Electrocardiogram Compression and Optimal Filtering Algorithm

MIHAELA LASCU, DAN LASCU
Department of Measurements and Optical Electronics
Faculty of Electronics and Telecommunications
Bd. Vasile Pârvan no.2
ROMANIA

http://www.etc.upt.ro

Abstract: - In this paper novel compression techniques are developed for portable heart-monitoring equipment that could also form the basis for more intelligent diagnostic systems thanks to the way the compression algorithms depend on signal classification. There are two main categories of compression which are employed for electrocardiogram signals: lossless and lossy. Design of an optimal Wiener filter is implemented to remove noise from a signal, considering that the signal is statistically stationary and the noise is a stationary random process that is statistically independent of the signal. Two programs for compression and Wiener optimal filtering are realised in MATLAB.

Key-Words: - Electrocardiogram, Compression, Filtering, Matlab, Noise, Diagnostic.

1 Introduction

The electrocardiogram (ECG) is one of the most important and widely used quantitative diagnostic tools in medicine. It is extremely useful for the diagnosis and management of heart abnormalities such as heart attacks and offers helpful clues to the presence of generalized disorders that affect the rest of the body, such as electrolyte disturbances and drug intoxication. ECGs can show long-term effects: previous cardiac events such as heart attacks that can result in permanent modification to the morphology of the ECG. Commercial ambulatory recorders typically have sample rates up to 360 samples per second with a resolution of 10 or 12 bit giving a bit rate of around 4000bit/s. A typical commercial sample rate of 256 samples per second with 10bit resolution on two channels over seven days implies a memory requirement of close to 400MB of data [3].

On top of the storage issue, there is increasing interest in remote monitoring, using real-time or off-line transmission of complete records. As a result, compression is a key concern for makers of ECG equipment.

There are two main categories of compression which are employed for ECG signals: lossless and lossy. Lossless compression refers to any scheme whereby the signal reconstructed after compression is identical in every respect to the original signal. By contrast, lossy schemes allow differences between the original and the reconstructed signal.

The ECG is a real-world signal and is generally acquired from a relatively noisy electrical environment. Any lossless compression scheme has to reconstruct this random signal perfectly. This severely limits the effective compression ratio of lossless schemes when applied to ECG data. Lossless compression schemes may offer compression ratios of two or less. However, if restrictions on perfect reconstruction of the noise are relaxed, there is considerable scope for enhancing performance by utilising knowledge concerning the morphology of the ECG and its cyclo-stationary characteristics.

Having established that lossy compression schemes offer the greatest scope for achieving useful compression ratios, two further categories may be identified within that class: direct and indirect transformation processes. Direct compression schemes are generally less computationally intensive and operate on the time-domain ECG signal, using relatively simple approaches such as piece-wise linear approximation. The highest compression ratio with the best reconstruction quality can only be achieved using indirect compression methods, also called transform methods.

The recognising beats techniques generally exploit the cyclo-stationary nature of the ECG record. The nature of the beats within the ECG must be understood. More specifically deviations from the typical beat must be explicitly or implicitly recognised in order to represent them efficiently. A typical recording consists of a series of ECG beats separated by periods of inactivity as illustrated in Fig.1.

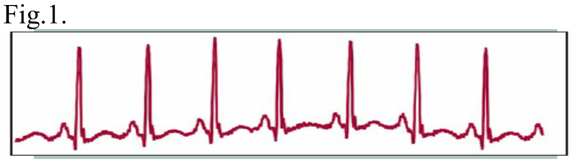


Fig.1. A typical ECG recording of a normal subject, clearly showing the cyclo-stationary nature of the ECG beat.

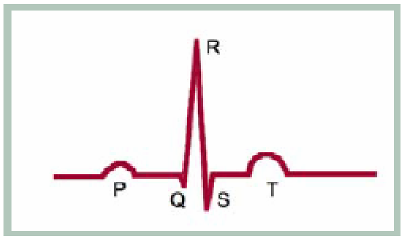


Fig.2. Idealised ECG beat, showing the P-wave, QRS-complex and T-wave.

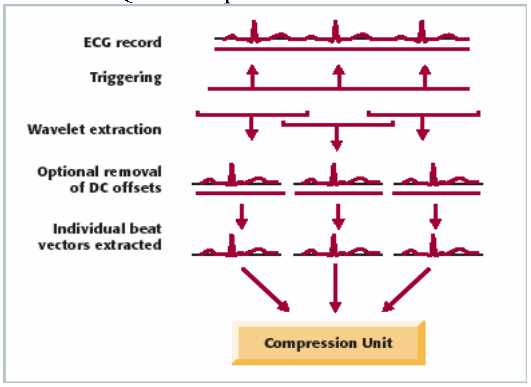

ECG record
Triggering
Wavelet extraction
Optional removal of DC offsets
Individual beat vectors extracted
Compression Unit

Fig.3. Generic ECG indirect compression scheme.

Note that even when the heart rate varies, the basic morphology and temporal extent of the beat are relatively unchanged; the main difference appears with the gap between beats. Important sections of the beat are labeled as the P-wave, QRS complex, and the T-wave, as shown in the idealized waveform of Fig.2. In practice there will often be two or more groups of beats, with each group having its own distinct morphology. These differences may be clinically diagnostic. Similarities within a group are exploited using indirect compression schemes.

Fig.3 shows the generic strategy used in many indirect compression methods, though a variant exists where the local DC removal step is omitted or carried out before triggering [4]. First, the ECG record is processed to locate the centre of each beat, thus allowing individual beat vectors to be extracted. These are then passed to the compression unit itself, which may be based on wavelet transforms and artificial neural networks, principal component analysis - PCA or non-linear principal component analysis- NLPCA.

ECG data compression algorithms are important for storage, transmission and analysis. An essential requirement of the compression algorithms is that the significant morphological features of the signal should not be lost upon reconstruction.

PCA is one of the most established techniques in multivariate statistical analysis and has been applied to ECG compression. If each beat consisted of N samples and each time-sample were allocated an axis in N-dimensional space, each beat could be plotted as a single point in N-dimensional space, with each sample voltage amplitude simply indicating the distance to plot along the corresponding axis. A collection of M beats may thus be plotted as a set of M points in this multidimensional space. It should be noted that all axes are equally important in this representation, as they are all involved in reconstruction of a beat. Also, the variables are not independent since there will be some correlation between adjacent samples and also with corresponding samples in other beats. This is a key feature that is exploited in PCA compression.

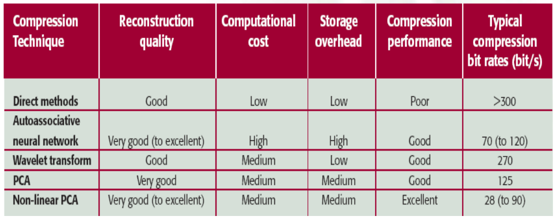
2 Neural Networks for Data Compression

Compression is achieved by restricting the number of hidden-layer neurons in the neural network compared with the number of input nodes and output neurons. This effectively forces the neural network during training to learn how to represent each ECG difference waveform [2] with fewer coefficients than the number of raw samples in the difference ECG. As autoassociative neural networks are selflearning, we do not specify how they represent the compressed data, although detailed examination of the weights indicates they learn by extracting something akin to the eigenvectors of a principal components decomposition, another key technique for ECG compression. An alternative to neural-network compression is through the use of the wavelet transform and its derivatives [4]. In contrast to the infinite-duration sinusoids encountered in Fourier analysis, a wavelet's oscillations dampen down to zero after a few cycles, and the function is localised in time, lasting only for a few cycles.

Using PCA compression, recognisable reconstruction of a given beat may be achieved by summing the contributions of just the first few basis vectors as these contain most of the energy. The eigenvectors themselves form part of the overhead but need to be stored only once for the whole set, which may have thousands of beats. The quality of the compression and reconstruction depends on how many of the PCA coefficients are used. Good reconstruction may be achieved using 10 or fewer coefficients [3].

As we can see in [3], Table 1 compares the performances of various compression techniques. PCA gives optimal compression performance and exceeds wavelet transform performance, though it requires marginally more processing overheads. The performance is slightly poorer than neural-network compression but the processing overhead is significantly lower. Non-linear PCA has significantly lower processing overheads than neural networks but provides comparable compression performance and fidelity. The fidelity is also selectable through the number of stored coefficients. Additional benefits indicate this approach to be suitable as the basis for a complete ECG analysis and classification system.

Table 1 Comparison of different ECG compression techniques.



Compression Technique	Reconstruction quality	Computational cost	Storage overhead	Compression performance	Typical compression bit rates (bit/s)
Direct methods	Good	Low	Low	Poor	>300
Autoassociative neural network	Very good (to excellent)	High	High	Good	70 (to 120)
Wavelet transform	Good	Medium	Low	Good	270
PCA	Very good	Medium	Medium	Good	125
Non-linear PCA	Very good (to excellent)	Medium	Medium	Excellent	28 (to 90)

After a study concerning Table 1 it can be seen that the autoassociative neural network compression technique has a very good to excellent reconstruction quality.

We make a short presentation concerning the auto-associative neural networks because this compression technique is also implemented in MATLAB.

A network compression ratio τ on an originally D-dimensional vector means that the middle hidden layer must have D/τ neurons [2], [5], [6]. For a linear network it represents D-dimensional inputs with a D/τ dimensional hidden layer. For a non-linear network, there is the added freedom to choose the dimensionality of the second and the fourth layers. We have choosen to keep the compression ratio between two layers constant. Therefore, second layers will have a dimensionality of $D/\tau^{0,5}$, representing a $\tau^{0,5}$ times compression from the input layer. The same compression ratio is also applied from the second layer to the third layer (the bottleneck). Therefore, once again, the bottleneck layers take a dimensionality of D/τ. This architecture for the non-linear networks is illustrated in Fig.4.

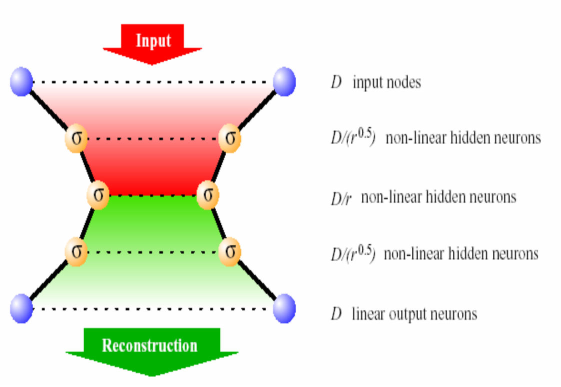

Input
D input nodes
$D/(r^{0.5})$ non-linear hidden neurons
D/r non-linear hidden neurons
$D/(r^{0.5})$ non-linear hidden neurons
D linear output neurons
σ
Reconstruction

Fig.4. A five-layer non-linear autoassociative network with bottleneck layer. The areas in red and green indicate respectively the compressing layers and the decompressing layers. The activation functions for the blue neurons are linear and those for the neurons in orange are sigmoidal.

The neural-network scheme to be used in this paper is the multilayer perceptron (MLP) model as in Fig.5.

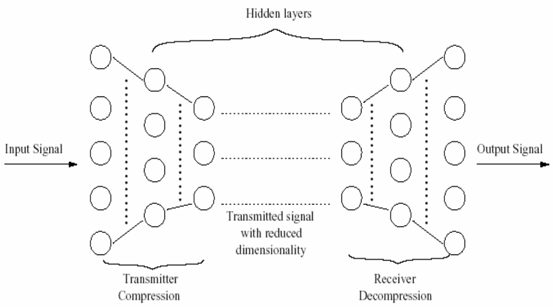


Fig.5. A five-layer architecture with reduced dimensionality at the hidden layers.

Multilayer neural networks have the ability to map inputs in a non-linear manner. There we use an MLP neural network for finding the non-linear relations between inputs. To achieve data compression, the hidden layers must have a lower dimensionality than the input and output layers. With a bottleneck at the hidden layers, the MLP is forced to find suitable relations between each input with a lower dimensionality. The data appears at the hidden layer with the lowest dimensionality, therefore, is a compact representation of the input data. The remaining parts of the MLP would reconstruct and expand non-linearly the compressed signal to the original dimensionality.

The entire compression process is in Fig.6 described.

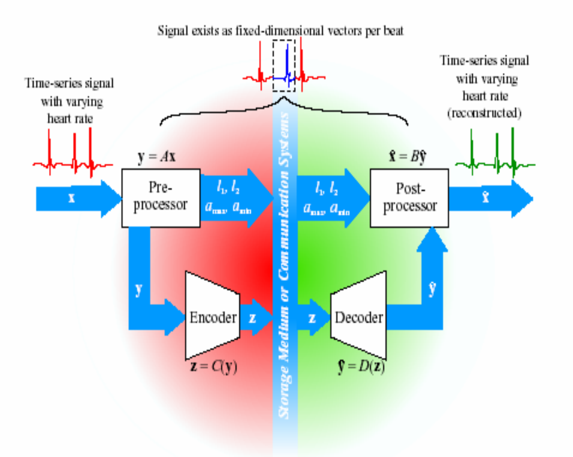

Signal exists as fixed-dimensional vectors per beat
Time-series signal with varying heart rate
Time-series signal with varying heart rate (reconstructed)
y = Ax
x̂ = Bŷ
x
Pre-processor
l_1, l_2
a_{max}, a_{min}
l_1, l_2
a_{max}, a_{min}
Post-processor
x̂
y
Encoder
z
Storage Medium or Communication Systems
z
Decoder
ŷ
z = C(y)
ŷ = D(z)

Fig.6. The complete compression system.

The locations of the R peaks are first determined and boundaries between beats are determined. The time-varying beats are then pre-processed to become fixed-dimensional input vectors. The input vectors are applied to a neural network with a bottleneck middle layer. Finally, the output vectors are post-processed to produce a reconstruction of the original time-series format.

Mathematically, we can describe the original ECG heartbeat as x^n, for the n-th beat. The length of the vector is equal to the number of data points the heartbeat occupies in the original time-series format and the elements take the values of the normalised amplitude. Throughout the complete compression and reconstruction process, information is lost in two major ways: the linear interpolation of the pre-processing and post-processing stages and the reduction of dimensionality at the neural network bottleneck. It is useful to have two definitions of error so that we can quantify the different contributions from the above two sources. We therefore propose to evaluate both the overall error, which is a measure including both error sources and the network error, which provides information on the second error source only. The network error is also used to evaluate the progress of learning when the network is being trained. The network error and the variance ratio reach minimum at roughly the same time. It can be shown that the global minimum of the sum-of-square error occurs at the point when the network regresses to the average vector of the training set, hence reconstructing the same average vector for all input patterns [2]. Therefore, the network has learnt the average of the training set for the global minimum of the sum-of-square error; this effect can be shown visually by snapshooting the reconstruction during the training process in Fig.7.

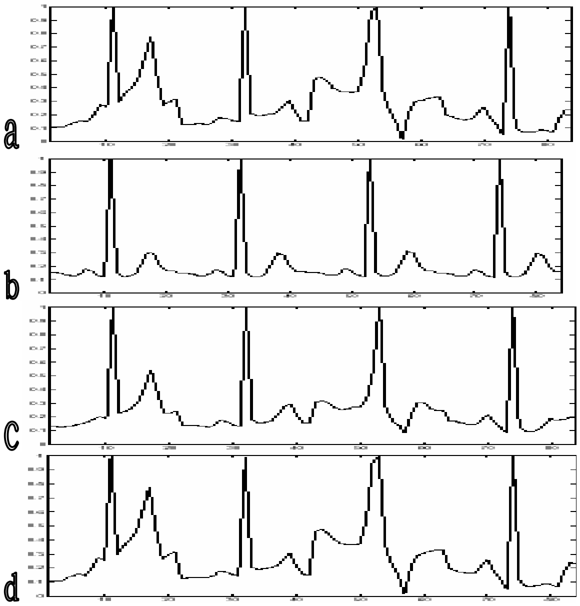


Fig.7.The network compression ratio is 1.5:1.(a) the original ECG, (b) a snapshot after 20 iterations of the learning algorithm – the reconstructed pattern is the average of the training set, (c) after 300 iterations, (d) after 5000 iterations.

Two implications can be drawn: the criterion for terminating the training process shall not be based solely on network error. It should be possible to improve the training algorithm by modifying the error criterion and including the variance ratio as part of the cost function.

3 The Wiener Filter

The filters described in literature [1], [5] can take into account only limited information about the temporal or spectral characteristics of the signal and noise processes. They are often labeled as ad hoc filters: one may have to try several filter parameters and settle upon the filter that appears to provide a usable result. The output is not guaranteed to be the best achievable result, because it is not optimized in any sense. For designing an optimal filter there is necessary to remove noise from the signal, given that the signal and noise processes are independent, stationary and random processes. We have to assume that the desired or ideal characteristics of the uncorrupted signal are known. The noise characteristics may also be assumed to be known. Wiener filter theory provides for optimal filtering by taking into account the statistical characteristics of the signal and noise processes. The filter parameters are optimized with reference to a performance criterion. The output is guaranteed to be the best achievable result under the condition imposed and the information provided. The Wiener filter is a powerful conceptual tool that changed traditional approaches to signal processing [1].

Considering the application of filtering a biomedical signal to remove noise, let us limit ourselves to a single-input, single-output, FIR filter with real input signal values and real coefficients. Fig.8. shows the general signal-flow diagram of a transversal filter with coefficients or tap weights w_i, i=0,1,2,…,M-1, input $x(n)$ and output $\widetilde{d}(n)$ [1]. The output is usually considered to be an estimate of some *desired* signal $d(n)$ that represents the ideal, uncorrupted signal, and is, therefore, indicated as $\widetilde{d}(n)$. If we assume for the moment that the desired signal is available, we could compute the *estimation error* between the output and the desired signal as

$$e(n) = d(n) - \widetilde{d}(n). \qquad (1)$$

Since $\widetilde{d}(n)$ is the output of a linear FIR filter, it can be expressed as the convolution of the input $x(n)$ with the tap-weight sequence w_i as:

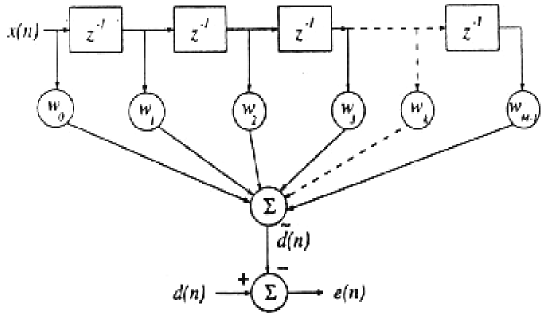


Fig.8. Block diagram of the Wiener filter.

$$\widetilde{d}(n) = \sum_{k=0}^{M-1} w_k x(n-k). \qquad (2)$$

For easier handling of the optimization procedures, the tap-weight sequence may be written as an Mx1 tap-weight vector:

$$\mathbf{w}=[w_0, w_1, w_2, \ldots, w_{M-1}]^T, \qquad (3)$$

where the bold-faced character **w** represents a vector and the superscript T indicates vector transposition. As the tap weights are combined with M values of the input in the convolution expression, we could also write the M input values as an Mx**1** vector:

$$x(n) = [x(n), x(n-1), \ldots, x(n-M+1)]^T \qquad (4)$$

The vector $\mathbf{x}(n)$ varies with time, at a given instant n the vector contains the current input sample $\mathbf{x}(n)$ and the preceding (M-1) input samples from x(n-1) to x(n-M+1). The convolution expression in equation (2) may now be written in a simpler form as the inner or dot product of the vectors **w** and $\mathbf{x}(n)$:

$$\widetilde{d}(n) = \mathbf{w}^T\mathbf{x}(n)=\mathbf{x}^T(n)\mathbf{w}=<\mathbf{x},\mathbf{w}>. \qquad (5)$$

The estimation error is then given by

$$e(n) = d(n) - \mathbf{w}^T\mathbf{x}(n). \qquad (6)$$

Wiener filter theory estimates the tap-weight sequence that minimizes the MS (mean square) value of the estimation error; the output could then be called the minimum mean-squared error (MMSE) estimate of the desired response, the filter being then an optimal filter. The mean-squared error (MSE) is defined as

$$\boldsymbol{J}(\mathbf{w}) = E[e^2(n)]=E[\{d(n)\text{-}\mathbf{w}^T\mathbf{x}(n)\} \{d(n)\text{-}\mathrm{x}^T(n)\mathbf{w}\}]= E[d^2(n)]\text{-}\mathbf{w}^T E[\mathbf{x}(n)d(n)] \text{-}E[d(n)\mathbf{x}^T(n)]\mathbf{w} + \mathbf{w}^T E[\mathbf{x}(n)\mathbf{x}^T(n)]. \qquad (7)$$

Note that the expectation operator is not applicable to **w** as it is not a random value. Under the assumption that the input vector **x**(n) and the desired response $d(n)$ are jointly stationary, the expectation expressions in the equation above have the following interpretations [1]: $E[d^2(n)]$ is the variance of $d(n)$, written as σ^2_d with the further assumption that the mean of $d(n)$ is zero; $E[x(n)d(n)]$ is the cross-correlation between the input vector $\mathbf{x}(n)$ and the desired response $d(n)$, which is an Mx1 vector:

$$\Theta = E[x(n)d(n)]. \qquad (8)$$

Note that $\Theta = [\theta(0), \theta(-1), \ldots, \theta(1-M)]^T$, where

$$\theta(-k) = E[x(n-k)d(n)], k = 0,1,2,\ldots, M-1. \qquad (9)$$

$E[d(n)x^{\mathrm{T}}(n)]$ is simply the transpose of $E[x(n)d(n)]$; therefore

$$\Theta^T = E[d(n)x^T(n)] \qquad (10)$$

$E[x(n)x^{\mathrm{T}}(n)]$ represents the autocorrelation of the input vector $\mathbf{x}(n)$ computed as the outer product of the vector with itself, written as

$$\Theta^T = E[x(n)x^T(n)] \qquad (11)$$

Setting this expression to zero, we obtain the condition for the optimal filter as

$$\Phi\mathbf{w}_0 = \Theta. \qquad (12)$$

This equation is known as the *Wiener-Hopf* equation. It is also known as the *normal equation* as it can be shown that [1], for the optimal filter, each element of the input vector $\mathbf{x}(n)$ and the estimation error $\mathbf{e}(n)$ are mutually orthogonal and furthermore, that the filter output $\widetilde{d}(n)$ and the error are mutually orthogonal. The optimal filter is obtained as

$$\mathbf{w}_0 = \Phi^{-1}\Theta. \qquad (13)$$

Applying the Fourier transform to the equation above, we get

$$W(\omega)S_{xx}(\omega) = S_{xd}(\omega), \qquad (14)$$

Which may be modified to obtain the Wiener filter frequency response $W(\omega)$ as

$$W(\omega)=\frac{S_{xd}(\omega)}{S_{xx}(\omega)} \tag{15}$$

where $S_{xx}(\omega)$ is the power spectral density (PSD) of the input signal and $S_{xd}(\omega)$ is the cross-spectral density (CSD) between the input signal and the desired signal. The frequency response of the Wiener filter may be obtained by modifying equation (15) by taking into account the spectral relationships $S_{xx}(\omega)=S_d(\omega)+S_\eta(\omega)$ and $S_{xd}(\omega)=S_d(\omega)$ which leads to

$$W(\omega)=\frac{S_d(\omega)}{S_d(\omega)+S_\eta(\omega)}=\frac{1}{1+\frac{S_\eta(\omega)}{S_d(\omega)}}. \tag{16}$$

where $S_d(\omega)$ and $S_\eta(\omega)$ are the PSDs of the desired signal and the noise process, respectively. Designing the optimal filter requires konwledge of the PSDs of the desired signal and the noise process.

We have designed a Wiener filter to remove the artifacts in the ECG signal. The equation of the desired filter is given in equation (15). The required PSD model may be obtained as follows. We created a piece-wise linear model of the desired version of the signal by concatenating linear segments to provide P, QRS and T waves with amplitudes, durations and intervals similar to those in the given noisy ECG signal. We computed the PSD of the model signal. We selected a few segments from the given ECG signal that are expected to be iso-electric; we computed in MATLAB their PSDs and obtained their average. The selected noise segments should have zero mean or have the mean subtracted out. Finally, we compared the results of the Wiener filter with those obtained by synchronized averaging and lowpass filtering. We have obtained following characteristics Fig.9, Fig.10, Fig.11, Fig.12.

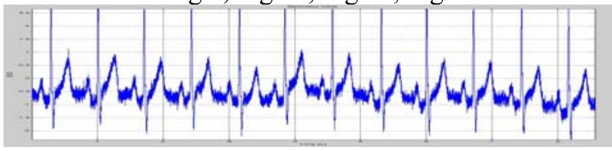


Fig.9. Initial noisy ECG signal.

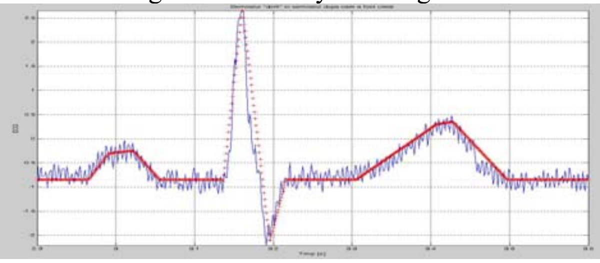


Fig.10. Desired signal approximation at the filter output.

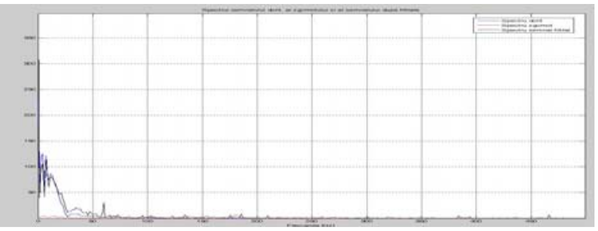


Fig.11. Desired spectrum – blue, noise spectrum-red, filtered signal – black.

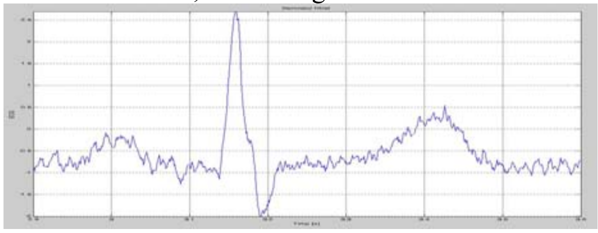


Fig.12. Cardiac cycle after Wiener filtering.

4 Conclusion

The most important point to observe here is that the filter was derived with models of the noise and signal processes (PSDs), which were obtained from the given signal itself in the present application. No cutoff frequency was required to be specified in designing the Wiener filter, whereas the Butterworth filter requires the specification of a cutoff frequency and filter order. Most signal acquisition systems should permit the measurement of at least the variance or power level of the noise present. A uniform PSD model can easily be derived. Models of the ideal signal and the noise processes may also be created using parametric Gaussian or Palladian models either in the time domain or directly in the frequency domain.

References:

[1] Rangayyan R.M., *Biomedical Signal Analysis*, Wiley-Intersciene, John Wiley & SONS, INC., 2002.

[2] Bishop C.M., *Neural Networks for Pattern Recognition* , Clarendon Press, 1995.

[3] Hamilton D., Sandham W., McQueen J., ECG Processing, *IEE Electronics Systems and Software*, 2005, pp.24-29.

[4] Aldroubi A., Unser M., *Wavelets in Medicine and Biology*, Publishing House CRC Press, 1996.

[5] Olansen J., Rosow E., *Virtual Bio-Instrumentation, Biomedical, Clinical and Healthcare applications in LabVIEW,* Publishing House Prentice Hall PTR, 2002.

[6] Tarassenko L., *A guide to neural computing applications,*Hodder Arnold Publication, 1998.