Laboratory Exercises and Projects on Biomedical Signal Analysis

Professor Rangaraj M. Rangayyan

Department of Electrical and Computer Engineering
Schulich School of Engineering
University of Calgary
Calgary, Alberta, Canada

`http://people.ucalgary.ca/~ranga/enel563`

Textbook: R. M. Rangayyan, "*Biomedical Signal Analysis,*" Second Edition, IEEE Press and Wiley, New York, NY. 2015.

Signal data files: Available for free downloading at

`http://people.ucalgary.ca/~ranga/enel563/SIGNAL_DATA_FILES`

Laboratory Exercise: Matlab-based computer laboratory exercise for two to three hours each.

Project: Biomedical signal analysis term project for about three months each.

Laboratory exercises and projects may be performed in groups of two students each.

Acknowledgments: I thank Sridhar Krishnan, Naga Mudigonda, Fábio José Ayres, Shantanu Banik, Faraz Oloumi, and Yunfeng Wu for assistance in developing the laboratory exercises and projects. Additional acknowledgments are provided in the textbook.

Contents

1 Laboratory Exercise: Acquisition of ECG and EMG Signals

1.1 Objectives

- Acquire the electrocardiogram (ECG) with and without noise at different heart rates.
- Acquire the electromyogram (EMG) at different levels of muscular contraction (force).

You will be using the signals acquired in subsequent lab exercises. You may share the signals among students within your group.

Please bring a memory device to save your signal data.

The instructions may need to be modified according to the equipment and materials available.

1.2 Acquisition of ECG signals

Please ensure that the signal acquisition hardware is turned OFF before connecting or disconnecting any device. The lab staff will help you identify the devices.

Prepare your experimental subject and acquire ECG signals as follows:

1. Apply a small amount of "Nuprep" gel (abrasive skin prepping gel) to the skin where the electrode will be placed: use the flat, inner parts of the left wrist, the right wrist, and the right ankle. Rub the gel in a circular motion and remove it with a tissue napkin.
2. Apply a small amount of "Signa Cream" (electrode cream) to the same areas. Rub the cream in a circular motion; do not remove the cream.
3. Apply an electrode to each prepared area as follows: black lead (positive) on the left arm; white lead (negative) on the right arm; green lead (reference or ground) on the right leg. This will give you Lead I of the standard ECG (see Section 1.2.5 and Figure 1.30 on p31 of the textbook).
4. Attach the appropriate wire (color coded) to each electrode and connect the ECG cable to the system.
5. Turn on the "Powerlab" system and the "Chart 5" software package on the computer. Select Chanel 3 from the "Settings" menu. Set the "Bioamp" amplifier, filter, and tracing options as follows: amplitude range 5 mV; lowpass 100 Hz; highpass 0.1 Hz; sampling rate 200 Hz; notch filter ON; chart speed 10:1. Please confirm your experimental setup with the lab instructor before proceeding to the next step.
6. Press "Start" to activate signal acquisition.
7. Use the "Auto-scale" feature (on the left-hand side of the screen) to calibrate, if needed. Repeat as necessary until the signal appears to be stable.
8. With the subject remaining steady, start a new recording session. Use a new file name before starting a new session. Record a stable ECG signal for two minutes. Press "Stop" to end signal acquisition. Use the "Save As..." feature to export your data in the "Chart Data (*.txt)" format; use an appropriate file name.

9. Change the notch filter setting to OFF. Get your subject to introduce some noise into the signal by moving one or both arms, or clenching and releasing his/her fists. Remove one of the electrodes, remove some of the gel, replace the electrode, and observe the noise in the signal. Start a new recording session with a new file name and record the noisy ECG signal for two minutes. Press “Stop” to end signal acquisition. Use the “Save As...” feature to export your data in the “Chart Data (*.txt)” format; use an appropriate file name.

10. Disconnect the electrodes from your subject. Ask the subject run or walk rapidly up and down the stairs nearby, or do some vigorous exercise so as to increase the heart rate. Reconnect the electrodes, start a new recording session with a new file name, and acquire the ECG signal for two minutes. Press “Stop” to end signal acquisition. Use the “Save As...” feature to export your data in the “Chart Data (*.txt)” format; use an appropriate file name.

You should acquire and save three ECG data files: a clean signal, a noisy signal at a certain heart rate, and another noisy signal at a higher heart rate. Each file should also contain the sampled time instant for each sample of the ECG. Verify that the filter and other signal acquisition parameters used are as specified above. Ensure that there are no “NaN” (Not a Number) entries in your data files. Save the data files on to your own memory device for use in subsequent lab exercises.

1.3 Acquisition of EMG signals

Please ensure that the signal acquisition hardware is turned OFF before connecting or disconnecting any device. The lab staff will help you identify the devices.

Prepare your experimental subject and acquire EMG signals as follows:

1. Connect the force transducer to Chanel 2. Ask your experimental subject to grip the force transducer with his/her dominant hand. Identify the extent of the muscle on the inner forearm that contracts. Identify two areas to place the electrodes: one about 5–6 cm from the wrist and the other 6–8 cm farther along the same muscle (toward the elbow). Apply a small amount of “Nuprep” gel to the skin where the electrodes will be placed. Rub the gel in a circular motion and remove it with a tissue. Apply a small amount of “Signa Cream” to the same areas. Rub the cream in a circular motion; do not remove the cream.

2. Using the ECG cable as in the previous experiment, connect the electrodes provided to acquire the EMG as follows: black lead (positive) on the inside of the arm, about 5–6 cm from the wrist; white lead (negative) 6–8 cm farther along the same muscle (toward the elbow). Attach the appropriate wire to each electrode.

3. Set the “Bioamp” amplifier, filter, and tracing options for the EMG signal on Channel 3 as follows: amplitude 5 mV; lowpass 1 kHz; highpass 10 Hz; sampling rate 2 kHz; notch filter ON; chart speed 20:1.

4. Set the input amplifier and filter options for the force signal on Channel 2 as follows: amplitude 50 mV; lowpass 1 Hz; AC coupling OFF; single-sided ON.

5. Press “Start” to activate signal acquisition. Ask your experimental subject to squeeze the force transducer to the maximum extent possible. Note the level of the force output; the

output is not calibrated in units of force. Ask the subject to relax the muscle for a few seconds, and squeeze the transducer in five steps, corresponding to approximately 10%, 30%, 50%, 70%, and 100% of the maximum, each lasting about two seconds, with resting periods of two seconds between each squeeze. Practice this sequence 2–3 times so that the entire sequence can be repeated with uniform force output at each level and the entire sequence can be completed in about 20 seconds. Use the "Auto-scale" feature (on the left-hand side of the screen for each channel) to calibrate, as necessary.

6. Ask your subject to relax and rest for a minute so as to prevent the effects of fatigue in the muscle being exercised. When the subject is ready to repeat the experiment, start a new recording session and record the full sequence of the EMG and force signals at five levels of contraction as above. Press "Stop" to end signal acquisition. Verify that the signals are of good quality and have the desired features. Use the "Save As..." feature to export your data in the "Chart Data (*.txt)" format; use an appropriate file name.

You should acquire and save one data file with two channels, force and EMG, at five levels of contraction with rest periods in between. Each file should also contain the sampled time instant for each sample of the EMG. Verify that the filter and other signal acquisition parameters used are as specified above. Ensure that there are no "NaN" entries in your data files. Save the data file on to your own memory device for use in subsequent lab exercises.

2 Laboratory Exercise: Matlab Practice

2.1 Objectives

- To become familiar with the basics of Matlab programming.
- To learn basic signal processing concepts: convolution and peak detection.

Study the document
`http://people.ucalgary.ca/~ranga/enel563/matlab_tutorial.pdf`

2.2 Convolution-A

Let $x(n)$ and $h(n)$ be two discrete-time signals, as follows:

$$\begin{aligned} x(n) &= 2\delta(n-1) + 3\delta(n-2) + 4\delta(n-3) - 5\delta(n-4), \\ h(n) &= [\delta(n) + \delta(n-1) + \delta(n-2)]/3. \end{aligned} \tag{1}$$

1. Write a Matlab program to represent the signals $x(n)$ and $h(n)$ as vectors, and plot the signals.
2. In your Matlab program, compute $y(n) = x(n) * h(n)$ using the command `conv`. Plot the input signals and the resulting signal in a single figure using `subplot` as required; use the `stem` command to obtain stem plots. Ensure that all subplots have the same limits on their axes.
3. In your lab report, describe the nature and effect of the convolution kernel $h(n)$.

2.3 Convolution-B

Let $x_1(t)$ and $x_2(t)$ be two continuous-time signals, as follows:

$$\begin{aligned} x_1(t) &= 2[u(t) - u(t-2)], \\ x_2(t) &= 5[u(t) - u(t-3)]. \end{aligned} \tag{2}$$

1. Prepare and plot the signals in a Matlab program. Let the sampling frequency be $f_s = 200$ Hz and let the time axis of the plots for this exercise run from 0 to 10 seconds.
2. In your Matlab program, compute the result of the convolution of the two signals and plot the result.
3. Explain the shape of the output signal.

2.4 Peak detection

Consider the following signal:

$$x(t) = 5\sin(2\pi 5t) + 3\cos(2\pi 7t)\ .$$

Write a Matlab program to do the following:

1. Generate a sampled version of the signal at the sampling frequency $f_s = 100$ Hz for the duration of $0 - 10$ seconds.
2. Write a Matlab function to find all the peaks and troughs or valleys (positive and negative) in the sampled signal. A sample represents a peak or trough (or valley) if the preceding and the subsequent samples are lower in magnitude than the sample under consideration. The function should receive the sampled signal, and return a vector containing the indices of the peaks in the signal. Plot the signal and mark the peaks using the following code:

   ```
   figure;
   plot(t, x, ’k-’); hold on;
   plot(t(peaks), x(peaks), ’k*’); hold off;
   xlabel(’time (s)’);
   ylabel(’x(t)’);
   ```

   where `peaks` is the vector of the indices of the peaks detected.
3. Modify the procedure described above to mark the troughs (valleys or local minima) in the signal with the ’o’ symbol.

2.5 Plot your ECG signals

1. Write a Matlab program to extract a window of data of duration 10 seconds from each of the three ECG signals you acquired in Lab 1, and plot the three extracted signals using `subplot`. Calibrate and label the axes appropriately. Compare the ECG signals and describe your observations in your report.

   You may also use `ecg_hfn.dat`, `ecg_hfn.m`, `ecg_lfn.dat`, `ecg_lfn.m`, `ECG3.dat`, `ECG4.dat`, `ECG5.dat`, `ECG6.dat`, and `ECGS.m`.
2. Prepare an additional expanded plot of a segment of the clean ECG signal you acquired so as to show two full cardiac cycles. Calibrate and label the axes appropriately.
3. Apply the peak detection algorithm you developed for the exercise in Section 2.4 to the ECG signal segment with two cardiac cycles, as described above, and plot the results. Write your observations and comments on the results.

2.6 Plot your EMG/force signals

1. Write a Matlab program to read the EMG/force data you acquired in Lab 1, and plot the entire data in a two-channel plot. Calibrate and label the axes appropriately. Describe the nature of the signal in your report.

   You may also use the data files

   `EMGforce.txt`, `EMGforce2.txt`, `EM_EMG_SQUEEZE1.txt`, and `EM_EMG_SQUEEZE2.txt` as well as the program `EMGforce.m`. The sampling rate is 2000 Hz per channel and the EMG sample values are in mV.

2. Prepare an additional expanded two-channel plot showing a segment of about $100 - 200$ milliseconds with increasing force at the beginning of one of the contractions and the corresponding EMG.

   Describe the nature of the signal in your report.

2.7 Exercises to be worked by hand

1. Write the equation for the convolution of two continuous-time signals.

2. Write the equation for the convolution of two discrete-time signals.

3. Calculate by hand the result of the convolution $y(n) = x(n) * h(n)$ for the signals given in Equation 1. Show all steps of your calculations in your report.

4. Calculate by hand the result of the convolution $y(t) = x_1(t) * x_2(t)$ for the signals given in Equation 2. Show all steps of your calculations in your report.

   Does this result agree with the result of your Matlab program for the related problem in Section 2.3? Describe and explain your findings.

2.8 Matlab commands to learn

Use the `help` command in Matlab and learn the following commands:
`load`, `length`, `conv`, `clc`, `clear all`, `close all`, `sin`, `function`, `figure`, `stem`, `plot`, `subplot`, `axis`, `axis tight`, `axis equal`, `xlabel`, `ylabel`, `title`.

3 Laboratory Exercise: Analysis of the Relationship Between Parameters of the EMG Signal and Muscular Force

3.1 Objectives

- To characterize the level of activity in EMG signals.
- To study the relationship between parameters of the EMG signal and muscular force.

3.2 Background

The EMG signal is the electrical signal associated with muscular contraction. From the EMG signal that you acquired in Lab 1 at various levels of muscular force exerted, it is evident that the level of activity of the EMG signal increases with force; see Section 1.2.4 and Figure 1.20 of the textbook for details on the EMG signal. In this lab exercise, you will explore several measures, features, or parameters derived from EMG signals, and study their relationships to the associated muscular force.

The dynamic range (DR) of a signal is the difference between its maximum and minimum values over a specified duration. This range of values is also referred to as the peak-to-peak swing or range.

The average power of a signal is provided by the mean-squared (MS) value over a specified duration of time; the average magnitude over the same duration is given by the root-mean-squared (RMS) value. With the mean of the EMG signal being zero, the RMS value is equal to the standard deviation of the signal.

An approximate indicator of the level of activity in a signal is given by the number of times the signal crosses the zero line (assuming a mean value of zero; if the mean is not zero, it should be subtracted from the signal). A zero crossing is said to occur when the sign of a sample of the signal is different from the sign of the preceding sample. The average number of zero crossings over a certain period of time is known as the zero-crossing rate (ZCR). It is expected that increased presence of high-frequency components in a signal will lead to larger ZCR. See Section 3.2.1 of the textbook for a discussion on statistical parameters of random signals. See also Sections 5.6 and 5.9 on statistical analysis of EMG signals.

In order to investigate the relationship that may exist between an independent variable (muscular force in the present exercise) and a measured parameter or variable (DR, MS, RMS, or ZCR), it is common practice to fit a mathematical function or curve to the data samples available and examine the accuracy of the underlying model (goodness of fit). In the present lab exercise, you will investigate the appropriateness of linear fits (straight-line or linear models) to represent the variation of each parameter with force.

See `http://mathworld.wolfram.com/CorrelationCoefficient.html` for details on linear least-squares fitting and the correlation coefficient.

3.3 Exercises to be worked by hand

You are given an array of sampled values of a signal $x(n)$, $n = 1, 2, \ldots, N$.

1. Write equations to define the following values:
   - (a) the mean value,
   - (b) the variance,
   - (c) the standard deviation,
   - (d) the MS value, and
   - (e) the RMS value of the signal.

2. Write an equation to transform the signal $x(n)$ to another signal $y(n)$ such that the values of $y(n)$ are in the range 0 to 1.

3.4 Parametric analysis of the EMG

Develop a Matlab program to perform the following analysis:

1. Load the EMG signal that you acquired in Lab 1. (Remove the header information.) Normalize the force signal such that the minimum value is zero and the maximum value (corresponding to the maximum voluntary contraction or MVC) is 100. Plot the EMG signal (in mV) and normalized force (in %MVC) against the time axis in two subplots.

   You may also use the data files

   `EMGforce.txt, EMGforce2.txt, EM_EMG_SQUEEZE1.txt, and EM_EMG_SQUEEZE2.txt` as well as the program `EMGforce.m`. The sampling rate is 2000 Hz per channel and the EMG sample values are in mV.

2. Manually identify portions (segments) corresponding to each level of contraction within which the force remains almost constant. Store the time limits of each segment in your Matlab code and mark the corresponding points on the plot described above.

3. For each segment of the EMG signal identified as above, compute the DR, MS, RMS, and ZCR parameters. Ensure that you normalize ZCR to zero crossings per second by dividing by the time duration (in seconds) of the corresponding segment. Similarly, ensure that the MS and RMS values are computed using the appropriate number of samples for each segment.

4. Plot the DR, MS, RMS, and ZCR values versus force in %MVC. Label the axes with the appropriate units.

5. Using the `polyfit` function in Matlab, obtain a straight-line (linear) fit to represent the variation of each EMG parameter versus force. Use `polyval` to evaluate the values of the dependent variable given by the model for the available values of the independent variable. Superimpose the linear models (straight-line fits) obtained on the plots of the parameters in the preceding step.

6. Compute the correlation coefficient, r, with r^2 given by the formula

$$r^2 = \frac{\left[\sum_{n=1}^{n=N} x(n)\, y(n) - N\, \bar{x}\, \bar{y}\right]^2}{\left[\sum_{n=1}^{n=N} x^2(n) - N\, \bar{x}^2\right]\, \left[\sum_{n=1}^{n=N} y^2(n) - N\, \bar{y}^2\right]}\,, \tag{3}$$

where N is the number of samples of x or y representing the variables DR, MS, RMS, ZCR, or force; $\bar{x}$ is the mean of x. Using r, analyze the goodness of fit for each parameter and discuss the appropriateness of the linear model. See Section 5.9 of the textbook for related examples.

7. Tabulate the parameters of the linear model and r for each of the variables DR, MS, RMS, and ZCR. Analyze the results and describe your findings.

3.5 Matlab commands to learn

Use the `help` command in Matlab and learn the following commands:
`polyfit, polyval, sign`.

4 Laboratory Exercise: Synchronized Averaging for Noise Reduction

4.1 Objectives

- To study the effect of synchronized averaging on noise.
- To study the effect of the number of trials in repeated stimulation for synchronized averaging.

4.2 Background

Linear filters do not perform well when the signal and noise spectra overlap. Synchronized signal averaging can separate a repetitive signal from noise without distorting the signal (see Sections 3.5.1 and 3.11 of the textbook). Epochs of event-related potentials (ERP) and somatosensory evoked potentials (SEP) may be obtained a number of times by repeated application of the stimulus; they may then be averaged by using the stimulus as a trigger to align the epochs. If the noise is random with zero mean and is uncorrelated with the signal, averaging will improve the signal-to-noise ratio (SNR).

Let $y_k(n)$ represent one realization or observation of a signal, with $k = 1, 2, \ldots, M$ representing the ensemble index, and $n = 1, 2, \ldots, N$ representing the time-sample index. Here, M is the number of copies (trials, events, epochs, or realizations) of the signal available, and N is the number of the time samples in each copy of the signal (event). We may express the observed signal as

$$y_k(n) = x_k(n) + \eta_k(n), \tag{4}$$

where $x_k(n)$ represents the original uncorrupted signal and $\eta_k(n)$ represents the noise in the k^{th} copy of the observed signal. Now, if for each instant of time n we average the M copies of the signal, we get

$$\bar{y}(n) = \frac{1}{M} \sum_{k=1}^{M} y_k(n) = \frac{1}{M} \sum_{k=1}^{M} x_k(n) + \frac{1}{M} \sum_{k=1}^{M} \eta_k(n);\ n = 1, 2, \ldots, N. \tag{5}$$

If the repetitions of the signal are identical and aligned, $\sum_{k=1}^{M} x_k(n) = Mx(n)$. If the noise is random, has zero mean, and has a variance of σ_η^2, $\sum_{k=1}^{M} \eta_k(n)$ will tend to zero as M increases, with the variance given by $M\sigma_\eta^2$; its RMS value is $\sqrt{M}\sigma_\eta$. Thus, the SNR of the signal will increase by a factor of $\frac{M}{\sqrt{M}}$ or $\sqrt{M}$. The larger the number of epochs or realizations that are averaged, the better will be the SNR of the result. Note that synchronized averaging is a type of ensemble averaging.

Kamath et al. [1] applied synchronized averaging to improve the SNR of cortical evoked potentials related to electrical and mechanical stimulation of the esophagus. Although improvement in SNR was obtained in some experiments, they also observed that habituation took place as the number of stimuli was increased beyond a certain limit, and that the use of the ERPs obtained after habituation in averaging led to a reduction in the SNR.

Kamath et al. estimated the SNR as follows:

Noise power:

$$\sigma_\eta^2 = \frac{1}{NT(M-1)} \sum_{k=1}^{M} \sum_{n=1}^{N} [y_k(n) - \bar{y}(n)]^2 \,. \tag{6}$$

Signal power:

$$\sigma_{\bar{y}}^2 = \frac{1}{NT} \left\{ \sum_{n=1}^{N} [\bar{y}(n)]^2 \right\} \;-\; \frac{\sigma_\eta^2}{M} \,. \tag{7}$$

$$SNR = \frac{\sigma_{\bar{y}}^2}{\sigma_\eta^2} \,. \tag{8}$$

Here, $T = 0.001\ s$ is the sampling interval.

Kamath et al. also computed the Euclidean distance between the original ERP signals and the averaged signal obtained as

$$D = \frac{1}{M} \sum_{k=1}^{M} \sqrt{\sum_{n=1}^{N} [y_k(n) - \bar{y}(n)]^2} \,. \tag{9}$$

4.3 Exercises to be worked by hand

1. Write the equation to define the mean of a random variable x in terms of its probability density function (PDF) $p_x(x)$.
2. A variable y is given as $y = x + \eta$ where x and η are statistically independent random processes. Starting with the joint PDF of the random processes x and η, derive an expression for the mean of $y = x + \eta$. Show all steps.
3. Write the definition of the variance of x using the expectation operator and the PDF of x.
4. With y defined as in item 2 above, derive the relationship between the variance of y, x, and η using the expectation operator (no need to use PDFs and integrals in this case). Show all steps.

4.4 Averaging of SEPs

Copy the data files `E11` to `E2424` and the Matlab program `esoepX.m`. The signals were filtered to the band $0.1 - 100\ Hz$ and sampled at $1000\ Hz$. The number of samples in each signal is $N = 511$. (Signal data courtesy of Dr. M. V. Kamath, McMaster University, Hamilton, ON, Canada.)

The signals are numbered as Ekk, where k is the trial number, with $k = 1, 2, \ldots, M$, and $M = 24$. The signals are numbered in the time sequence of the stimulation trials.

Write a Matlab function to compute the average of a certain number of signals (trials) to be specified as an input parameter.

Write a Matlab function to compute the SNR and Euclidean distance between a certain averaged signal and the corresponding set of original signals used, as specified in Equations 8 and 9.

Write a Matlab program to compute several averages of selected ERP signals as follows:

1. Trials $1-6$, $7-12$, $13-18$, and $19-24$.
2. Trials $1-8$, $9-16$, and $17-24$.
3. Trials $1-12$ and $13-24$.
4. Trials $1-24$.

For the case with the trials $1-6$, plot each of the signals used as well as the result of averaging using subplots; for the remaining cases, plot only the results of averaging. For all cases, compute the SNR and Euclidean distance values as given above.

Prepare a table of the various SNR and the Euclidean distance values obtained along with the corresponding signals (trials) used and the number of signals averaged (M). Analyze the results and discuss your findings.

4.5 Matlab commands to learn

Use the `help` command in Matlab and learn the following commands:
`sum, sqrt`

Reference: 1. M. V. Kamath, G. Tougas, S. Hollerbach, R. Premji, D. Fitzpatrick, G. Shine, and A. R. M. Upton, "Estimation of habituation and signal-to-noise ratio of cortical evoked potentials to oesophageal electrical and mechanical stimulation," Medical and Biological Engineering and Computing, 35:343-347, 1997.

5 Laboratory Exercise: Frequency-domain Analysis of Biomedical Signals

5.1 Objectives

- To study methods for frequency-domain analysis of biomedical signals.
- To study the frequency content of ECG, EEG, EMG, and PCG signals.

5.2 Background

In this laboratory exercise, you will obtain estimates of the power spectral density (PSD) of various biomedical signals, including the ECG, EEG, EMG, and PCG. In order to obtain an estimate of the PSD of a signal, or a segment of interest of the signal, follow the procedure given below.

1. Select the segment of interest from the signal.
2. Subtract the mean of the segment; this will set the DC component in the PSD to zero.
3. Obtain the Fourier transform (FT) of the segment. The Matlab command for the computation of the FT is `fft` (representing the fast Fourier transform or FFT algorithm).
4. Obtain the squared magnitude of the FT.
5. Normalize the squared magnitude spectrum by dividing by its maximum.
6. Convert the result to dB: take the logarithm to base ten of the result above and multiply by ten. This is an estimate of the PSD of the original signal in dB.
7. Select the first half of the PSD as above. Prepare the corresponding frequency axis in Hz, spanning the range $[0, f_s/2]$, where f_s is the sampling frequency of the signal.
8. Plot and analyze the result. Ensure that the axes are labeled in Hz and dB.

Note: In order to obtain a good estimate of the PSD of a process, several PSD estimates obtained from several observations of the related signal should be averaged. Furthermore, the signal should be multiplied with an appropriate window. However, windowing and averaging are not included in this lab exercise. See Chapter 6 of the textbook for details.

5.3 Exercises to be worked by hand

1. Write an equation to define the FT $X(f)$ of a continuous-time signal $x(t)$.
2. Write an equation to define the discrete Fourier transform (DFT) $X(k)$ of a discrete-time signal $x(n)$.
3. Explain how the frequency f (in Hertz) of a sample of the DFT is related to the index k of the DFT and f_s.

4. The DFT of a discrete-time signal is obtained with N samples. Draw the frequency axis indicating the sample number $k = 1, 2, \ldots, N$, and label the points or parts corresponding to DC, $0.5\, f_s$, f_s, the positive frequency range, and the negative frequency range.

5. Starting from the definition of the autocorrelation function (ACF) of a continuous-time signal, prove that the Fourier transform of the ACF gives the PSD of the signal.

5.4 Function for PSD

Write a Matlab function to compute the PSD (in dB) of a given signal using the procedure explained above.

5.5 PSD of ECG signals

Select one cardiac cycle from your noise-free ECG recording from Lab 1. Obtain and analyze its PSD. Similarly, obtain the PSD of one cardiac cycle from your noisy ECG signal. Compare the two PSDs and discuss their similarities and/or differences. Include in your report a plot of each ECG segment and its PSD.

You may also use `ecg_hfn.dat, ecg_hfn.m, ecg_lfn.dat, ecg_lfn.m`, `ECG3.dat, ECG4.dat, ECG5.dat, ECG6.dat,` and `ECGS.m`.

Repeat the experiment above with a larger number of samples in the FFT with zero padding of the signal; study the details provided in Matlab for the command `fft`. Compare the PSDs and describe your observations.

Repeat the experiment with a segment containing 20 cycles of the ECG in your noise-free ECG recording; let the number of samples in the FFT be equal or to greater than the number of samples in the signal segment. Compare the PSDs obtained with segments containing 20 cycles and one cycle of the same signal, and describe your observations.

5.6 PSD of an EEG signal

Download one of the `eeg1*.dat` files; there are eight channels available. See also `eeg1.m`. The sampling rate is $f_s = 100$ Hz. Obtain and analyze the PSD of the entire EEG signal in the selected channel. Include in your report a plot of the EEG signal selected and the corresponding PSD.

What is the dominant rhythm present in the signal?

5.7 PSD of EMG signals

From the EMG signal you acquired in Lab 1, select one segment for each level of muscular contraction. Obtain and analyze the PSD of each segment. Include in your report plots of each EMG signal segment and the corresponding PSD.

You may also use the data files

EMGforce.txt, EMGforce2.txt, EM_EMG_SQUEEZE1.txt, and EM_EMG_SQUEEZE2.txt as well as the program EMGforce.m. The sampling rate is 2000 Hz per channel and the EMG sample values are in mV.

Are there any major differences between the PSDs at various levels of contraction? Is there a trend in the PSDs against the level of contraction?

5.8 PSD of PCG signals

Using the program pcg3read.m, read the data file pec1.dat. The program gives a plot of three channels: ECG, PCG, and carotid pulse. The sampling rate is $f_s = 1000$ Hz per channel. Select a systolic segment of the PCG signal and obtain its PSD. Select a diastolic segment and obtain its PSD.

Repeat the above with the data in the file pec33.dat. This signal contains systolic murmur. Use the dicrotic notch in the carotid pulse to determine the boundary between the systolic and diastolic segments. Subtract 52 ms from the dicrotic notch position to obtain the initial time instant of the diastolic segment; refer to Sections 2.3 and 4.11 of the textbook for discussions on this topic. Obtain the PSDs of the systolic and diastolic segments.

In each case, include a plot of the PCG segment and its corresponding PSD in your report. Discuss the results obtained.

5.9 Typical bandwidths of biomedical signals

Prepare a table giving the name of each signal analyzed in each experiment conducted and the corresponding bandwidth (in Hz) containing most of the power in the signal. Explain how you determined the bandwidth in each case. Discuss the results in your table.

5.10 Matlab commands to learn

Use the help command in Matlab and learn the following commands:
ginput, fft, log10

6 Laboratory Exercise: Filtering of the ECG for the Removal of the 60 Hz Power-line Artifact

6.1 Objective

- Rejection of 60 Hz power-line interference from ECG signals.

6.2 Background

A major source of interference in ECG signals is the 60 Hz power-line frequency. The frequency component at 60 Hz can be removed by using a notch filter, in which a zero is placed on the unit circle at the location corresponding to 60 Hz.

The notch filter at 60 Hz can be derived as follows. The angular position of the required zero on the unit circle is given as

$$\theta_o = 2\pi \frac{\text{notch filter frequency } f_o}{\text{sampling frequency } f_s}. \tag{10}$$

Complex zeros appear as conjugate pairs, i.e., if we place a zero at $z_1 = real + j\ imag$, we have its conjugate at $z_2 = real - j\ imag$ (see Figure 1).

Because z_1 is on the unit circle, its magnitude is unity. The transfer function $H(z)$ of the notch filter is

$$H(z) = \frac{(z - z_1)(z - z_2)}{z^2} = (1 - z_1\, z^{-1})(1 - z_2\, z^{-1}). \tag{11}$$

Refer to Section 3.6.3 of the textbook for details.

Note: The output of the filter $y(n)$ can be found for an input $x(n)$ by using the `filter` command in Matlab. Pay attention to the significance of z, z^{-1}, and the notation of the filter coefficient array in Matlab! The command `y = filter(b, a, x)` filters the data in the array `x` with the filter described by the arrays `a` and `b` to create the filtered vector `y`. The filter is a "Direct Form II Transposed" implementation of the difference equation

$$y(n) = b(1)x(n) + b(2)x(n-1) + \ldots + b(N+1)x(n-N) - a(2)y(n-1) - \ldots - a(M+1)y(n-M). \tag{12}$$

Observe that $a(1) = 1$. M and N give the order of the filter.

6.3 Exercises to be worked by hand

1. Derive the transfer function for a notch filter with the notch frequency of $f_o = 60$ Hz and the sampling rate of $f_s = 200$ Hz. Include a scale factor such that the gain at DC is unity. Derive the difference equation of the notch filter.

2. What is the effect of the filter designed as above if applied to your EMG signal from Lab 1 with $f_s = 2$ kHz?

3. Draw the signal-flow diagram of the filter.

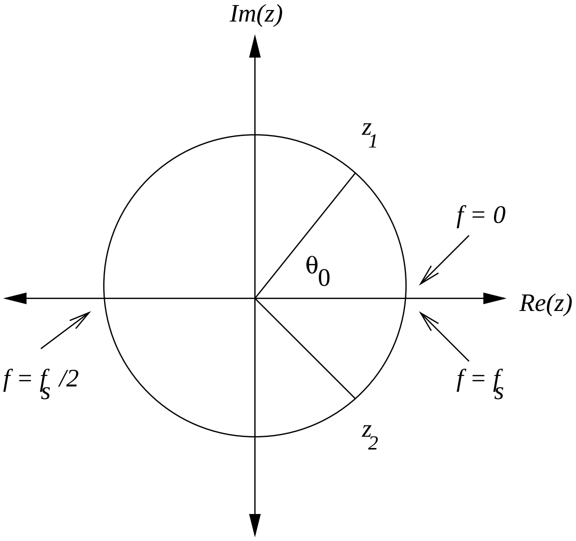

Figure 1: Unit circle in the z plane with the zeros of a notch filter.

6.4 Notch filter for 60 Hz

1. Write a Matlab program to implement the notch filter designed as above.
2. Apply the filter to the signal in the data file `ecg_60hz_200.dat`. Plot the ECG signal before and after filtering. Study the nature of the artifacts in the noisy signal and in the output of the filter.
3. Apply the notch filter to a segment of one of your noisy ECG signals from Lab 1 with $3-5$ cardiac cycles. Plot the ECG signal before and after filtering. Study the nature of the artifacts in the noisy signal and in the output of the filter.
4. Analyze the characteristics and the effects of the filter in the frequency domain by obtaining and plotting the pole–zero diagram and the transfer function (magnitude and phase response) of the filter, as well as the PSDs of the input and output signals.

6.5 Matlab commands to learn

Use the `help` command in Matlab and learn the following commands:
`filter, freqz, zplane, fft`.

7 Laboratory Exercise: Filtering of the ECG for the Removal of Noise

7.1 Objectives

- Removal of high-frequency noise.
- Removal of low-frequency noise (baseline artifact).
- Rejection of power-line interference.

7.2 Background

ECG signals are often contaminated with a combination of high-frequency noise, low-frequency noise (wandering baseline), and power-line frequency artifacts. High-frequency noise may be removed by applying a lowpass filter, and low-frequency noise may be removed by using a highpass filter. However, improper selection of the cutoff frequencies of the two filters could lead to distortions in the ECG signal, such as excessive smoothing or widening of the QRS complex, and distortion of the normally isoelectric PQ and ST segments. Refer to Sections 3.5 and 3.6 of the textbook for details.

The 60 Hz frequency component can be removed by using a notch filter, in which a zero is placed on the unit circle at the location corresponding to 60 Hz. However, if the power-line signal is not a perfect sinusoid, the signal could contain significant components at harmonics of 60 Hz. A comb filter with zeros at each component to be rejected could then be used to remove the artifact. Refer to Section 3.6.3 of the textbook for details.

7.3 Exercises to be worked by hand

1. A filter is specified to have a zero in the z domain at $z = 1$. Derive the impulse response $h_1(n)$ and the transfer function $H_1(z)$ of the filter. Is this a lowpass, highpass, bandpass, or band-reject filter?
2. Another filter is specified to have a double-zero (or two zeros) at $z = -1$. Derive the impulse response $h_2(n)$ and the transfer function $H_2(z)$ of the filter. Is this a lowpass, highpass, bandpass, or band-reject filter?
3. A researcher uses the two filters $H_1(z)$ and $H_2(z)$ as above in cascade (series). Derive the impulse response $h(n)$ and the transfer function $H(z)$ of the combined filter. Is this a lowpass, highpass, bandpass, or band-reject filter?
4. Draw the pole–zero plot for the combined filter.
5. Draw the signal-flow diagram for the combined filter.

7.4 Filtering ECG signals

Filter the noisy ECG signal that you acquired in the first lab as follows. You may also use `ecg_hfn.dat`, `ecg_hfn.m`, `ecg_lfn.dat`, and `ecg_lfn.m`.

1. Apply the von Hann lowpass filter as in Equation 3.99 in the textbook. Specify the filter in terms of the a and b arrays via the `filter` command in Matlab.
2. Modify the derivative-based filter given by Equation 3.131 in the textbook for the removal of low-frequency artifacts so that the gain at the maximum frequency present in the input signal is unity. Use the `filter` command and apply the filter to your signal.
3. Apply the notch filter that you designed in Lab 6 for the rejection of 60 Hz to your signal. Ensure that the filter is normalized to have unit gain at DC. Use the `filter` command.
4. Study the effect of each filter applied individually to the ECG signal, in the time domain by plotting the entire duration of the signal (or a relevant portion of $10 - 15$ seconds), as well as by plotting a part of the signal with $2 - 3$ cycles of the ECG. If you do not get useful results with your ECG signal, you may test your methods using an ECG signal from another group in your class. Comment upon and present an analysis of your results.
5. Study each filter in the frequency domain by obtaining its transfer function (magnitude and phase), pole–zero plot, as well as the Fourier spectra of the input and output signals. Comment upon and present an analysis of your results.
6. Apply all three filters to your ECG signal in series, and study the combined filter and the result as specified above. Comment upon and present an analysis of your results.

Note: Obtain the transfer function (magnitude and phase) of the combination of the three filters by using the `freqz` command. When three filters $H_1(z) = \frac{B_1(z)}{A_1(z)}$, $H_2(z) = \frac{B_2(z)}{A_2(z)}$, and $H_3(z) = \frac{B_3(z)}{A_3(z)}$ are applied in series (cascade), the net transfer function is given by

$$H(z) = H_1(z)\, H_2(z)\, H_3(z) = \frac{B_1(z)\, B_2(z)\, B_3(z)}{A_1(z)\, A_2(z)\, A_3(z)};$$

that is, the transfer function of the combined filter is the product of the transfer functions of the individual filters. Equivalently, if $h_1(n)$, $h_2(n)$, and $h_3(n)$ are the impulse responses of the three filters, we have the impulse response of the combined filter given by

$$h(n) = h_1(n) * h_2(n) * h_3(n);$$

that is, the impulse response of the combined filter is the convolution of the impulse responses of the individual filters.

The coefficients of the polynomial $B_1(z)\, B_2(z)$ are given by polynomial multiplication of $B_1(z)$ and $B_2(z)$. The same result is achieved by convolution of the corresponding b arrays, which can be done using the `conv` function in Matlab. You may use `conv` in series or in a nested call to obtain the coefficients of $B_1(z)\, B_2(z)\, B_3(z)$, which will give the b array of the combined filter.

7.5 Matlab commands to learn

Use the `help` command in Matlab and learn the following commands:
`filter, freqz, zplane, fft, conv`.

8 Laboratory Exercise: Detection of the QRS and Parameterization of the ECG

8.1 Objectives

- To detect QRS complexes in ECG signals using the Pan–Tompkins algorithm.
- To measure parameters of the ECG for analysis of the heart rate and cardiac rhythm.

8.2 Background

The algorithm developed by Pan and Tompkins identifies QRS complexes based on analysis of the slope, amplitude, and width of the QRS. The various stages of the algorithm are shown in Figure 2.

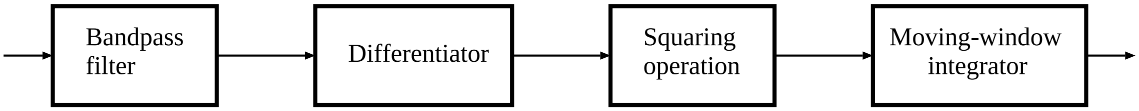

Figure 2: Block diagram of the Pan–Tompkins algorithm for the detection of QRS complexes in ECG signals.

The bandpass filter, formed using lowpass and highpass filters, reduces noise in the ECG signal. Noise such as muscle noise, 60 Hz interference, and baseline drift are removed by bandpass filtering. The signal is then passed through a differentiator to provide a large response at the high slopes that distinguish QRS complexes from low-frequency ECG components such as the P and T waves.

The next operation is the squaring operation, which emphasizes the higher values expected due to QRS complexes and suppresses smaller values related to the P and T waves, as well as noise in the output of the preceding stage. The squared signal is then passed through a moving-window integrator of window length $N = 30$ samples (for the sampling frequency of $f_s = 200$ Hz). The expected result is a single smooth peak related to the QRS complex for each ECG cycle. The output of the moving-window integrator may be used to detect QRS complexes, measure RR intervals, and determine the duration of the QRS complex (see Figure 3).

See Section 4.3.2 of the textbook for details.

8.3 Exercises to be worked by hand

1. A digital filter is specified in terms of its impulse response as

$$h_1(n) = \delta(n) + \delta(n-1) + \delta(n-2).$$

Derive its transfer function as well as the magnitude and phase parts of its frequency response. Explain the nature and effects of the filter.

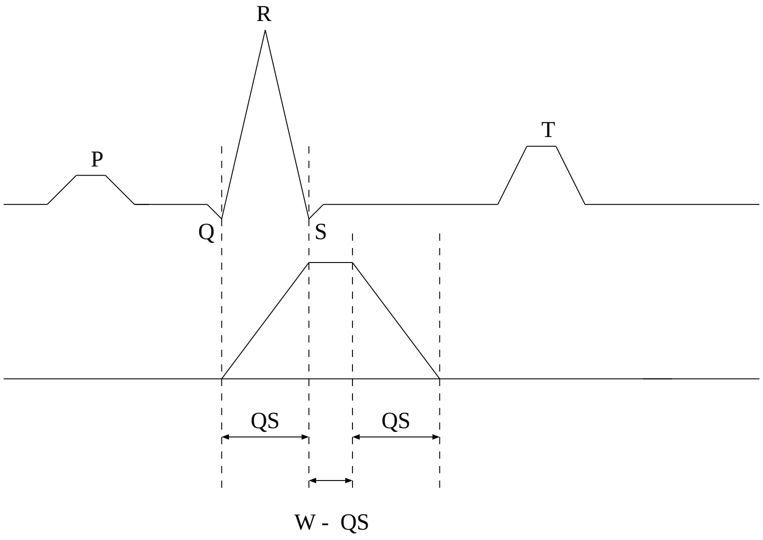

Figure 3: The relationship of a QRS complex to the moving-window integrator output. (a) Schematic ECG signal. (b) Output of the moving-window integrator. QS: QRS complex width. W: width of the integrator window, given as N/f_s s.

2. Another digital filter is specified in terms of its impulse response as

$$h_2(n) = \delta(n) - \delta(n-1).$$

   Derive its transfer function as well as the magnitude and phase parts of its frequency response. Explain the nature and effects of the filter.

3. The two filters given above are used in series to filter a signal. Derive and plot the impulse response of the combined filter.

   Derive the transfer function as well as the magnitude and phase parts of the frequency response of the combined filter. Explain the nature and effects of the combined filter.

8.4 Detection of the QRS wave

1. Develop a Matlab program to perform the various filtering procedures that compose the Pan–Tompkins algorithm. Use the `filter` command for each step; see Section 4.3.2 of the textbook. Process each of the three ECG signals that you acquired in Lab 1 using your program. Prepare plots of the signal being processed over the full duration (or a relevant portion of about $10-15$ seconds) and over two cardiac cycles, before and after each stage of the Pan–Tompkins method.

   You may also use `ECG3.dat`, `ECG4.dat`, `ECG5.dat`, `ECG6.dat`, and `ECGS.m`.

2. Plot the pole–zero diagram and the frequency response (magnitude and phase) of each of the initial lowpass, highpass, and the combined bandpass filters, and also the derivative-based and integrating filters. Study the plots and describe the characteristics as well as the effects of each stage of the QRS-detection algorithm.

3. Implement a thresholding procedure including a blanking interval for the detection of QRS complexes from the output of the Pan–Tompkins algorithm. Develop Matlab code to use the output of the Pan–Tompkins algorithm to detect QRS complexes. Mark the detected peak locations on the plots of the output of the integrator and also the corresponding original ECG signal, including a correction factor for the delays introduced by the filters.

4. Include steps in your program to compute automatically the following parameters for each ECG signal:

   (a) Total number of beats detected in each signal and the heart rate in beats per minute.

   (b) Average RR interval and standard deviation of RR intervals of each signal (in *ms*).

   (c) Average QRS width computed over all the beats in each signal (in *ms*).

   Verify your results by visual comparison of the outputs of your program with the original signals. Determine the number of missed beats or incorrectly detected beats, if any, in each signal. Tabulate and analyze the results for your ECG signals.

Note: The amplitude of an ECG signal may start with a value other than zero. As a consequence, the differentiator in the Pan–Tompkins algorithm will amplify the initial step, possibly resulting in an erroneous beat detection. In order to prevent this problem, subtract the value of the first sample of the ECG signal from the entire ECG prior to processing by the Pan–Tompkins algorithm.

8.5 Matlab commands to learn

Use the `help` command in Matlab and learn the following commands:
`filter, freqz, zplane`.

9 Laboratory Exercise: Detection of Spike-and-wave Complexes in EEG Signals

9.1 Objectives

- Detection of spike-and-wave complexes in EEG signals using template matching.
- Design and implementation of a matched filter to detect spike-and-wave complexes in EEG signals.

9.2 Background

A spike-and-wave complex (SWC) is a well-defined event in an EEG signal. The complex is composed of a sharp spike followed by a wave with a frequency of about 3 Hz; the wave may contain a half period or a full period of an almost-sinusoidal pattern. One may, therefore, extract a copy of an SWC from an EEG channel and use it for template matching. By computing the cross-correlation function (CCF), the template may be correlated with the same EEG signal from which it was extracted to detect similar events that appear at other instants of time, or with the EEG from another channel of the same subject or even a different subject to search for similar patterns or events.

Another approach to solve this problem is to design a matched filter to facilitate the detection of SWCs in EEG signals. When a sample observation of a typical version of a signal event is available (a template of an SWC in the present exercise), it becomes possible to design a filter that is matched to the characteristics of the event. If an observed signal is expected to contain repetitions of the event with almost the same characteristics, the signal may be passed through the matched filter. The output should provide peaks at the time instants of the occurrences of the event. Matched filters are designed to perform correlation between the input signal and the signal template. The output of the matched filter may be thresholded to detect peaks that correspond to SWCs in the given EEG record.

See Sections 4.4 and 4.6 of the textbook for details.

9.3 Template matching

Copy the files `eeg2*.dat` and `eeg2.m`. The signals are sampled at the rate of 100 Hz. Select an EEG signal and a template corresponding to an SWC. Write your own Matlab code to compute the CCF between the template and the signal. Apply a peak-detection algorithm to locate the peaks in the result. Mark the locations of the corresponding SWC complexes in the EEG signal. Apply time-delay corrections as required and explain.

Plot the EEG signal and the CCF, including marks to identify the peaks in the CCF and the SWC complexes detected in the EEG signal.

Apply the same template and procedures to two other EEG channels and analyze the results.

9.4 Matched filtering

Derive the impulse response of the matched filter from the template selected in the preceding experiment. Plot the template and the impulse response of the matched filter on the same figure and explain the relationships between the two.

Perform the matched filtering operation on the same EEG signals as in the preceding experiment by using the `filter` command in MATLAB. Process the output of the matched filter to detect peaks and mark SWCs in the original EEG signals corresponding to the peaks detected in the matched filter output. Take into account the delay in the output of the matched filter when identifying the onset of SWCs in the EEG signals.

Apply the same matched filter and procedures to two other EEG channels and analyze the results.

Discuss the similarities and differences between template matching and matched filtering and the results provided by the two methods.

10 Laboratory Exercise: Frequency-domain Analysis of Heart Sounds

10.1 Objectives

- Segmentation of phonocardiographic (PCG) signals.
- Computation of an averaged power spectral density (PSD) of a segment of a PCG signal.
- Parametric characterization of PSDs.
- Auditory analysis of heart sounds.

10.2 Background

The PCG is a recording of the heart sounds. Some of the important features of the PCG include: the frequency content of murmurs and heart sounds, the maximum intensities of the heart sounds, the intensity patterns of murmurs, and the timing sequence of the heart sounds and murmurs.

Heart sounds are caused by the acceleration or deceleration of blood in the heart's chambers. There are two major heart sounds that occur during a cardiac cycle. The first heart sound (S1) is due to ventricular contraction, and occurs at the same time as the QRS complex in the ECG signal. (See Section 1.2.9 of the textbook.) The frequency content of S1 is limited to a low-frequency band of about $10 - 120$ Hz.

The closure of the semilunar (pulmonary and aortic) valves gives rise to the second heart sound (S2). S2 occurs at about the end of the T wave in the ECG. The frequency content of S2 is usually higher than that of S1, in the range of about $10 - 200$ Hz.

The intervals between S1 and S2 of a cardiac cycle, and between S2 of a cycle and S1 of the next cycle (corresponding to ventricular systole and diastole, respectively) are normally silent. Murmurs, caused by certain cardiovascular defects and diseases, may occur in these intervals. Murmurs are high-frequency noise-like vibrations that arise when the velocity of blood becomes high due to an irregularity, orifice, or defect through which the blood flows.

The PSD of a signal is given by the squared magnitude of the Fourier transform (FT) of the signal. In the case of PCGs, the random nature of the vibrations and the associated signals causes variation in the frequency content from one heart beat (cardiac cycle) to another. Therefore, the PSD computed using one PCG cycle or segment will not provide an accurate representation of the PSD of the signal. In order to overcome this limitation, we could extract PCG segments from several cardiac cycles using the ECG as a reference (trigger) signal, compute the PSD of each segment, and average the PSDs over several cardiac cycles. In such a procedure, each segment should correspond to the same phase or period of the cardiac cycle. The procedure may be applied to derive separate PSDs for the different distinct parts of the PCG signal, such as S1, the systolic segment after S1, S2, and the diastolic segment after S2 (until the next S1). The averaged PSDs may then be characterized in terms of quantitative features, such as moments. See Sections 6.3.5 and 6.5 of the textbook.

10.3 Spectral analysis of heart sounds

1. Copy the files `pec1.dat, pec33.dat, pec41.dat, pec42.dat, pec52.dat,` and `pec_sound.m`.

   Each `.dat` file contains sampled values of the ECG, PCG, and carotid pulse signals of a subject. The sampling rate per channel is $f_s = 1,000$ Hz. Use the program `pec_sound.m` provided to read, separate, and plot the data, as well as to listen to the PCG signals. Some of the data files have artifacts at the beginning and/or ending of the recording session: delete such portions in your program.

   The files `pec1` and `pec52` contain signals from normal subjects. The files `pec33` and `pec42` contain signals from two subjects with ventricular septal defect (a hole between the two ventricles, causing blood to leak from the left ventricle to the right ventricle during systole), causing systolic murmur in the PCG. The file `pec41` contains signals from a subject with aortic stenosis (stiffened leaflets of the aortic valve causing incomplete opening of the valve and constrained ejection of blood into the aorta during ventricular systole), causing systolic murmur in the PCG.

2. For each case, apply the Pan–Tompkins algorithm (that you developed in an earlier lab exercise) to the ECG signal and detect the QRS complexes. Note that the Pan–Tompkins algorithm is designed for signals sampled at 200 Hz. In order to apply the algorithm to the ECG signal in the present exercise, prefilter only the ECG channel using a second-order Butterworth lowpass filter with a cutoff frequency of 60 Hz and downsample by a factor of five before applying the Pan–Tompkins algorithm. You may also have to modify your threshold parameters for the detection of the QRS complexes. Transfer the QRS point detected for each beat to the beginning of the corresponding S1 point in the PCG channel after incorporating the required delay and scale factors.

3. Segment the systolic portions of the PCG signal by selecting a window of duration $300 - 350$ ms starting from the beginning of each QRS complex. Ensure that the window that you define for each subject includes the beginning of S1 and systolic murmur (if present), but not S2. (You may determine this duration by visual inspection of each signal. See Sections 4.3.3 and 4.10 of the textbook for methods to detect the beginning of S2 using the carotid pulse.) Subtract the mean of each signal segment and compute the PSD of each segment, given as the square of the absolute value of the FT of the signal segment, divided by the number of samples in the FT array. Obtain the averaged PSD of the systolic portions of each PCG signal, using as many cardiac cycles as possible in a synchronized-averaging procedure (see Section 6.3.5 of the textbook).

4. For each PCG signal, prepare a plot of one sample systolic segment, a plot of the PSD of the single systolic segment, and a plot of the average PSD computed using as many cardiac cycles as available. Ensure that all of your plots have the time axis labeled in seconds, the frequency axis in Hertz, etc., as applicable. Provide an interpretation of your results.

5. Compute the mean frequency of the averaged PSD for each case (see Equation 6.30 of the textbook). Use only one half of the PSD (for positive frequency only) to compute the mean frequency. Estimate the approximate bandwidth of each PCG signal. Prepare a table of the mean frequency and the approximate bandwidth for each of the five signals provided. Give the appropriate units for the parameters that you compute from the PSDs. Analyze the parameters and discuss your findings.

6. Listen to each PCG signal (or part thereof) and describe your observations in your report. Relate your auditory analysis to the PSDs

   For playback and auditory analysis of heart sounds, use the `sound` command in Matlab. The sampling rate of the signals provided to you is 1 kHz. Time-scale (oversample or interpolate) the given PCG signals to 8 kHz by using the `interp` command. Some of the data files have artifacts at the beginning and/or ending of the recording session: delete such portions in your program. (A loud and strange sound could affect your perception of the immediately following sounds.)

   If there is a problem with the `sound` command in Matlab on the computers in the lab, refer to the commands provided in `pec_sound.m` to convert the PCG data array into a `.au` audio file. You may also convert the PCG signal into a `.wav` file using the command `wavwrite`. You may then listen to the `.au` or `.wav` files using audio tools and headphones.

11 Laboratory Exercise: Morphological Analysis and Pattern Classification of ECG Signals

11.1 Objectives

- To detect, segment, and extract discriminative features from QRS complexes in ECG signals for the characterization of normal beats and premature ventricular contractions (PVCs).
- To classify ECG waveforms into two groups: normal and PVC.

11.2 Background

Rhythm monitoring of cardiac patients is one of the vital functions of a coronary care unit. One particular type of cardiac abnormality to be monitored is a PVC. A PVC, which is an aberrantly conducted beat triggered by an ectopic impulse-generating focus, is characterized by a shorter preceding RR interval (hence called premature), and a bizarre QRS complex. The shape of the QRS complex of a PVC depends upon the location of the ectopic focus and the path of conduction taken by the impulse. This makes morphological analysis of the PVC a challenge, because a PVC does not possess any specific waveform. In addition, there could be several ectopic foci generating different types of PVCs in a given ECG signal.

The detection and identification of PVCs requires information regarding both the rhythm and the morphology of QRS complexes. Three features of interest for each beat are

- the preceding RR interval,
- the QRST area ($QRSTA$), and
- the correlation coefficient with the template of a normal beat (CC).

In order to measure the RR interval, the QRS complexes in the ECG signal have to be detected first. The Pan–Tompkins method that you implemented in an earlier laboratory exercise could be used for this purpose. The RR interval is given by the time interval between the detected marker point for the present QRS complex and the corresponding point for the preceding QRS complex. The RR interval is a useful measure to analyze the cardiac rhythm represented by the given ECG signal.

The QRST area may be derived by selecting a segment of the ECG including the QRS complex and the following T wave (about 300 ms in duration) using the peak detected in the output of the Pan–Tompkins method for reference, subtracting the mean, rectifying the result, and computing the area under the resulting curve. See Figure 5.3 in the textbook.

In order to compute CC, we need to select a normal beat to serve as the template. Then, the normalized dot product between the template and the corresponding portion of the current beat could be computed to obtain CC, as follows:

$$CC = \frac{\sum_{n=0}^{N-1} \left[x(n) - \min(x)\right] \left[y(n) - \min(y)\right]}{\sqrt{\sum_{n=0}^{N-1} [x(n) - \min(x)]^2 \sum_{n=0}^{N-1} [y(n) - \min(y)]^2}} \,. \tag{13}$$

Here, $x(n)$ is the template; $y(n)$ is the ECG beat being analyzed; $\min(x)$ and $\min(y)$ are the minimum values of $x(n)$ and $y(n)$, respectively; and N is the duration (number of samples) of the template and the beat being analyzed.

Ectopic beats (PVCs) are expected to be characterized by lower CC, higher $QRSTA$, and lower RR than normal beats.

11.3 Analysis and classification of ECG signals

Copy the data file `ecgpvcs.mat` and the program `pvcs.m`. Use only the first signal (`ecg1`) in the file `ecgpvcs.mat`.

Develop Matlab code to implement the following pattern classification procedure. See Chapter 9 of the textbook for related details and examples.

A. Training:

1. Use the first 40% of the given ECG signal for the training step. Apply the Pan–Tompkins method and detect the beats in the signal.
2. Segment each cardiac cycle (QRS-T complex) by taking a few samples before and a few samples after the corresponding marker point detected in the output of the Pan–Tompkins method. Use a total duration of about 300 ms for each beat. The P wave need not be considered in the present exercise. Prepare a plot of the entire ECG signal with the segmentation points marked for each beat in the signal. See Figure 5.11 of the textbook.
3. Prepare an array with one line per beat in the training signal, giving the ECG beat number; the type of the beat as normal (0) or PVC (1); $QRSTA$; and CC. In this step, you need to determine, by visual inspection, the type of each beat in the training part of the signal. Save this array for use in the testing step. Prepare a table of the mean and standard deviation of $QRSTA$ and CC computed separately for the normal beats and PVCs in the training part of the ECG signal.
4. Prepare a plot of the training portion of the given signal, with each beat labeled as 'o' or 'x', representing a normal beat or PVC, respectively.
5. Prepare a scatter plot of CC versus $QRSTA$ for all of the beats detected. Mark the plot with 'o' or 'x' for each normal beat or PVC, respectively. See Section 9.11 of the textbook.

B. Testing:

1. Apply the QRS detection and parameterization procedures as above to the remaining part of the ECG signal (testing part). For each beat detected, form a feature vector as $[QRSTA, CC]$. Prepare a table of the mean and standard deviation of $QRSTA$ and CC computed separately for the normal beats and PVCs in the testing part of the ECG signal. Prepare a scatter plot of CC versus $QRSTA$ for all of the beats detected in the testing part. Mark the plot with 'o' or 'x' for each normal beat or PVC, respectively.

2. For each beat, find the k nearest neighbors ($k-$NN) in the training set using the feature vector $[QRSTA, CC]$ and the Euclidean distance measure, and classify the beat as normal or PVC using the $k-$NN rule with $k = 1$. (See Section 9.4.4 of the textbook.)

   Repeat the procedure with $k = 3$.

3. Prepare a plot of the testing portion of the given signal, with each beat labeled as 'o' or 'x', representing a normal beat or PVC, respectively, as determined by the $k-$NN rule in your program.

4. Check the results of your classification procedure, and compute the accuracy of classification, including measures of true-positive, false-positive, true-negative, and false-negative fractions. Note also the number of beats not detected and false detections of beats (if any) by your program. Prepare a summary of your results in a table.

Note: The ranges of the values of CC and $QRSTA$ could be substantially different, which creates difficulties in $k-$NN analysis. To overcome this, normalize the $QRSTA$ values in the training set and also in the testing set by dividing by the maximum value of the feature in the training set.

See Sections 9.4.4 and 9.11 of the textbook for related material.

11.4 Matlab commands to learn

Use the `help` command in Matlab and learn the following commands:
`filter, freqz, zplane`.

12 Project: Wiener Filtering of ECG Signals

12.1 Objectives

- Filter and remove artifacts in ECG signals.
- Design a Wiener filter using the autocorrelation function (ACF) and power spectral density (PSD) functions of ECG signals and noise.
- Compare the performance of the Wiener filter with other time-domain and frequency-domain filters.

12.2 Specific tasks

1. Get at least three ECG signals: one with a substantial amount of high-frequency noise, one with low-frequency noise, and one that is relatively free of noise. You may use your own ECG signal files from Lab 1 or the data files

   `ecg_hfn.dat, ecg_hfn.m, ecg_lfn.dat, ecg_lfn.m, pec*.dat,` and `plotpec.m.`

   The sampling rate is 1000 Hz or 200 Hz: please check each case carefully. Resample the signals to the sampling rate desired for your project if required.

2. Implement one time-domain and one frequency-domain filtering method (see Chapter 3 of the textbook) to remove high-frequency and low-frequency artifacts in the signals.

3. Implement the Wiener filter (see Section 3.8 of the textbook). For this purpose, estimate the ACF and PSD of noise-free ECG signals and the noise found in the examples available by averaging over multiple observations of the processes and fitting appropriate functions. Implement the Wiener filter in the frequency domain as in Equation 3.185 in the textbook.

4. Implement the Wiener filter in the time domain by obtaining the inverse Fourier transform of the transfer function from the preceding step, truncation as needed, and using the Matlab `filter` command.

5. Compare the results obtained using the various filters in your study.

13 Project ECG-A: Morphological Analysis and Pattern Classification of ECG Signals

13.1 Objectives

- Filter and remove artifacts in ECG signals.
- Detect, segment, and extract discriminative features from QRS complexes in ECG signals for the characterization of normal beats and premature ventricular contractions (PVCs).
- Classify ECG beats (waveforms) into two groups: normal and PVC.

13.2 Specific tasks

1. Get an ECG signal with PVCs. You may use the data file `ecgpvcs.mat` and the program `pvcs.m`. Use only the first signal (`ecg1`) in the file `ecgpvcs.mat`. The sampling rate is 200 Hz.
2. Implement two filtering methods (see Chapter 3 of the textbook) — one in the time domain and another in the frequency domain — to remove high-frequency and low-frequency artifacts in the signal. You may use the signals `ecg_hfn.dat` and `ecg_lfn.dat` to test your programs for noise removal; see the corresponding `.m` files.
3. Implement the Pan–Tompkins method to detect QRS complexes in the ECG signal; see Section 4.3.2 of the textbook.
4. Implement methods to derive at least two features to characterize each segmented ECG beat, including the QRST area and the correlation coefficient with the template of a normal beat selected from the same ECG signal; see Chapter 5 of the textbook.
5. Implement the k-nearest neighbor method to classify each beat as a normal or a PVC (see Chapter 9 of the textbook). Use features from the first $10-20\%$ of the signal as the training or reference set for the classifier and the remaining signal to test the classifier. Check the results of your classification procedure and compute the accuracy of classification, including measures of true-positive, false-positive, true-negative, and false-negative fractions. Note also the number of beats not detected and false detections of beats (if any) by your program.
6. Prepare a summary of your results in a table.

14 Project ECG-B: Morphological Analysis and Pattern Classification of ECG Signals

14.1 Objectives

- Filter and remove artifacts in ECG signals.
- Detect, segment, and extract discriminative features from QRS complexes in ECG signals for the characterization of normal beats and premature ventricular contractions (PVCs).
- Classify ECG beats (waveforms) into two groups: normal and PVC.

14.2 Specific tasks

1. Get an ECG signal with PVCs. You may use the data file `ecgpvcs.mat` and the program `pvcs.m`. Use only the first signal (`ecg1`) in the file `ecgpvcs.mat`. The sampling rate is 200 Hz.
2. Implement two filtering methods (see Chapter 3 of the textbook) — one in the time domain and another in the frequency domain — to remove high-frequency and low-frequency artifacts in the signal. You may use the signals `ecg_hfn.dat` and `ecg_lfn.dat` to test your programs for noise removal; see the corresponding `.m` files.
3. Implement the method of Balda et al. to detect QRS complexes in the ECG signal; see Section 4.3.1 of the textbook.
4. Implement methods to derive at least two features to characterize each segmented ECG beat, including the form factor and the QRS width; see Chapter 5 of the textbook.
5. Implement a linear discriminant function to classify each beat as a normal or a PVC (see Chapter 9). Use features from the first $25 - 30\%$ of the signal as the training or reference set for the classifier and the remaining signal to test the classifier. Check the results of your classification procedure and compute the accuracy of classification, including measures of true-positive, false-positive, true-negative, and false-negative fractions. Note also the number of beats not detected and false detections of beats (if any) by your program.
6. Prepare a summary of your results in a table.

15 Project EMG-A: Analysis of the Relationship Between Parameters of the EMG Signal and Muscular Force

15.1 Objectives

- Characterize the level of activity in EMG signals.
- Analyze the relationship between parameters of the EMG signal and muscular force.

15.2 Specific tasks

1. Get EMG signals related to various levels of muscular activity or force. You may use the data files

   `EMGforce.txt, EMGforce2.txt, EM_EMG_SQUEEZE1.txt,` and `EM_EMG_SQUEEZE2.txt`
   as well as the program `EMGforce.m`. The sampling rate is 2000 Hz per channel and the EMG sample values are in mV.

2. Normalize the force signal such that the minimum value is zero and the maximum value (corresponding to the maximum voluntary contraction or MVC) is 100. Filter the EMG signal to remove noise and artifacts (experiment with Butterworth lowpass and highpass filters with various bandwidths). Plot the EMG signal (in mV) and normalized force (in %MVC) against the time axis.

3. Develop methods for automatic identification of portions (segments) corresponding to each level of contraction within which the force remains close to the corresponding peak values. Explain your algorithm in your report.

4. For each segment of the EMG signal identified as above, compute at least four suitable parameters (see Chapters 3, 5, and 6 of the textbook). Ensure that your list of EMG features includes the following: RMS value, fractal dimension using Higuchi's method, and mean and median frequencies via spectral analysis. Compute also the average force (in %MVC) for each segment.

5. Plot the values of the various parameters versus force in %MVC. Label the axes with the appropriate units. Analyze the results in terms of statistical variation of the parameters in relation to force.

6. Using the `polyfit` function in Matlab, obtain a straight-line (linear) fit to represent the variation of each EMG parameter versus force. Use `polyval` to evaluate the values of the dependent variable given by the model for the available values of the independent variable. Superimpose the linear models (straight-line fits) obtained on the plots of the parameters in the preceding step. Analyze the results.

7. Compute the correlation coefficient, r, with r^2 given by the formula in Equation 5.28 in the textbook. Using r, analyze the goodness of fit for each parameter and discuss the appropriateness of the linear model. See

   `http://mathworld.wolfram.com/CorrelationCoefficient.html`

   for details on linear least-squares fitting and the correlation coefficient. See Chapter 5 of the textbook for related examples.

8. Tabulate the parameters of the linear model and r for each of the EMG parameters. Analyze the results and describe your findings.

9. Perform all of the above steps with each of the four EMG signals provided as above. Study the effects of filtering the EMG signals on the features extracted and their relationships with force. Analyze and compare the results.

16 Project EMG-B: Analysis of the Relationship Between Parameters of the EMG Signal and Muscular Force

16.1 Objectives

- Characterize the level of activity in EMG signals.
- Analyze the relationship between parameters of the EMG signal and muscular force.

16.2 Specific tasks

1. Get EMG signals related to various levels of muscular activity or force. You may use the data files

   `EMGforce.txt`, `EMGforce2.txt`, `EM_EMG_SQUEEZE1.txt`, and `EM_EMG_SQUEEZE2.txt`
   as well as the program `EMGforce.m`. The sampling rate is 2000 Hz per channel and the EMG sample values are in mV.

2. Normalize the force signal such that the minimum value is zero and the maximum value (corresponding to the maximum voluntary contraction or MVC) is 100. Filter the EMG signal to remove noise and artifacts (experiment with Butterworth lowpass and highpass filters with various bandwidths). Plot the EMG signal (in mV) and normalized force (in %MVC) against the time axis.

3. Develop methods for automatic identification of portions (segments) corresponding to each level of contraction within which the force remains close to the corresponding peak values. Explain your algorithm in your report.

4. For each segment of the EMG signal identified as above, compute at least four suitable parameters (see Chapters 3, 5, and 6 of the textbook). Ensure that your list of EMG features includes the following: mean-squared value, fractal dimension using power spectral analysis (see Section 6.6 of the textbook), zero-crossing rate, and turns count. Compute also the average force (in %MVC) for each segment.

5. Plot the values of the various parameters versus force in %MVC. Label the axes with the appropriate units. Analyze the results in terms of statistical variation of the parameters in relation to force.

6. Using the `polyfit` function in Matlab, obtain a straight-line (linear) fit to represent the variation of each EMG parameter versus force. Use `polyval` to evaluate the values of the dependent variable given by the model for the available values of the independent variable. Superimpose the linear models (straight-line fits) obtained on the plots of the parameters in the preceding step. Analyze the results.

7. Compute the correlation coefficient, r, with r^2 given by the formula in Equation 5.28 in the textbook. Using r, analyze the goodness of fit for each parameter and discuss the appropriateness of the linear model. See

   `http://mathworld.wolfram.com/CorrelationCoefficient.html`

   for details on linear least-squares fitting and the correlation coefficient. See Chapter 5 of the textbook for related examples.

8. Tabulate the parameters of the linear model and r for each of the EMG parameters. Analyze the results and describe your findings.

9. Perform all of the above steps with each of the four EMG signals provided as above. Study the effects of filtering the EMG signals on the features extracted and their relationships with force. Analyze and compare the results.

17 Project: Frequency-domain Analysis of Heart Sounds

17.1 Objectives

- Segmentation of phonocardiographic (PCG) signals using the ECG and carotid pulse as references.
- Computation of averaged power spectral densities (PSDs) of the systolic and diastolic segments of a PCG signal.
- Parametric representation of PCG signals.
- Auditory analysis and classification of heart sounds.

17.2 Specific tasks

1. Get a few PCG signals of normal subjects and patients with murmurs. You may use the files `pec1.dat, pec33.dat, pec41.dat, pec42.dat, pec52.dat,` and `pec_sound.m`.

   Each `.dat` file contains sampled values of the ECG, PCG, and carotid pulse signals of a subject. The sampling rate per channel is $f_s = 1,000$ Hz. Use the program `pec_sound.m` provided to read, separate, and plot the data, as well as to listen to the PCG signals. Some of the data files have artifacts at the beginning and/or ending of the recording session: delete such portions through your program.

   The files `pec1` and `pec52` contain signals from normal subjects. The files `pec33` and `pec42` contain signals from two subjects with ventricular septal defect (a hole between the two ventricles, causing blood to leak from the left ventricle to the right ventricle during systole), causing systolic murmur in the PCG. The file `pec41` contains signals from a subject with aortic stenosis (stiffened leaflets of the aortic valve causing incomplete opening of the valve and constrained ejection of blood into the aorta during ventricular systole), causing systolic murmur in the PCG.

2. Detect the QRS complexes in the ECG signals by applying the method of Balda et al. (see Section 4.3.1 of the textbook). Note that the algorithm is designed for signals sampled at 200 Hz. In order to apply the algorithm to the ECG signal in the `pec` signals, prefilter only the ECG channel using a suitable Butterworth lowpass filter and downsample by a factor of five before applying the algorithm. Transfer the QRS point detected for each beat to the beginning of the corresponding S1 point in the PCG channel after incorporating the required correction or scaling factors.

3. Detect the dicrotic notch in the carotid pulse signal by implementing the method described in Section 4.3.3 of the textbook.

4. Segment the systolic and diastolic portions of the PCG signal by using the related QRS complex and dicrotic notch positions detected in the preceding steps; see Section 4.10 of the textbook.

5. Obtain the averaged PSDs of the systolic and diastolic portions of each PCG signal, using as many cardiac cycles as possible in a synchronized-averaging procedure (see Chapter 6 of the textbook). Plot and compare the PSD of an individual systolic or diastolic segment with the average PSD of several systolic or diastolic segments.

6. Compute the mean frequencies of the averaged systolic and diastolic PSDs for each case (see Chapter 6 of the textbook). Estimate the approximate bandwidth of each PCG signal as in the corresponding averaged systolic and diastolic PSDs. Explore the design of a spectral ratio as in Equation 6.41 in the textbook to distinguish between normal PCG signals and those with murmurs (define suitable values for the frequency limits f_1, f_2, and f_3). Prepare a table of the frequency-domain parameters as above for each of the five signals provided. Give the appropriate units for the parameters that you compute from the PSDs. Analyze the parameters and discuss your findings.

7. Listen to each PCG signal and describe your observations in your report. Relate your auditory analysis to the PSDs.

   The sampling rate of the signals provided to you is 1 kHz. Time-scale (oversample or interpolate) the given PCG signals to 8 kHz by using the `interp` command. Some of the data files have artifacts at the beginning and/or ending of the recording session: delete such portions in your program. (A loud and strange sound could affect your perception of the immediately following sounds.) Refer to the commands provided in `pec_sound.m` to convert the PCG data array into a `.au` audio file. You may also convert the PCG signal into a `.wav` file using the command `wavwrite`. You may then listen to the `.au` or `.wav` files using suitable audio tools.

18 Project VAG-A: Analysis and Classification of VAG Signals

18.1 Objectives

- Derive parameters to characterize the level of activity and complexity in VAG signals.
- Use the derived parameters for screening of VAG signals (classification as normal or abnormal).
- Study the effects of prefiltering or bandwidth selection on the features and classification accuracy.

18.2 Specific tasks

1. Get VAG signals of a number of normal subjects and patients with knee-joint pathology from the file `VAG_Signals89Share.zip`. The sampling rate is $2,000$ Hz.
2. Filter the VAG signals to remove noise and artifacts (experiment with Butterworth lowpass and highpass filters with various bandwidths).
3. Develop methods for automatic identification of portions (segments) of each signal that are quasistationary. You may use statistical or spectral measures. Explain your algorithm in your report.
4. For each segment of the VAG signal identified as above, compute at least four suitable parameters (see Chapters 5 and 6 of the textbook). Ensure that your list of features includes the following: mean-squared value, fractal dimension using Higuchi's method, and mean and median frequencies via spectral analysis.
5. Tabulate the parameters and their statistics for the groups of normal and abnormal VAG signals. Analyze the results and describe your findings.
6. Split the available data into two parts: one for training and another for testing a classifier. Develop a linear discriminant classifier using the parameters of the training set to classify the signals as normal or abnormal. Evaluate the classifier in terms of the true-positive, true-negative, false-positive, and false-negative rates as well as the overall classification accuracy.
7. Test the classifier with the test part of the dataset and evaluate the results as above.

19 Project VAG-B: Analysis and Classification of VAG Signals

19.1 Objectives

- Derive parameters to characterize the level of activity and complexity in VAG signals.
- Use the derived parameters for screening of VAG signals (classification as normal or abnormal).
- Study the effects of prefiltering or bandwidth selection on the features and classification accuracy.

19.2 Specific tasks

1. Get VAG signals of a number of normal subjects and patients with knee-joint pathology from the file `VAG_Signals89Share.zip.` The sampling rate is $2,000$ Hz.
2. Filter the VAG signals to remove noise and artifacts (experiment with Butterworth lowpass and highpass filters with various bandwidths).
3. Develop methods for automatic identification of portions (segments) of each signal that are quasistationary. You may use statistical or spectral measures. Explain your algorithm in your report.
4. For each segment of the VAG signal identified as above, compute at least four suitable parameters (see Chapters 5 and 6 of the textbook). Ensure that your list of features includes the following: form factor, fractal dimension via power spectral analysis, and turns count.
5. Tabulate the parameters and their statistics for the groups of normal and abnormal VAG signals. Analyze the results and describe your findings.
6. Split the available data into two parts: one for training and another for testing a classifier. Develop a k-nearest neighbor classifier (see Chapter 9 of the textbook) using the parameters of the training set to classify the signals as normal or abnormal. Evaluate the classifier in terms of the true-positive, true-negative, false-positive, and false-negative rates as well as the overall classification accuracy.
7. Test the classifier with the test part of the dataset and evaluate the results as above.

20 Project: Analysis of the Relationship Between Parameters of the EEG Signal and Sleep Stage

20.1 Objectives

- Characterize the nature of activity in EEG signals.
- Analyze the relationship between parameters of the EEG signal and sleep stage.

20.2 Specific tasks

1. Get EEG signals related to various stages of sleep. You may use the data in `SleepEEGdata.mat` and the program `SleepEEGread.m`. (Data courtesy of Dr. R. Agarwal [1].) The sampling rate is 200 Hz.
2. Filter the EEG signal to remove noise and artifacts. Experiment with Butterworth lowpass and highpass filters with various bandwidths. Evaluate the results in terms of the visual appearance and Fourier power spectra of the signals before and after filtering as well as the effects of filtering on sleep stage classification accuracy.
3. For each segment of the EEG signal provided, compute at least three suitable parameters (see Chapters 5 and 6 of the textbook). Ensure that your list of features includes the following: form factor and two spectral ratios as in Equation 6.41 in the textbook (define suitable values for the frequency limits f_1, f_2, and f_3).
4. Plot the values of the various parameters versus sleep stage. Analyze the results in terms of the variation of the parameters in relation to sleep stage. See Section 6.7 of the textbook.
5. Develop a pattern recognition or classification scheme (see Chapter 9 of the textbook) to label each EEG segment automatically with a sleep stage. Use the first 200 epochs from any one EEG channel to develop or train the scheme and the remaining epochs to test the scheme. Use the sleep stage data provided to verify your results.
6. Perform all of the steps given above with any two EEG channels. Analyze and compare the results.

Reference [1] R. Agarwal and J. Gotman. Computer-assisted sleep staging. IEEE Transactions on Biomedical Engineering, 48(12):1412-1423, 2001.

21 Preparation of Reports on Laboratory Exercises and Projects

21.1 Reports on laboratory exercises

Complete all exercises and report your findings appropriately, with the following observations:

- Plots, graphs: Use graphs to explain your results as necessary. You may plot more graphs than indicated in the exercises in order to provide a good explanation of your results. Always label the axes of your graphs and use appropriate axis-scale factors to facilitate visualization of details of interest. If the units of a variable are not calibrated or unknown, label the corresponding axis as "arbitrary units" or "AU."
- Matlab code: Your code must be clear and properly documented, with names of variables and functions that are appropriate for the specific exercise.
- Report: Your report must be a well-structured document in which the results obtained in each experiment are introduced in the text (e.g., "Figure 4 shows the sampled signal $x(n)$...") and discussed as necessary.

21.2 Project reports

Your project report must include an introductory review of your chosen subject area and problem, complete technical details of the methods studied and implemented (equations, procedures, and algorithms), illustrations of results, critical analysis and discussion of the results obtained, and references. More attention should be paid to the signal processing and analysis techniques studied than to the specific type of signal used or the application area of the project. Do not include your computer code. See papers in the IEEE Transactions on Biomedical Engineering `http://ieeexplore.ieee.org/xpl/tocresult.jsp?isnumber=5672100` for examples of research reports.

In a statement signed by each team member submitted with the report, you should specify how the project workload was shared, in percentage, between the members of your group; for equal assignment of marks, it is expected that each member has contributed equally to the project.

The recommended length of the report is eight pages in the IEEE Transactions on Biomedical Engineering format with two columns and single-spaced printing, including illustrations and references. For assistance on preparing and formatting your report, see

`http://www.ieee.org/publications_standards/publications/authors/authors_journals.html`.

Meet regularly to discuss your project, present and discuss ongoing work and results, and maintain progress in the project. Do not leave large portions of the project work for later parts of the term!

21.3 Design your own project!

Projects must involve computer implementation (in Matlab or any other language of your choice) of algorithms for digital signal processing and analysis, testing of the methods with real-life signals from any biomedical application area of your choice, and analysis of the results. The research, programming, data analysis, and report for the project must all be conducted, performed, and written by yourself (or in teams of two or three students each).

The algorithms you study and implement need not be original, but must be technically more advanced and sophisticated than the laboratory exercises. You may select a few previously published methods for filtering for the removal of artifacts, recognition or detection of events, and time-domain and/or frequency-domain analysis of biomedical signals, study them, implement them, test them with biomedical signals, and report on your own findings. The studies related to the project must be your own.

If you wish to develop your own project, ensure that the following requirements are met:

- Your project involves the processing of real-life biomedical signals. Several signals are available on the website `http://people.ucalgary.ca/~ranga/enel563/SIGNAL_DATA_FILES/`

  Refer to lab exercises in the textbook and the preceding sections of this document for details regarding the signals and related lab exercises.

  Signals of various types are available on the website `http://www.physionet.org/physiobank/` made available by the National Institute of Biomedical Imaging and Bioengineering (NIBIB), an institute within the National Institutes of Health (NIH), USA.

- Your project includes at least two methods to filter and remove noise or artifacts (see Chapter 3 of the textbook).

- Your project includes at least one method to detect and segment events, episodes, components, or parts of the input signal(s) for further analysis (see Chapter 4 of the textbook).

- Your project includes at least three methods for parametric representation and quantitative analysis of signals or their components (see Chapters 5 and 6 of the textbook).

- Your project includes at least one method for pattern classification and diagnostic interpretation of signals or their components (see Chapter 9 of the textbook).

Wish you success with your project and studies!