

Processing Different Format Waveforms Across Various Databases Using Matlab

Priyanka Mayapur

B.E Student, Dept. Of Electronics and Communications Engineering, Agnel Institute of Technology and Design, Assagao, Goa, India

Abstract- Signal Processing could be stated as a science that enhances our ability to communicate, understand and share the data, thus helping us extract this information for effective studies. Not being able to access a comprehensible format data could turn out to be the breaking point for a key factor or a principal research. This following paper proposes specific algorithms or rather techniques on how to convert such incomprehensible data into a comprehensive fact using four different formats with an accuracy of 99.99%. One such signal chosen for reference here is Electrocardiogram (ECG) and the plotting and processing have been implemented using MATLAB. We have specifically chosen Lead-II ECG Configuration for our research work. This technique is applicable for any real-time general waveform that exists in these specific formats as discussed in the paper.

Index Terms: — Data Format Conversion, ECG, Matlab Processing, Ontology, Signals.

I. INTRODUCTION

The biggest challenge for researchers working in their respective domain arises when an attribute of processing a particular category signal is challenged by the signal processing limitations being the current limitation to science and medicine which could be risky. [3] [4]. An emerging tool in today's era is the computer processing of any kind of information that has replaced the traditional technique of visual analysis, thus evaluating data faster and comprehensively. Therefore the work devoted in this paper deals with the biomedical signals and specific techniques to read and process a particular waveform available in four different formats taken across various databases where in the applied technique could be used for any general signal or waveform that needs to be accessed and processed in Matlab.

II. THE ELECTROCARDIOGRAM (ECG)

ECG, invented by Willem Einthoven is an advanced technique that records the bioelectrical signals originated from the heart for diagnostic purposes. [1] [2]. The 12-lead ECG configuration, out of which, Lead-II ECG Configuration as it is most useful lead for detecting cardiac arrhythmias and P Mitrale as it lies close to the cardiac axis and allows the best view of P and R waves. A normal tracing of ECG consist the following attributes as mentioned in Fig 1.

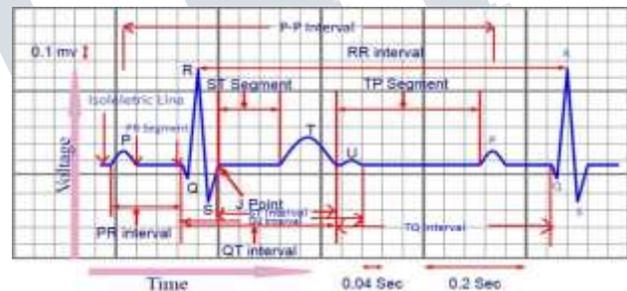


Fig. 1. A General ECG Waveform

III. ECG DATABASE COLLECTION

The databases used in our paper have been collected from ECG specific devices and from database banks available online.

A. The Physionet Bank

The PhysioNet bank is a very good place to look out for, for a start. PhysioNet is a data dictionary which offers free web access to large collections of recorded signals saved in specific formats and is involved in the ECG database. The data is shared through the PhysioNet bank. The PhysioNet contains three components: PhysioBank, PhysioToolkit and PhysioNetworks. PhysioBank being a biomedical research community includes digital recording of Physiologic Signals, time series and related data. PhysioToolkit is a software library used for processing and analyzing physiological signals and detection of physiologically significant event. PhysioNetworks is a component of the PhysioBank and PhysioToolkit that represents virtual laboratory [7]. The raw ECG database collected from

**International Journal of Engineering Research in Electronics and Communication
Engineering (IJERECE)
Vol 5, Issue 12, December 2018**

physionet.org is available in the .mat format in form which consist a data of 2 rows and 3600 columns sampled at a particular frequency [11].

ECG signals required for analysis were collected from internationally recognized databases [9]:

- PhysioNet MIT-BIH DB: Massachusetts Institute of Technology– Beth Israel Hospital Arrhythmia Database
- AHA DB: The American Heart Association ECG Database
- ESC DB: The European Society of Cardiology ST-T Database



Fig. 2. PhysioNet Structure and Signal from MIT-BIH (Left to Right)

Annotated ECG signals collected from MIT-BIH are described by three things: a text header file (.hea), a binary file (.dat) and a binary annotation file (.atr).

Header file consists of detailed information such as number of samples, sampling frequency, format of ECG signal, type of ECG leads and number of ECG leads, patient's history and the detailed clinical information. In binary data signal file, the signal is stored in 212 formats which mean each sample requires number of leads times 12 bits to be stored. The binary annotation file consists of beat annotation [8].

B. The UCI Database

UCI is basically a machine learning repository where the data is stored in the form of .dat or .txt format. The statistical characters of a given person are captured in the records of UCI Arrhythmia Dataset.



Fig. 3. UCI Machine Learning Repository

The UCI Arrhythmia Dataset consists of 452 records with 279 attributes per record. Each record is assigned to 1 of 16 classes: a class label of 1 indicates normal ECG patterns while a class label between 2 to 16 indicates abnormal ECG patterns or arrhythmia of varying types. [10].

C. The ECG Simulator

The aim of a MATLAB based ECG simulator is to typically produce the normal lead II ECG waveforms of different leads and as many arrhythmias as possible as it saves a lot of time and removes difficulties of taking the real ECG signals

with invasive and non-invasive methods. Any given ECG waveform can be simulated using the ECG simulator [12]. The ECG Electrocardiogram Amplifier records electrical activity generated by the heart and will reliably record ECG from humans, animals and isolated organ preparations and amplifies the signal provided to it (Fig 5a).



Fig. 4. The ECG Simulator

D. Electrocardiograph

An electrocardiograph is a machine (Fig 5b) used to perform electrocardiography using electrodes placed on the skin, as a result producing an electrocardiogram (ECG).



Fig. 5. An ECG Amplifier (a) and ECG Machine (b)

IV. MATLAB

The MATLAB platform is optimized for solving engineering, biomedical, scientific problems and computational mathematics. It integrates numerical analysis, matrix computation, signal processing and graphics in an easy-to-use environment, where problems and solutions are expressed just as they are written mathematically, without traditional programming. [34] [35].

V. THE FORMAT ONTOLOGY

The various formats collected for processing the data were .mat, .csv, .xml, .dat /.txt.

At the end, all the formats were converted to .mat, processed, analyzed, evaluated and plotted on MATLAB R2016.

A. The .MAT Format

MAT-files are binary MATLAB files that store workspace variables. Starting with MAT-file Version 4, there are several subsequent versions of MAT-files that support an increasing set of features. MATLAB releases R2006b and later all support all MAT-file versions. By default, all save operations create Version 7 MAT-files. The only exception

**International Journal of Engineering Research in Electronics and Communication
Engineering (IJERECE)
Vol 5, Issue 12, December 2018**

to this is when you create new MAT-files using the matfile function. [13]

B. The .CSV Format

In computing, a comma-separated value (CSV) file stores tabular data (numbers and text) in plain text. Each line of the file is a data record. Each record consists of one or more fields, separated by commas. The use of the comma as a field separator is the source of the name for this file format [36].

C. The .DAT Format

TXT is a file extension for a text file, used by a variety of text editors. Text is a human-readable sequence of characters and the words they form that can be encoded into computer-readable formats. As ASCII file is a raw text (.txt) or data (.dat) file that contains the raw data without formatting such as variable and value levels. You can save the data in a delimited format such as .csv [37].

D. The .XML Format

XML is a file extension for an Extensible Markup Language (XML) file format used to create common information formats and share both the format and the data on the World Wide Web, intranets, and elsewhere using standard ASCII text. XML is similar to HTML. An XML file can be processed purely as data by a program or it can be stored with similar data on another computer or it can be displayed, like an HTML file. [38].

VI. FLOWCHART MODEL

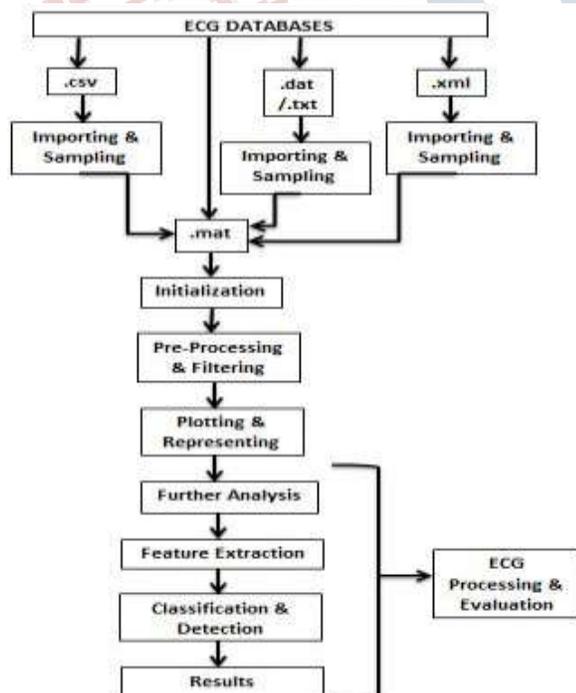


Fig. 6. Flowchart of the Proposed Methodology

VII. THE PROPOSED METHODOLOGY STEPS

To attain the objectives of this research work, several steps were followed which are explained below:

A. .MAT PROCESSING AND PLOTTING

The sample databases collected from physionet.org were exported in the .mat format in the desired lead and length. Having the primary format ready, there wasn't a necessity of conversion in this case as the .mat format is the basic format required for a signal to be in, in order to plot and process it in Matlab.

- a) First of all, in order to plot a signal, we have to make sure that signal is saved in the right folder. Once confirmed, we need to specify the correct path for the signal and then plot it.
- b) While plotting, it could be so that, some signals could be plotted directly, but most of them required re-dimensioning of the variable matrix and initialization of the signal.
- c) Dimensioning and Re-dimensioning of the variable matrix could be done by defining a variable size matrix using a constructor.
- d) If the signal is raw, which usually is unless it's taken from a filtered database, we need to perform initialization and remove the base and gain by using the following formula:

$$Xi = \frac{Xi - Base}{Gain} \quad (1)$$

Where Xi= ECG Sample

Base= Baseline Value

Gain= Gain Factor

e) Base and Gain provide specific information about the signal so to convert raw units into physical units (default units); we need to carry out this step. Once done, now it was time for plotting, i.e. loading and reading the signal.

f) The values of the particular signal were loaded into a variable by choosing a particular sampling frequency and time and now this had to be plotted which was done by using a specific command used in Matlab:

```
var1=load('C:\Users\hp\Downloads\ECG\Sample1.mat');
Plot(var1)
```

g) This will get the signal plotted in Matlab which would consist of two overlapped ECG. This is splitted into a single signal, where the amplitude and samples or time is represented by the y and x axis respectively. You can then name and mark the various attributes using legends and color markers.

h) Then it would be followed by pre-processing the signal wherein the noise such as Power Line interference, Electrode Pop, EMG noise, Baseline Wandering, Patient electrode artifacts are removed or suppressed using specific

**International Journal of Engineering Research in Electronics and Communication
Engineering (IJERECE)
Vol 5, Issue 12, December 2018**

filters in order to extract the required information from the signal and for noise reduction. This could be done either by performing Amplitude Normalization where in each sample of signal is divided from max of absolute value of signal in order to limit signal dynamic range from -1 to 1, i.e.

$$\text{Variable} = \frac{x_i}{\max(|x|)} \quad (2)$$

Where x_i = ECG Sample at a point
 x = ECG Sample

Or by filtering the signal using specific filters. Later, this processed ECG or any desired signal could be used for further evaluation depending upon the specific objective and algorithms and also using specific thresholds.

In order to sum it up, we further implemented our paper in order to detect and classify the various cardiac peculiarities and analyze the abnormalities from Lead-II ECG Configuration. First the basic steps were carried out as mentioned above, and to add to it, this was followed by feature extraction where the morphological as well as dynamic features using certain algorithms, Pan Tompkins being one, were found out followed by classification using classifiers.

B. CSV PROCESSING AND PLOTTING

After analyzing the data from physionet.org, now we extracted the signals from the Simulator, first the Simulator was connected to the ECG Amplifier so as to amplify the signal and then connected to the DSO followed by connecting this DSO to the PC using a USB cable wire. A Digital Oscilloscope (DSO) is a complex electronic device composed of various software and electronic hardware modules that work together to capture, process, display and store data that represents the signals of interest of an operator [42].

The signals extracted from this device were saved in the .csv format using specific software and saved in the respective folder.

a) To plot this signal on Matlab, we need to first process the .csv signal and get it into a format that's readable by Matlab or basically into a .mat format. In our case, this was done by first importing the data into a variable.

```
var2 = importdata ('Sample2.csv')
```

b) Once imported, reshape the data and store it in another variable which would be used in later equations.

c) Then, to make the signal free from noise or artifacts, Butterworth Filters were used. Infact, they had to be designed.

d) In order to design the Butterworth High Pass Filter, first a distinct sampling frequency (F_s1), a cut-off frequency (F_c1) and a selective order ($N1$) were chosen and initialized.

e) This was then followed by designing the filter by using the filter specification object called as the `fdesign.response`.

```
filtSpecs = fdesign.response
```

Where `fdesign.response` returns a filter specification object, `filtSpecs`, of filter response response.

The response could be a highpass, lowpass or passband or stopband filter.

f) This would be then followed by constructing an object and calling the Butter method and obtaining the filter coefficients using a `fdtool` (Open Filter Design and Analysis Tool used to design, analyze and modify filter designs). Here a slight modification has been adapted as shown below:

```
filtSpecs = fdesign.response(spec)
```

Wherein, the `spec` specifies the variables used to define a filter design, such as the passband frequency or the stopband attenuation.

g) To get a clear a clear view of the format used, we first constructed an object called as `fdesign.highpass` that constructs a highpass filter specification object `var3` and sets the specification property to `specs` and called the method as shown below:

```
var3 = fdesign.response (specs);  
var3 = fdesign.highpass (N, F3dB',N, Fc1, Fs1);
```

Where N = Filter order

F_{3dB} = cutoff frequency for the point 3dB point below the passband value. It is specified in normalized frequency units.

F_{c1} = Cutoff Frequency

F_{s1} = Sampling Frequency

h) This was then followed by finding out the filter coefficients, the values of which were again assigned to a variable.

i) Since the output obtained from this filter would be a bit distorted, therefore, the same output was again passed through a low pass filter.

j) Once again the same procedure was followed while designing the Butterworth Low Pass Filter, first a distinct sampling frequency (F_{s2}), a cut-off frequency (F_{c2}) and a selective order ($N2$) were chosen and initialized.

k) An object called `fdesign.lowpass` was constructed and the Butter method was called where a lowpass filter specification object `var4` was created and specification property to `specs` was set as shown below:

International Journal of Engineering Research in Electronics and Communication
Engineering (IJERECE)
Vol 5, Issue 12, December 2018

```
var4 = fdesign.response (specs);  
var4 = fdesign.lowpass ('N2, F3dB', N2, Fc2, Fs2);
```

- l) Once again the filter coefficients were found and stored in a variable.
- m) At the end of all this Zero Phase Filtering procedure, finally the wave was now being plotted.
- After this, the normal procedure of choosing a sampling frequency and time in order to represent and plot the signal was carried out followed by processing of the signal using various algorithms along with feature extraction and classification and also using specific thresholds.
- n) Another way which was usually seen to have been used to plot a csv signal was to use the command csvread.

```
M = csvread('filename')
```

C. DAT OR .TXT PROCESSING AND PLOTTING

The database collected from UCI Repository was extracted in .dat or .txt format.

- a) Now, in the UCI Arrhythmia Dataset, the data file was the same as a csv file. Only thing, you could save the data file, Sample3.dat as a csv file, Sample3.csv and carry out the entire same procedure as described above for processing and plotting of a csv file to convert it into the .mat format.
- b) Sampling has been done twice in order to increase the samples, so that the plotting could be done easily and clearly.
- c) Only thing, in this file, missing data is denoted as "?".
- d) First, the missing value symbols "?" are replaced with empty strings "", and the new data is saved into a new file.
- e) Then sample.read_csv() is called to read the whole dataset as a samples DataFrame. Then the first 279 columns are extracted as variable X which is the 279 features. The 280th column is stored in variable y which is the labels for the 452 records.
- f) After the above preparation, we impute the missing values, splitting the dataset into a training set and a test set.
- g) The empty strings between the delimiter "," in the CSV file is read and interpret as NaN in samples. NaN can be replaced by meaningful values by sklearn.preprocessing.Imputer.

```
X = Imputer (missing_values='NaN',  
strategy='mean', axis=0).fit_transform(X)
```

missing_values = 'NaN' means all the values encoded as np.nan will be imputed.

strategy = 'mean' means replacing the missing values using the mean along the axis.

axis = 0 means imputing along columns.

The above line of code iterates each column and imputes its missing values by the mean value of that column [10].

- h) Then the same above procedure for csv file plotting was followed in order to plot the Zero Phase Filtered signal, followed by processing the signal using specific algorithms and thresholds.
- i) It is always important to keep in mind that in order to plot a signal, we have to make sure that signal is saved in the right folder. Once confirmed, we need to specify the correct path for the signal and then plot it.
- j) Also, it was seen that the dat file could also be plotted in the similar manner as shown below:

```
fid = fopen ('D:\Sample3.dat', 'r')
```

D. XML PROCESSING AND PLOTTING

The collection of database from the ECG machine and processing it in Matlab was one of the most challenging things we faced. The ECG Machine stores the database in the form of either xml file or pdf file. Pdf file in our case wasn't really helpful enough to process except for viewing the signal. The main challenge was to process the xml file in order to obtain accurate results. Usually the ECG is recorded in an image consisting of all 12 channels, since distinct diseases manifest differently in each of the leads, it is important to isolate the different leads involved (in our case: Lead-II configuration).

- a) To begin with, as always it's important to keep in mind that, in order to plot a signal, we have to make sure that signal is saved in the right folder. Once confirmed, we need to specify the correct path for the signal and then plot it.
- b) The xml signal needs to be indexed and stored as data structure in Matlab compatible format. In other words, conversion of the file into .mat format in order to upload and read it.
- c) So, the very first step is to convert the xml data into structures. The actual data is in the form of double coded value. They need to be compressed, encoded, decompressed and decoded in order to make it machine readable and store this value in a variable.

```
var5 = xml2struct ('Sample4.xml')
```

- d) Signals are said to be in physical units when the data values represent the actual real life values as closely as possible, being aware of the fact that everything on the computer is in the form of digital and discrete data rather than analog and continuous. This would also include precious 64 bit double precision floating value points but

**International Journal of Engineering Research in Electronics and Communication
Engineering (IJERECE)
Vol 5, Issue 12, December 2018**

again this is as close as we can get, almost closer to the actual physical values, so they are referred to as physical.

e) So to begin with after converting the xml data to Matlab vectors stored in structures, now it was time to extract the Base64 encoded data along with removing structural space, newlines and line feeds.

Base64 Encoding

Base64 is an encoding scheme used to represent arbitrary sequences of octets and binary data in an ASCII format. Bytes forming the data are broken into buffers of 24 bits (3 bytes at a time). The resulting buffer of 3 bytes is then broken in 4 packs of 6 bits each. Those 6 bits form a number corresponding to the index in the character set supported by Base64 (A-Z, a-z, 0-9, + and /). [41].

f) This would be then followed by decoding the data from Base64 structure.

```

i. i = uint*(x)
ii. i = uint8(x)
```

The first equation converts the vector x into an unsigned integer (i). The second equation converts the vector x into an unsigned (1 byte) 8-bit integer (ii). This has an output range from 0 to 255 and the uint* class is primarily meant to store integer values.

Base64 Decoding

Base64 decoding is the process of transforming a base64 representation a string of odd-looking text back into the original binary or text data [40].

```

decoded = uint8(base64decode(data))
```

So in this case, the data is converted into type uint8 as this the requirement for the next part of the code.

g) Since the ECG signal consists of 12 leads, it was now time to isolate and extract each of these 12 leads by setting the leadOffset to zero in order to prevent the baseline shift.

```

leadOffset = 0;
```

h) This would be then followed by extracting the chunk header by typecasting the size of the ECG data, header, into unsigned 32 and 16 bits integer in the first two steps and then into signed header 16 bits integer.

i) Then the data size above to read the appropriate number of bytes beyond the header will be used followed by converting the 8 bit bytes into 10 bit codes.

j) Then LZW Decompression was done where in the data was compressed with 10 bit LZW codes followed by deinterleaving it into signed 16 bit integers.

LZW (Lempel-Ziv-Welch) Decompression

LZW is a universal lossless data compression algorithm that builds a string translation table from the text being compressed. [39].

k) This would be then followed by converting the lead cell array into numeric matrix and reconstructing the leads.

l) And finally after all this, it was time to plot the signal finally having all the 12 leads represented on the signal.

m) Then the same procedure of filtering was adapted as in the done for the csv file using a Butterworth filter having a frequency range of 550-1100 Hz to get a noise free signal (Zero Phase Filtering).

n) Then we extracted the Lead-II ECG from the 12 lead configuration signal and then processing of ECG was done by using specific algorithms and thresholds.

VIII. RESULTS

The results of the four different format types of processing are displayed below:

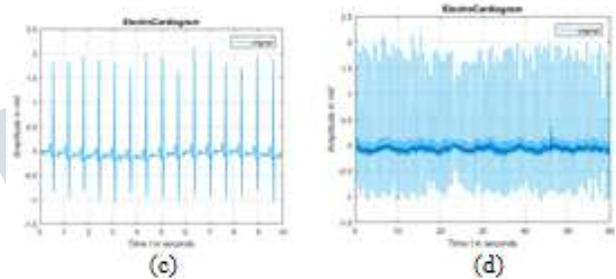


Fig. 7 Original ECG Signal Plotting (10, 60 sec) in Matlab (c, d)

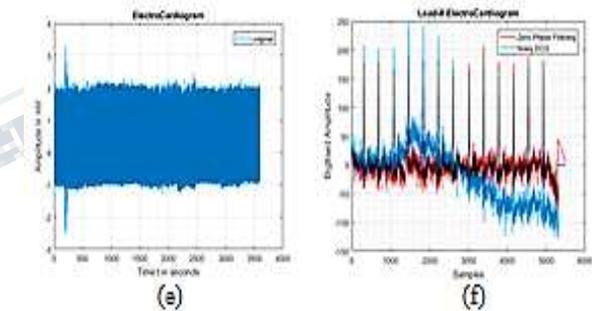


Fig. 8. Plotting of an Original ECG Signal (3060 sec) in Matlab (e) and Zero Phase Filtering of ECG Signal from (.csv) to (.mat) (f)

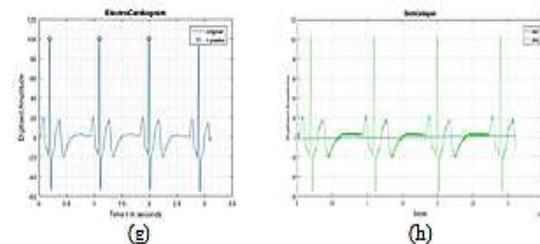


Fig. 9. Plotting of .csv (g) and .dat Signal (h)

**International Journal of Engineering Research in Electronics and Communication
Engineering (IJERECE)
Vol 5, Issue 12, December 2018**

A signal always needs to be filtered before plotting and processing else it might result in a chaotic shape and might give wrong results due to artifacts present as shown below:

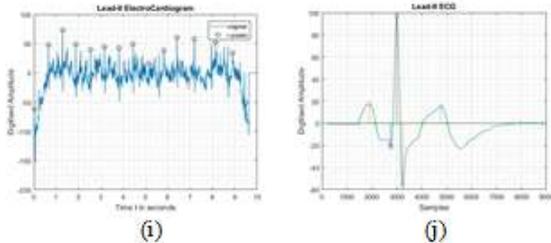


Fig. 10. Chaotic Plotted Waveform (i) and Processing of the Features (j)

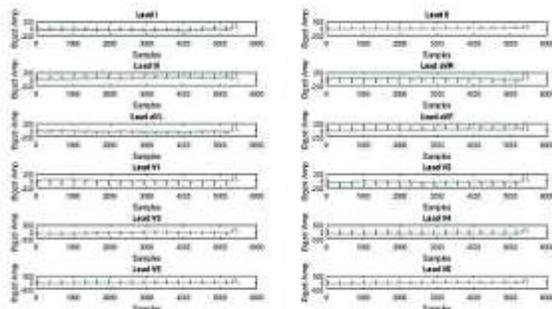


Fig. 11. Plotting of 12 Lead Configuration ECG Signal from the .xml format to .mat signal

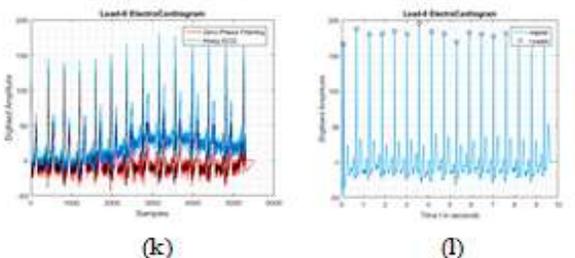


Fig. 12. Zero Phase Filtering of the .xml Signal (k) and a Lead-II ECG Configuration extracted from 12 Lead Configuration ECG signal (l)

IX. CONCLUSION

In this paper, we have used a basic methodology using an efficient tool like Matlab which is basically a software tool very helpful to scientists and researchers as it is very accurate as well data and files could be easily edited and complimented using the toolbox available, thus having an immense effect on ECG or any general Signal Processing. An algorithm or more like a technique was designed to read and process the formats like .mat, .csv, .dat or .txt and .xml in Matlab collected across various databases. And for further classification and evaluation, algorithms and

thresholds were used and applied in order to obtain a distinct result with 99.99% accuracy.

REFERENCES

- [1] Maheswari Arumugam, Arun Kumar Sangaiah, "Detection of cardiac arrhythmia from ECG signals", IIOA3 JOURNAL, 8th August 2015.
- [2] Vandana Verma, Sagar Singh Rathore, "Detection of Characteristic Points of ECG Signal", International Journal of Innovative Research in Computer and Communication Engineering, Vol. 3, Issue 11, November 2015.
- [3] Rob Theriault, Cardiac Dysrhythmia Interpretation.
- [4] Biology, Preliminary Volume 1, The Dynamic Science, Units 1 and 2, CTI Reviews.
- [5] A.Peterkova, M. Stremy, "The raw ECG signal processing and the detection of QRS complex", IEEE European Modelling Symposium, 2015 IEEE.
- [6] S.T.Sanamdikar, Dr.S.T.Hamde, Dr.V.G.Asutkar, "A Literature Review on Arrhythmia Analysis of ECG Signal", International Research Journal of Engineering and Technology (IRJET), Volume: 02 Issue: 03, June-2015.
- [7] Anusha F.G, Jeba Sheela.A, "Automatic Identification ECG Anomalous Using Xml Data Processing", International Journal of Engineering Development and Research, 2015.
- [8] Sonal Pokharkar, Amit Kulkarni, "ECG Real Time Feature Extraction Using MATLAB", International Journal of Technology and Science, Volume V, Issue 1, 2015 pp. 1-4.
- [9] Pankhuri Trivedi, Shahanaz Ayub, PhD, "Detection of R Peak in Electrocardiogram", International Journal of Computer Applications, Volume 97 No.20, July 2014.
- [10] Rui Duan, Sabah Mohammed and Jinan Fiaidhi, "ECG-based Heartbeat Classification Using Ensemble Methods", International Journal of Bio-Science and Bio-Technology Vol. 9, No. 4, (2017), pp. 1-18.
- [11] Er. J.S Dhir, Er. N.K. Panag, "ECG Analysis and R Peak Detection Using Filters and Wavelet Transform", International Journal of Innovative Research in Computer and Communication Engineering, Vol. 2, Issue 2, February 2014.

**International Journal of Engineering Research in Electronics and Communication
Engineering (IJERECE)
Vol 5, Issue 12, December 2018**

- [12] ECG Simulation using MATLAB, R. Karthik, B.E. (ECE)
- [13] Matlab 7, Data Import and Export [Online].
- [14] Mr. Vyankatesh S. Thorat, Dr. Suresh D. Shirbahadurkar, Mrs. Vaishali V. Thorat, "Classification of Cardiac Arrhythmia With Respect to ECG and HRV Signal", International Journal of Advanced Research in Computer and Communication Engineering, Vol. 5, Issue 4, April 2016.
- [15] A.R. Gomes e Silva, H.M. de Oliveira, R.D. Lins, Converting ECG and other paper legated biomedical maps into digital.
- [16] Zaw Min Naing et al, "Analysis on Conversion Process from Paper Record ECG to Computer based ECG", MOJ Applied Bionics and Biomechanics, Volume 1 Issue 2 – 2017.
- [17] Deepak Kumar Garg, Diksha Thakur, Seema Sharma, Shweta Bhardwaj, "ECG Paper Records Digitization through Image Processing Techniques", International Journal of Computer Applications (0975 – 888) Volume 48– No.13, June 2012.
- [18] Ms. P. G. Patel, Prof. J. S. Warriar, Prof. U. R. Bagal, "ECG Analysis And Detection Of Arrhythmia Using MATLAB", Vol 1, Issue 11, December, 2012.
- [19] M. K. Islam, A. N. M. M. Haque, G. Tangim, T. Ahammad, M. R. H. Khondokar, Member, IACSIT, "Study and Analysis of ECG Signal Using MATLAB LABVIEW as Effective Tools", International Journal of Computer and Electrical Engineering, Vol. 4, No. 3, June 2012.
- [20] PATRICK O. BOBBIE, CHAUDARY ZEESHAN ARIF, HEMA CHAUDHARI, SAGAR PUJARI, Electrocardiogram (EKG) Data Acquisition and Wireless Transmission, School of Computing and Software Engineering.
- [21] Adam Szczepanski, Khalid Saeed, and Alois Ferscha, "A New Method for ECG Signal Feature Extraction", L. Bolc et al. (Eds.): ICCVG 2010, Part II, LNCS 6375, pp. 334341, 2010, Springer-Verlag Berlin Heidelberg 2010.
- [22] MATLAB, The Language Of Technical Computing, The Mathworks.
- [23] Bruce, Biomedical Signal Processing And Signal Modeling.
- [24] Suresh R. Devasahayam, "Signals and Systems in Biomedical Engineering, Signal Processing and Physiological Systems Modeling".
- [25] ElectroCardioGraphy-ECG made extra easy.
- [26] Advanced ECG Interpretation.
- [27] The Normal ECG and its (Normal) Variants.
- [28] ECG Monitor, Project Id H3290.
- [29] USB TAP Probe User Guide.
- [30] Signal Processing Toolbox, For Use with MATLAB, Users Guide Version 5.
- [31] P and T Wave Detection in Electrocardiogram (ECG) Signals, Chao Lin, Seminar SC, June 2011.
- [32] in.mathworks.com
- [33] Malcolm S. Thaller, M.D, The Only EKG Book Youll Ever Need.
- [34] Studentsfocus.com
- [35] www.ardentcollaborations.com
- [36] online-converting.ru
- [37] www.convertfiles.com
- [38] whatis.techtarget.com
- [39] www.stringology.org
- [40] www.opinionatedgeek.com
- [41] tools.ietf.org
- [42] www.testandmeasurementtips.com
- [43] stackoverflow.com
- [44] www.ece.iit.edu
- [45] Giron-Sierra, Jose Maria, "Digital Signal Processing with Matlab examples".
-