stability of time-domain features for electromyographic pattern recognition,” *Journal of neuroengineering and rehabilitation*, vol. 7, no. 1, p. 1, 2010.