ECG DATA COMPRESSION USING OPTIMUM QMF BANK

Multi Rate Signal Processing

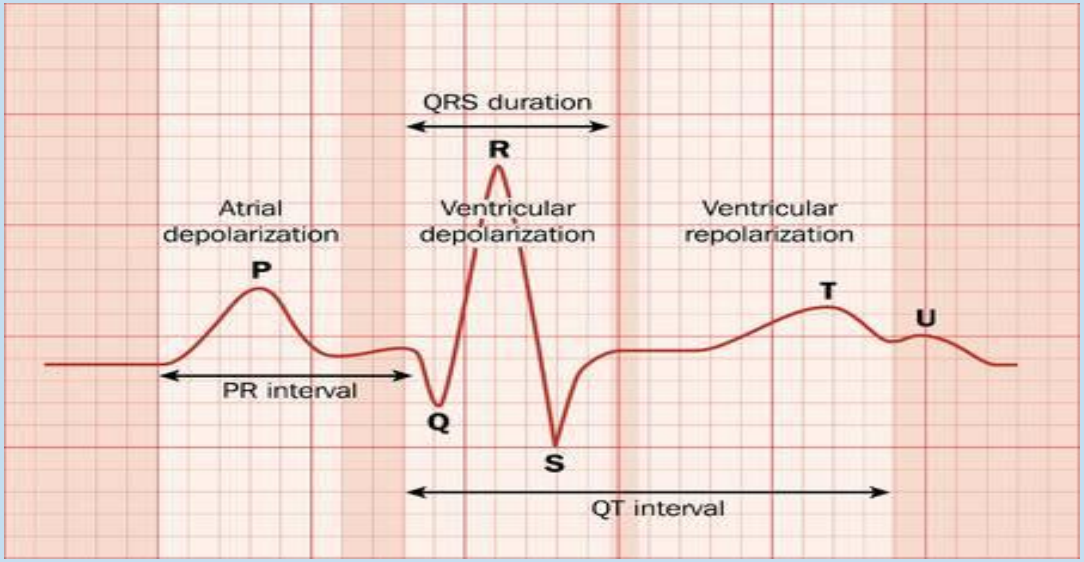
Submitted to:
Asst. Prof. Avinash Ratre

Submitted by:
Sumedha (2K19/SPD/17)
Sudiksha Shukla (2K19/SPD/25)

INTRODUCTION

Electrocardiography (E.C.G.) :

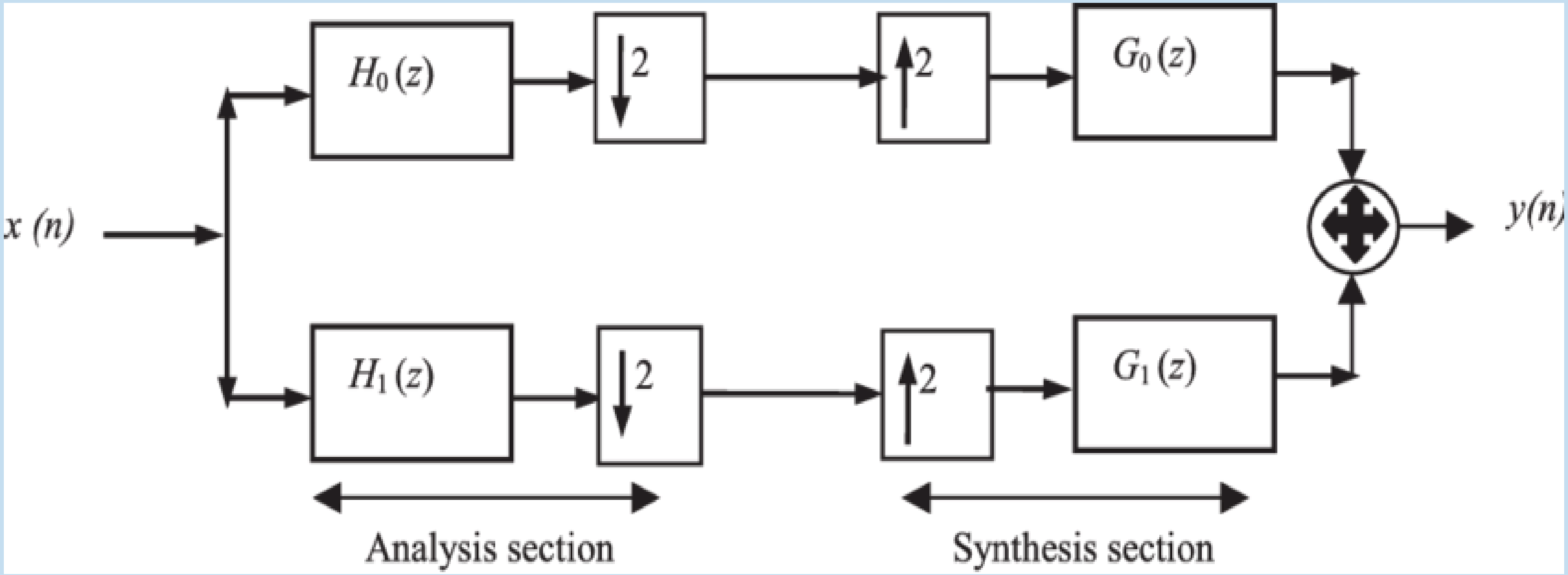
- ECG is the recording of electrical activity of the heart.
- It is a bioelectric signal generated by the human heart, by the action of depolarization and repolarization of cardiac cells.



NEED FOR ECG DATA COMPRESSION :

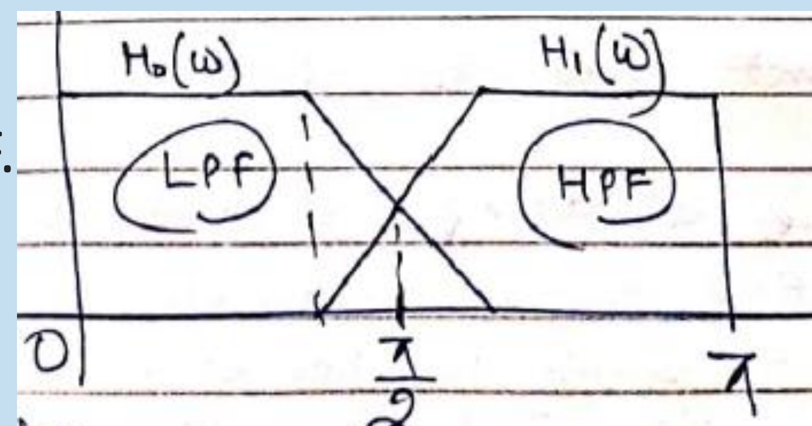
- A computerized system of biological signal processing requires a huge amount of data, it is very difficult to store and process these signals.
- Hence, we need a way to reduce the amount of data storage, and this needs to be done keeping their critical clinical content in order to rebuild the signal.
- Compressing information reduces the transport time and the storage space and reduces the memory capacity in portable systems.
- It increases the number of channels to be transferred and broadens the bandwidth.
- ECG data compression is very important because it reduces the storage requirement for future diagnosis of a cardiac patient and plays an important role in efficient transmission in telemedicine and e-health care systems.
- ECG data compression techniques are classified into two types, i.e., lossy and lossless. In lossy compression techniques, the reconstructed signal is not an exact replica of the original input signal.

Quadrature Mirror Filter :



- A **quadrature mirror filter** is a filter whose magnitude response has mirror image symmetry about π/2 frequency.
- For QMF to behave as a LTI system:

- ❑ Ho(w) = H(w) is an LPF
- ❑ H1(w) = H(w-π) -> mirror-image HPF.
- ❑ Go(w) = 2*H(w) -> LPF
- ❑ G1(w) = -2*H(w-π) -> HPF



ROLE OF QMF BANK FOR ECG DATA COMPRESSION :

- In this project data compression technique is based on the optimum two channel quadrature mirror filter (QMF) bank .

- Data compression is done by decomposing the signal using optimum QMF bank and truncating the irrelevant coefficients using level thresholding.

- Here, a Kaiser window is used to obtain the coefficients of the filter bank.

- Linear optimization technique is employed for optimization of filter coefficients.

- Run-length encoding is used to improve the compression without loss of significant information.

- Following conditions need to be fulfilled :

a) Suitable optimization is needed to design a QMF bank for avoiding amplitude distortion.
b) FIR filter must be chosen for removing phase distortion.
c) All the analysis and synthesis filters must be designed in such a way so that no aliasing distortion is produced.

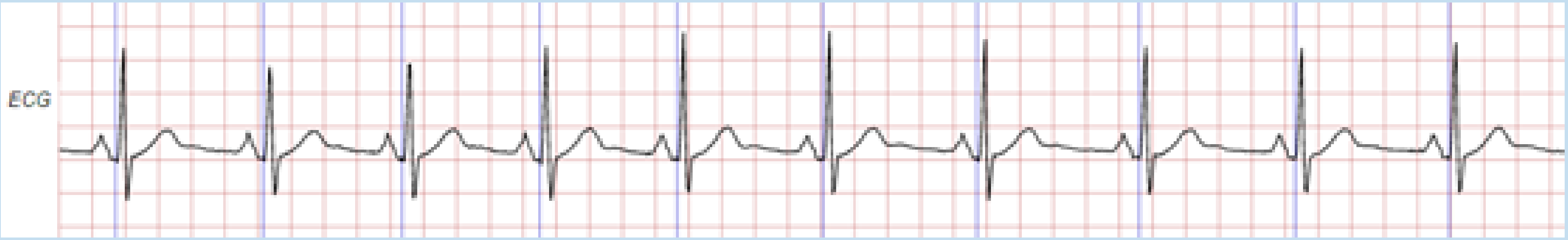
DATASET USED :

Link for Dataset: https://www.physionet.org/static/published-projects/apnea-ecg/apnea-ecg-database-1.0.0.zip

Details:

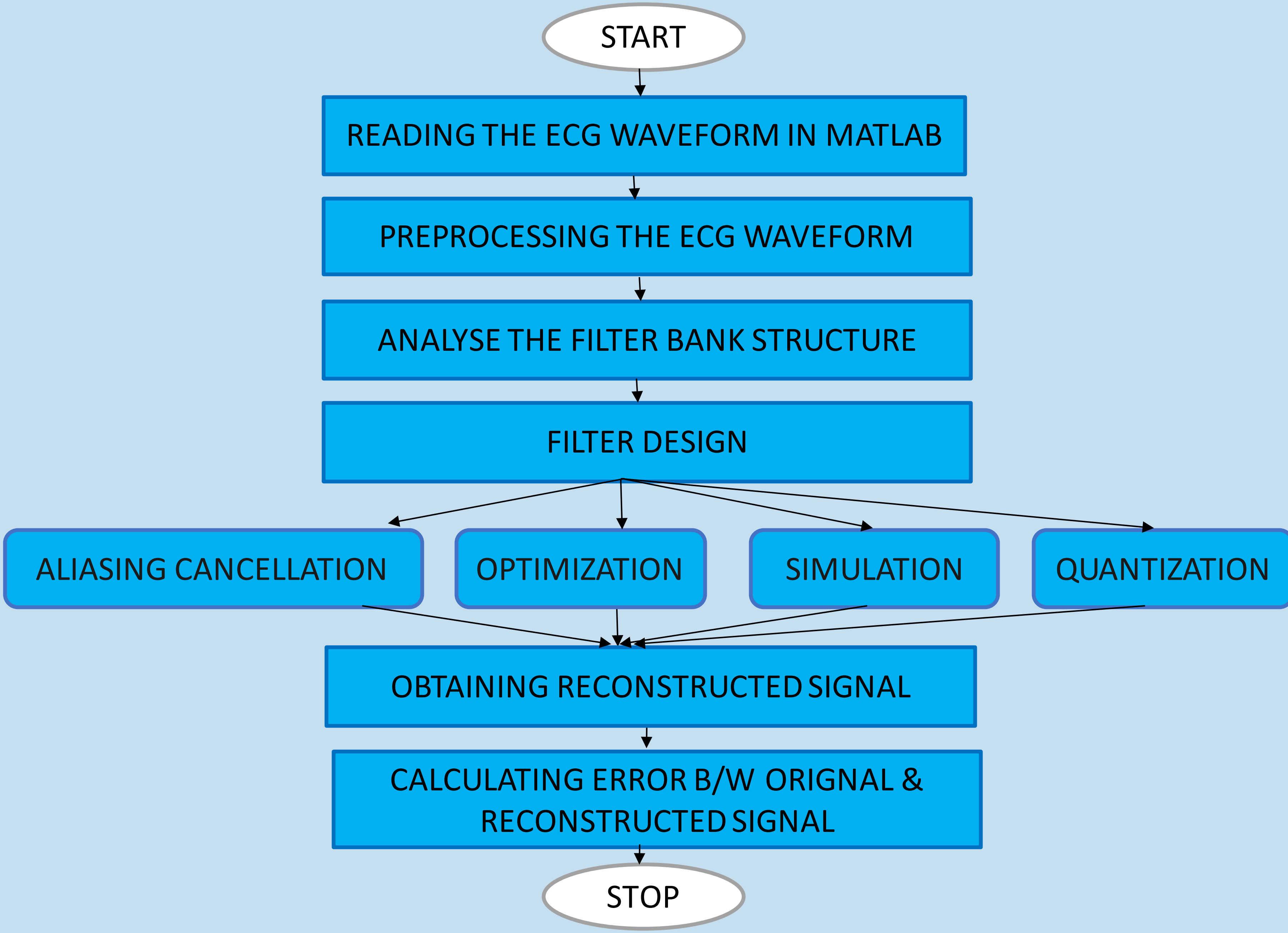
- The data in this directory have been contributed by Dr. Thomas Penzel of Phillips-University, Marburg, Germany.
- The data consist of 70 records, divided into a learning set of 35 records , and a test set of 35 records.
- Recordings vary in length from slightly less than 7 hours to nearly 10 hours each.
- Each recording includes a continuous digitized ECG signal
- The files with names of the form rnn.dat contain the digitized ECGs
- The .hea files are header files that specify the names and formats of the associated signal files
- The qrs files are machine-generated (binary) annotation files

DATASET VISUALIZATION :



Record length	08:12:50
Clock frequency	100 ticks per second
Annotator: apn	(489 annotations)
	A 470
	N 19
Annotator: qrs	(29938 annotations)
	N 29938
Signal: ECG	1 tick per sample; 200 adu/mV; 12-bit ADC, zero at 0; baseline is 0

FLOWCHART


START
READING THE ECG WAVEFORM IN MATLAB
PREPROCESSING THE ECG WAVEFORM
ANALYSE THE FILTER BANK STRUCTURE
FILTER DESIGN
ALIASING CANCELLATION
OPTIMIZATION
SIMULATION
QUANTIZATION
OBTAINING RECONSTRUCTED SIGNAL
CALCULATING ERROR B/W ORIGNAL & RECONSTRUCTED SIGNAL
STOP

READING THE ECG WAVEFORM IN MATLAB

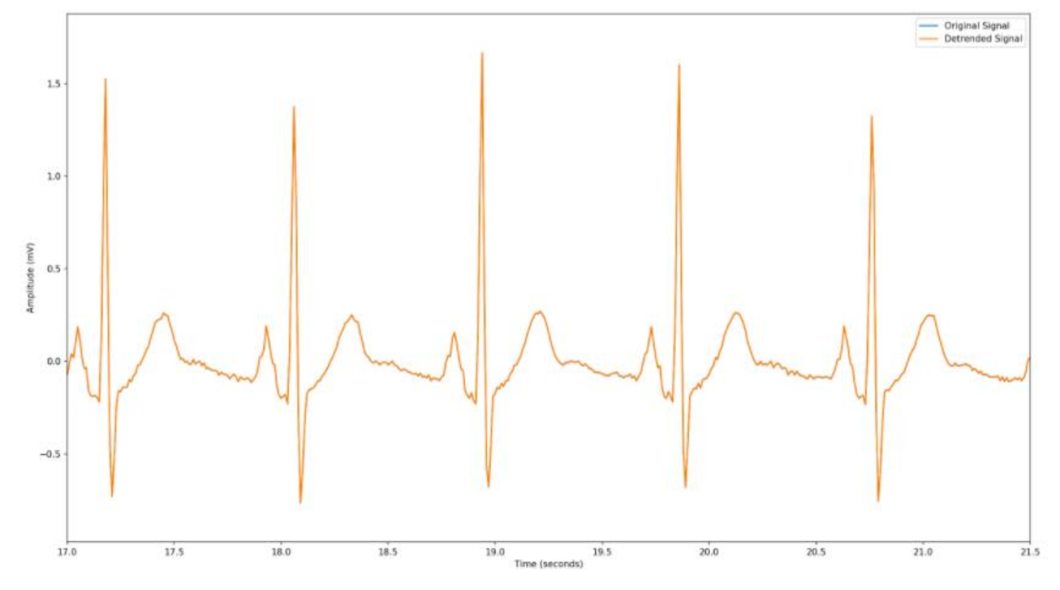
Code:

```
folder = fullfile(matlabroot,'toolbox','matlab','audiovideo', ...
         ["chirp.mat","gong.mat","train.mat","splat.mat"]);
fs = 8192;
sds = signalDatastore(folder,'SampleRate',fs);
```

```
data = readall(sds);

tiledlayout('flow')
for i = 1:length(data)
    nexttile
    fsst(data{i},fs,'yaxis')
end
```

Output:



PRE-PROCESSING RAW WAVEFORM IN MATLAB

STEPS:

- Reducing the amount of information in the signal prior to compression so that it intuitively improves the compression ratio.

- In pre-processing step we have removed the DC offset (This is generally not considered useful information)

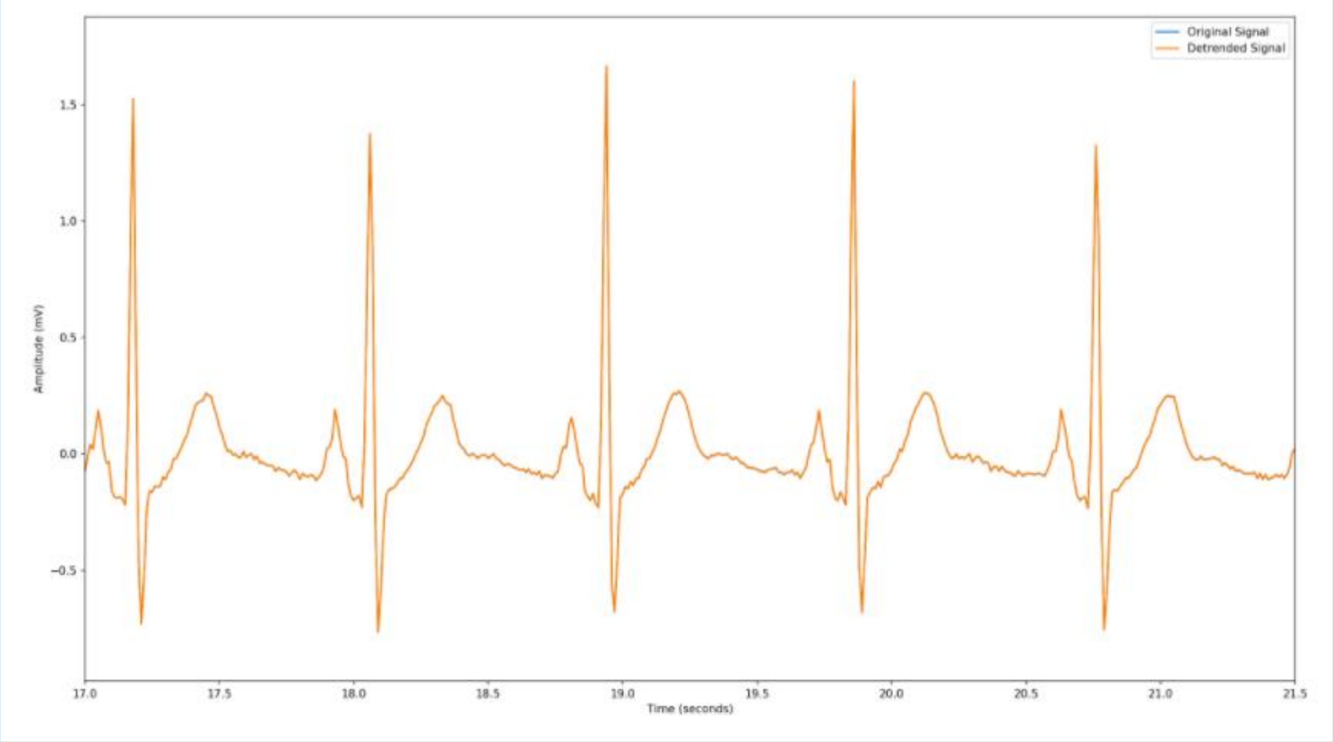
Code:

```
sds_ac = real(ifft(f));
[sds_ac, sds - mean(x)]
```

PRE-PROCESSING RAW WAVEFORM IN MATLAB

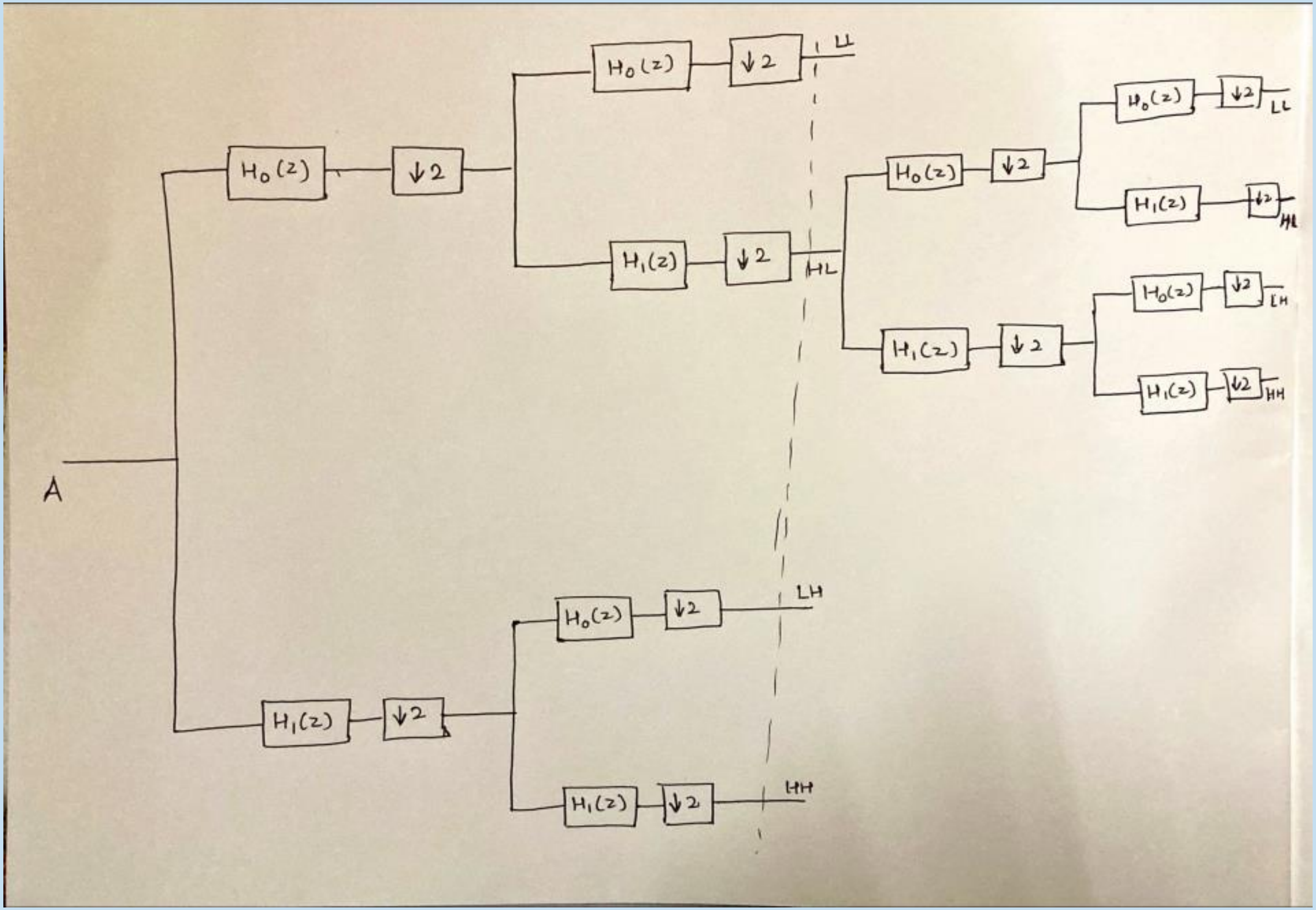
OUTPUT:

- Since the signal has very little drift or DC offset, so detrending effectively makes no difference
- Pre-processed signal is similar to original one



ANALYSIS OF FILTER BANK STRUCTURE

The filter bank tree structure is as follows:



FILTER DESIGN

- The basic idea to design this system is to make the reconstruction as perfect as possible. In order to do this, we divided filter design in 3 steps

$Perfect\ Reconstruction$:

(1) $Aliasing\ Cancellation$
(2) $Phase\ Distortion$
(3) $Amplitude\ Distortion$

ALIASING CANCELLATION:

- Following conditions need to be met for aliasing cancellation:

{ $F0\ (z)$ = $H1(-z)$ $F1\ (z)$ = $-H0(-z)$
Which is equivalent to $H1\ (z)$ = $H0\ (-z)$ => $h1\ (n)$ = $(-1)\ nh0(n)$

And the cancellation requirements above become: { $F0\ (z)$ = $H0(z)$ $F1\ (z)$ = $-H0(-z)$

If we translate these into time domain, we have:
$h1\ (n)$ = $(-1)\ nh0(n)$ { $f0\ (n)$ = $h0(n)$ $f1\ (n)$ = $-(-1)\ nh0(n)$

According to this relationship, we can generate $h1\ (n)$, $f0\ (n)$, $f1\ (n)$ from $h0(n)$

FILTER DESIGN

CHOOSING KAISER WINDOW:

- We chose Kaiser window as a basic model to design $h0\ (n)$ because of the flexibility.

- Comparing to other window filters, the ripple shape of Kaiser window is much easier to control and modify with filter coefficients, N and β. The coefficients we start from are N = 29, Fc = 0.5, β = 9.

- We chose Kaiser window as a basic model to design $h0\ (n)$ because of the flexibility.

- Comparing to other window filters, the ripple shape of Kaiser window is much easier to control and modify with filter coefficients, N and β. The coefficients we start from are N = 29, Fc = 0.5, β = 9.

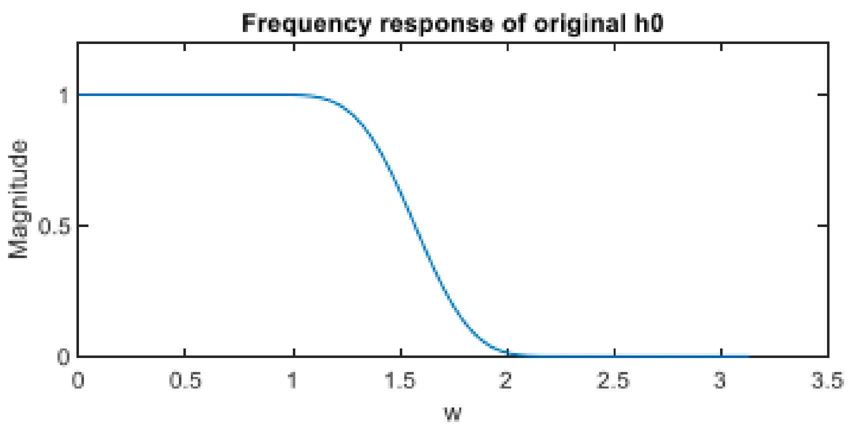


Fig. 1

FILTER DESIGN

PHASE DISTORTION:

This is not really a problem. $h0\ (n)$ is a FIR filter, so that it will not cause any phase distortion.

AMPLITUDE DISTORTION:

- We can eliminate amplitude distortion if and only if
- $H0(z)$ has the form of: $H0\ (z) = c0z\ {-2n0} + c1z\ {-(2n1+1)}$
- However, in this case, the order of $h0(n)$ filter is 1 which means if $h0\ (n)$ can offer good low pass response, amplitude distortion is inevitable.
- To minimize amplitude distortion we go for optimization

OPTIMIZATION:

- Coefficients are chosen such that Er $should\ be\ as\ small\ as\ possible$
- $Er = \sum(H\ 2\ (\omega) + H\ 2\ (\pi - \omega) - 1)\ 2\ \pi\ \omega{=}0$
- $Es = \sum\ H\ 2\ (\omega)\ \pi\ \omega{=}stopband$, $this\ refers\ to\ energy\ loss\ out\ of\ stop\ band$

FILTER DESIGN

CODE FOR ALIASING CANCELLATION:

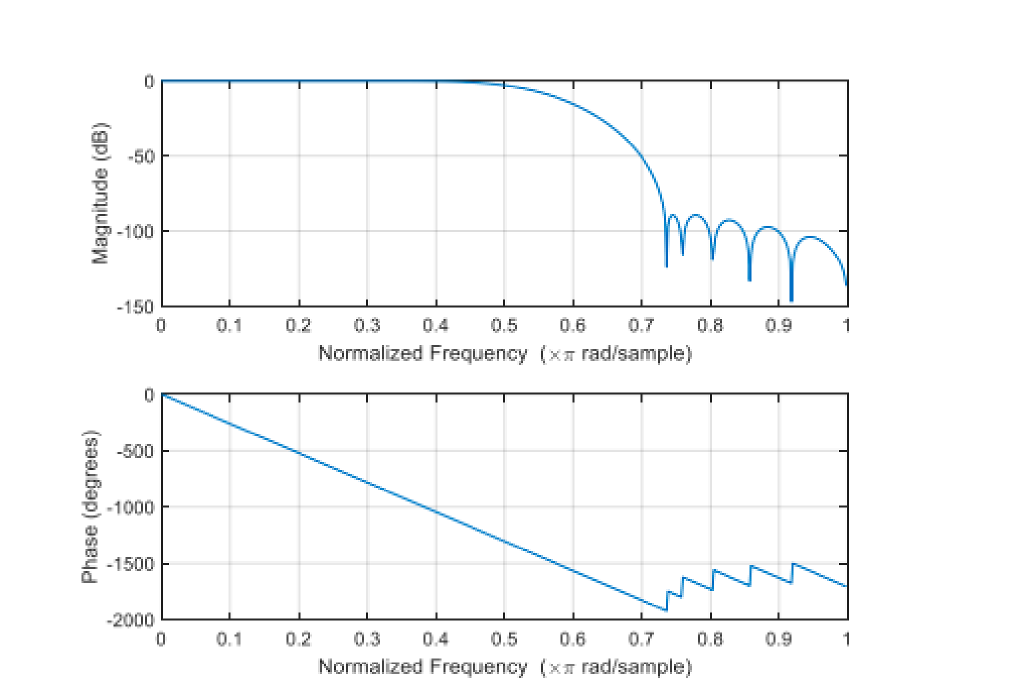
```matlab
% generate low pass filter h0 based on Kaiser window
N = 29;
Fc = 0.5363563;
flag = 'scale';
Beta = 9;
win = kaiser(N+1, Beta);
h0 = fir1(N, Fc, 'low', win, flag);
figure(1);
freqz(h0);
% generate h1
n=rem(1:30,2);
 m=1*(n>0.5)+(-1)*(n<=0.5); % times a 1,-1,1,-1... sequence
h1=h0.*m;
% intuitive plot of h0, h1, and amplitude distortion
[H0,w]=freqz(h0,1,512);
hmag0 = abs(H0);
figure(2);
subplot(2,2,1)
plot(w,hmag0);
title('Frequency Response of h0');
[H1,w]=freqz(h1,1,512);
hmag1 = abs(H1);
subplot(2,2,2)
plot(w,hmag1);
title('Frequency Response of h1');
subplot(2,2,[3 4])
plot(w,(hmag0.^2+hmag1.^2))
title('H0(w)^2+H1(w)^2');
```

FILTER DESIGN

CODE FOR OPTIMIZATION:

```
% Optimization factor Er
one=ones(512,1);
Er=sum((hmag0.^2+hmag1.^2-one).^2);
Er;
```

FREQUENCY RESPONSE AFTER OPTIMIZATION:



FILTER DESIGN

SIMULATION:

- The basic idea of simulation is to choose an input signal x(n), make it go through the QMF system and generate the output signal $\hat{x}(n)$.
- The method we used for up-sampling by 2 is inserting zeros. This will result in a magnitude reduction in time domain by 2 after the second convolution in f0 and f1 filter. So, when we calculate the absolute error between input and output, $\hat{x}(n)$ must be multiplied by 2 before deducting x(n).
- The length of x(n) (input signal) and $\hat{x}(n)$ (output signal) are different because of convolution. $\hat{x}(n)$ is 2*length(h0) pads longer than x(n). As a result, we need to choose the specific part of $\hat{x}(n)$ to calculate the absolute error.

FILTER DESIGN

CODE FOR SIMULATION AND CALCULATING ERROR BETWEEN ORIGNAL SIGNAL AND RECONSTRUCTED SIGNAL:

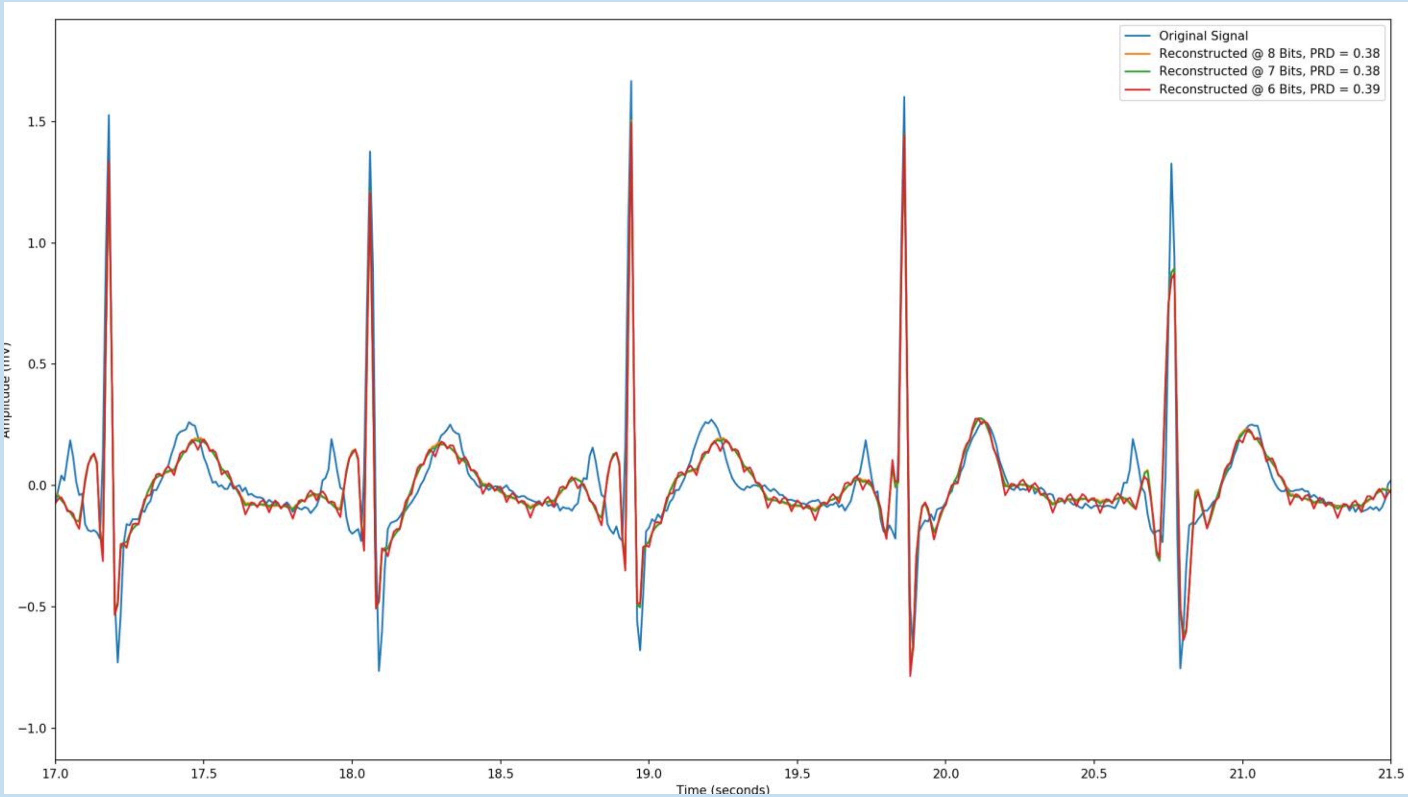
```
% Simulation
% Generate input signal x(n)
ntaps = 65;
f = [0.0 0.9 0.95 1.0];
mag = [ 1.0 1.0 0.7071 0.0];
b = fir2(ntaps, f, mag);
n1 = length(b);
len1 = 256 - n1 + 1;
data = 5*[zeros(1, 24) ones(1, 48) zeros(1, 48) -1*ones(1, 48)
zeros(1,23)];
x = conv(b, data);
% x(n) go through QMF system
u0=conv(x,h0);
v0=dyaddown(u0,1);
w0=dyadup(v0,1);
u1=conv(x,h1);
v1=dyaddown(u1,1);
w1=dyadup(v1,1);
f0=h0;
f1=-h1;
xhat=conv(w0,f0)+conv(w1,f1);
% Simulation results
figure(3);
subplot(3,1,1)
stem(x);
title('input signal x(n)');
subplot(3,1,2)
stem(xhat);
title('output signal xhat(n)');
error=abs(xhat(31:286).*2-x);
subplot(3,1,3)
```

QUANTIZATION

CODE:

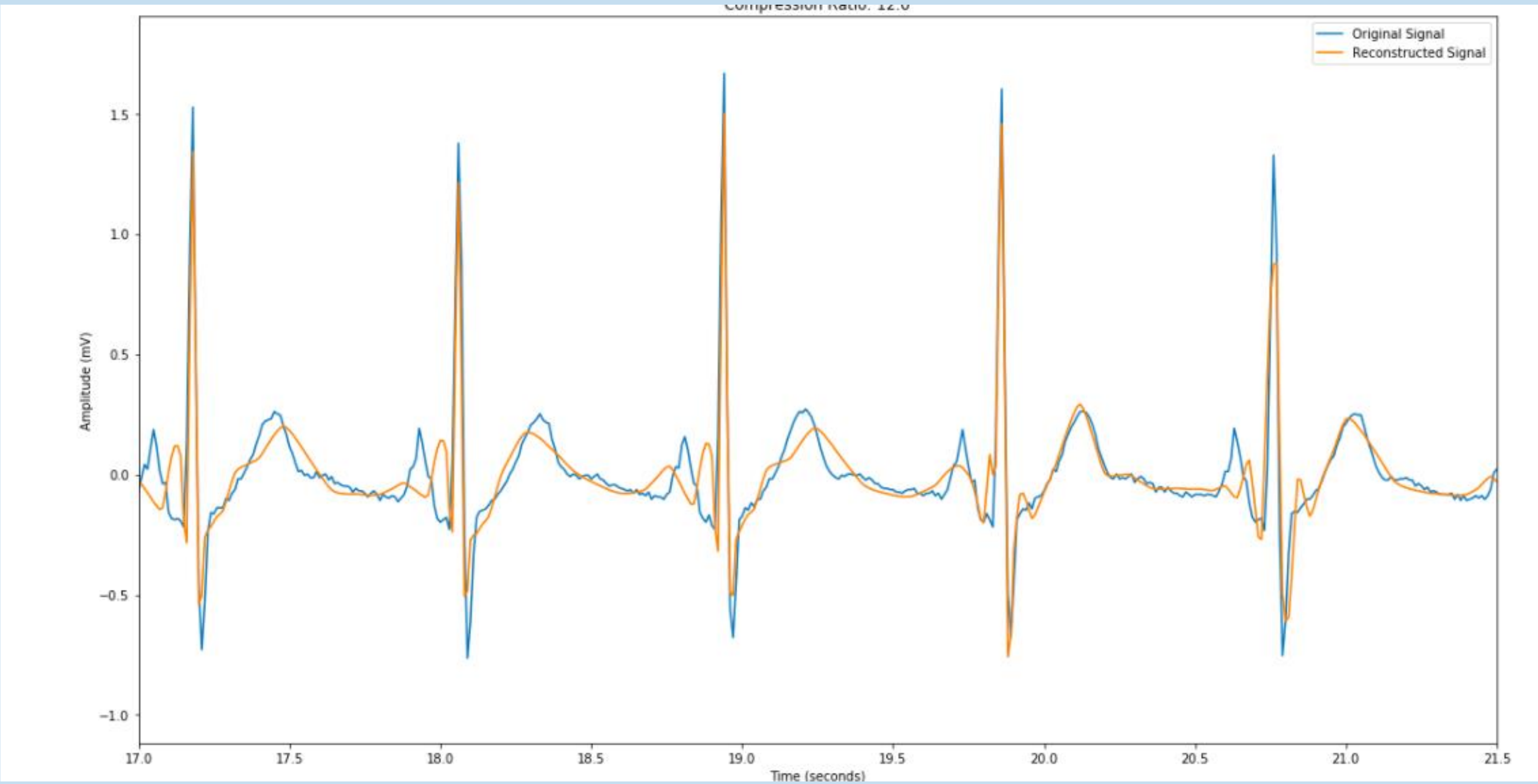
```
% QUANTIZATION
function [x_hat]=qtest(x,q_step)
x_c=round(x/q_step);
x_hat=x_c*q_step;
imshow(x_hat);
end
A_HL_LL=A_HL_1(1:300,1:480);  % Separate the region
  max(max(A_HL_LL)) % Find the largest element
```

OUTPUT AFTER QUANTIZATION:



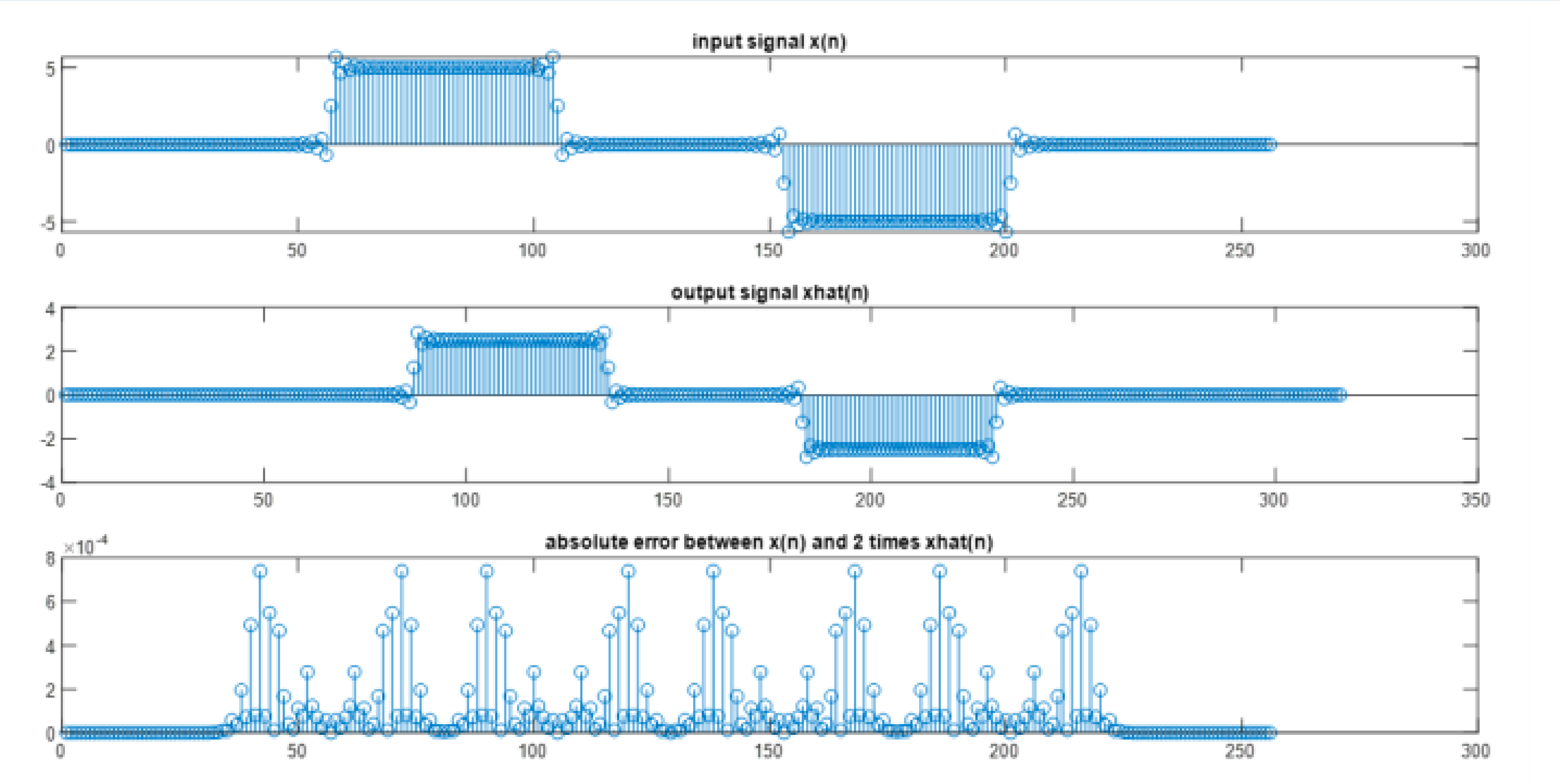
Plots of reconstructed signals vs number of bits used for quantization

OUTPUT



Original Signal vs Reconstructed Signal

ERROR BETWEEN ORIGNAL SIGNAL AND RECONSTRUCTED SIGNAL



CONCLUSION

- ECG data compression plays a key role in various systems viz., Holter monitor system, mobile ECG and telemedicine and e-health care system.
- Here, an algorithm based on the optimum QMF bank and RLE is presented for the ECG data compression.
- The first step of this work is to design optimum QMF filter bank, in which optimization is done using a linear function.
- Second step is to apply an ECG signal to optimum QMF bank to decompose it into different frequency bands.
- The coefficients truncation is done after applying a level thresholding.
- Further, RLE algorithm is used to improve the performance.
- Quantization is done to "filter out" some high frequency components in the process and make the signal less messy.
- We have obtained the error between original and reconstructed signal and noticed that the error is small. Hence, we are able to compress the data with no loss of useful information.