

به نام خدا

تعیین ضربان قلب با استفاده از تبدیل ویولت

نویسنده:

احسان میررحیمی

کلمات کلیدی:

Wavelet , ECG

چکیده:

در این مقاله با استفاده از دیتابیس موجود در Physionet.org ضربان قلب را از یک سیگنال ECG با استفاده از تبدیل ویولت محاسبه کرده ایم.



بخش اول :

در این قسمت با استفاده از تولباکس ویولت نرم افزار *Matlab* سیگنال های مورد نظر را آنالیز می کنیم . *Database* هائی که در اختیار داریم به صورت فایل هائی با سه فرمت مختلف می باشند (*.atr .hea .dat*). که هیچ کدام از این سه نوع فرمت برای تولباکس ویولت شناخته شده نمی باشد . (تنها فایل هائی با فرمت *.mat* در این تولباکس مورد استفاده قرار می گیرند). پس باید سیگنال اصلی را به صورت فایل قابل استفاده ی *Matlab* تبدیل کنیم. بعد از تبدیل سیگنال اصلی از فرمت *.dat* به *.mat* نتیجه به صورت یک فایل در فرمت *.mat* در می آید که آماده ی پردازش در *Matlab* است. مسئله ی مهمی که در آنالیز ویولت وجود دارد این است که برای پردازش سیگنال چه نوع ویولتی استفاده کنیم پاسخ این سوال بستگی به کاربرد مسئله و نوع سیگنال مورد پردازش می باشد مثلاً در یک سیگنال *ECG* یکی از بهترین گزینه ها برای انتخاب تابع ویولت استفاده از ویولت *Daubechies* می باشد که از نظر شکل کلی به سیگنال *ECG* نزدیک است . از بین ویولت های *Daubechies* نیز *db4* را به علت شباهت بیشتری که به سیگنال *ECG* دارد انتخاب می کنیم. شبیه بودن ویولت و سیگنالی که آن را پردازش می کنیم، باعث می شود که بتوانیم جزئیات دقیق تری از سیگنال را استخراج کنیم. و هنگام تقریب زدن و بازسازی سیگنال، سیگنال نزدیکتری بدست آوریم.

معرفی الگوریتم های استفاده شده

الگوریتم حذف نویز:

در سیگنال ECG دو نوع نویز مهم وجود دارد نویز *Baseline* که یک نویز محیطی بوده و در واقع مقداری DC به سیگنال بیمار اضافه می کند. برای حذف این نویز از الگوریتم *Moving Average filter* استفاده می کنیم. این الگوریتم در *Matlab* با دستور *Smooth* پیاده سازی می شود.

$$s2 = \text{smooth}(s1, \text{Span})$$

در دستور *smooth* ، *Span* پارامتر لازم برای هموار سازی سیگنال با استفاده از *Moving Average filter* می باشد و از تقسیم پریود سیگنال به طول بازه های نمونه برداری بدست می آید.

$$S2(1) = S1(1)$$

$$S2(2) = (S1(1) + S1(2) + S1(3))/3$$

$$S2(3) = (S1(1) + S1(2) + S1(3) + S1(4) + S1(5))/5$$

$$S2(4) = (S1(2) + S1(3) + S1(4) + S1(5) + S1(6))/5$$

.

.

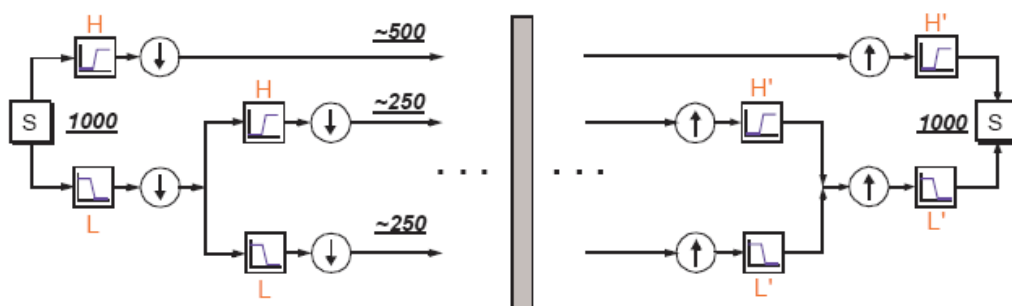
.

.

(1-1)

الگوریتم تجزیه و بازسازی سیگنال در ویولت:

تجزیه و بازسازی سیگنال از روی فیلترهای بالا گذر و پائین گذر و فرآیندهای *Upsampling*, *Downsampling* بدست می آید.



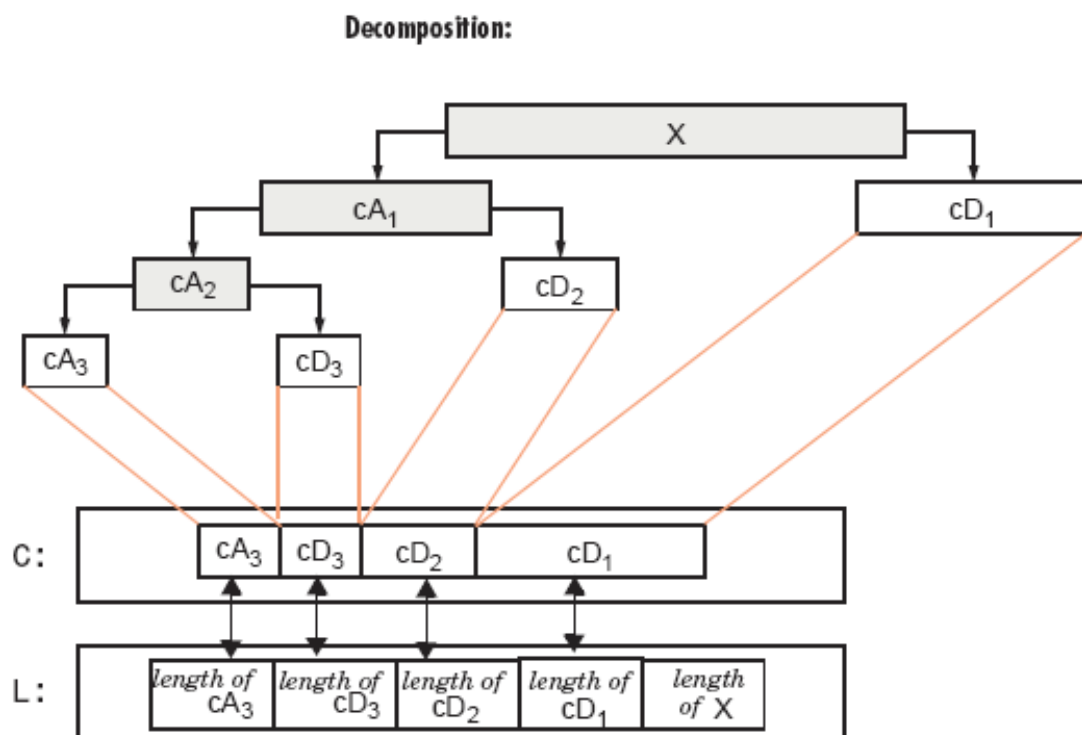
شکل 1-1 مراحل تجزیه و بازسازی سیگنال

همانطور که در قسمت های قبل توضیح داده شد سیگنال از دو فیلتر بالا گذر و پائین گذر عبور می کند و فرآیند *Downsampling* هم روی سیگنال انجام می شود. در *Matlab* مجموعه ی این فرآیندها با فرمت زیر انجام می شود:

$$[C, L] = \text{wavedec}(X, N, 'wname')$$

که X سیگنال مورد نظر، N تعداد سطوحی که سیگنال را تجزیه می کنیم و $wname$ نام ویولتی است که استفاده می کنیم. C برداری است که حاوی ضرایب *Approximation*، *detail* می باشد و L هم حاوی طول هر بخش می باشد.

مثلا برای $N=3$ نتیجه ی تجزیه ی سیگنال به صورت زیر می باشد.



شکل 2-1 تجزیه ی سیگنال و معرفی بردار های C, L

جزئیات دقیق بدست آمدن ضرائب Approximation و Detail به صورت زیر می باشد:

$$CA_1 = X * h \rightarrow (\downarrow 2)$$

$$CD_1 = X * g \rightarrow (\downarrow 2)$$

$$CA_2 = CA_1 * h \rightarrow (\downarrow 2)$$

$$CD_2 = CA_1 * g \rightarrow (\downarrow 2) \quad (1-2)$$

.

.

$$CA_j = CA_{j-1} * h \rightarrow (\downarrow 2)$$

$$CD_j = CA_{j-1} * g \rightarrow (\downarrow 2)$$

انتخاب Threshold:

انتخاب Threshold در کاربرد های مختلف بستگی به سیگنال مورد انتظار دارد و با روش های متنوعی انجام می شود. که در برنامه از الگوریتم میانگین گیری استفاده کرده ایم در این روش که بطور تجربی برای طیف وسیعی از سیگنال هاس ECG نتیجه ی مطلوب داده ، به این صورت عمل می کند که مقدار متوسط سیگنال را از میانگین سیگنال کم می کند و سیگنال های کوچکتر از این مقدار را حذف و بقیه را عبور می دهد(این کار در اصل برای حذف سایر قسمت های سیگنال ECG مثل بخش Q یا S صورت می گیرد که در اثر مربع شدن در بخش الگوریتم R _ detection امکان دارد با سیگنال R اشتباه شوند).

$$Threshold = \frac{[(\max \text{ signal value}) - (\text{mean value})]}{2}$$

در این قسمت برای استخراج ضربان قلب از سیگنال ECG نیاز به سیگنال R داریم برای دست یابی به این سیگنال به این ترتیب عمل می کنیم:

ابتدا سیگنال را در ۸ سطح تجزیه می کنیم با توجه به اینکه جزئیات مورد نظر (بخش R سیگنال ECG) در ضرائب (d3-d5) موجود می باشد با استفاده از این ضرائب سیگنال را بازسازی می کنیم. بعد از باز سازی سیگنال باید سیگنال حاصل را مربع کنیم تا مقادیر منفی آن حذف شوند. با عمل *Threshold* بخش های اضافه را حذف می کنیم (تا بجای استخراج بخش R بخش Q یا S بدست نیایند) و نهایتا سیگنال R را بدست می آوریم. بعد از بدست آوردن سیگنال R نیاز به استخراج قله های سیگنال R و زمان متناظر آنها داریم تنها نکته ی مهم در این بخش این است که با توجه به اینکه در تجزیه ی سیگنال فرآیند های *Upsampling*, *Downsampling* روی تعداد نمونه های سیگنال تاثیر می گذارند برای بدست آوردن زمان متناظر با هر قله نیاز به فرکانس نمونه برداری برای سیگنال R داریم. این فرکانس با توجه به این خاصیت بدست می آید که زمان حضور سیگنال اصلی (سیگنال کامل ECG) و سیگنال R باهم برابرند و بازه ی زمانی سیگنال مورد استفاده ی ما $20s$ می باشند. با توجه به این زمان و تعداد نمونه های سیگنال R فرکانس نمونه برداری و در نتیجه زمان هر قله ی سیگنال R را بدست می آوریم. برای تعیین ضربان قلب احتیاج داریم بدانیم در فاصله ی هر دو قله ی سیگنال R چند مربع نوار کاغذی ECG جای می گیرد با توجه به اینکه طول هر مربع $0.2s$ است تعداد مربعات برابر با فاصله ی زمانی دو قله تقسیم بر این عدد ($0.2s$) می باشد بعد از بدست آوردن تعداد مربعات برای بدست آوردن ضربان قلب باید این عدد را بر 300 تقسیم کنیم عدد حاصل ضربان قلب بیمار می باشد.

```

%Beat Rate extraction Using Discrete Wavelet Transform
%program databases are available in:
%http://www.physionet.org
clear;close all;clc;
load 100_ECG_0_20

%%%Eliminate Baseline Drift
s1=ECG_1;s2=smooth(s1,150);ecgsmooth=s1-s2;

%%%apply Wavelet Transform
[C,L]=wavedec(ecgsmooth,8,'db4');
[d1,d2,d3,d4,d5,d6,d7,d8]=detcoef(C,L,[1,2,3,4,5,6,7,8]);

%%%Denoise
[thr,sorh,keepapp]=ddencmp('den','wv',ecgsmooth);
cleanecg=wdencmp('gbl',C,L,'db4',8,thr,sorh,keepapp);

%%%thresholding1
max_value=max(cleanecg);
mean_value=mean(cleanecg);
threshold=(max_value-mean_value)/2;

%%%R detection algorithm
a5=appcoef(C,L,'db4',5);
C1=[a5;d5;d4;d3];
L1=[length(a5);length(d5);length(d4);length(d3);length(cleanecg)];
R_detect_signal=waverec(C1,L1,'db4');
R_detect_squared=R_detect_signal.^2;

%%%Beat_Rate_Extraction_Algorithm
for a=1:length(R_detect_squared)
    if R_detect_squared(a)>threshold
        R_detect_new(a)=R_detect_squared(a);
    else
        R_detect_new(a)=0;
    end
end
mean_R_detect=5*mean(R_detect_new);
for q=1:length(R_detect_new)-1
    if R_detect_new(q)< mean_R_detect
        R_detect_new(q)=0;
    end
end

end
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
d=0;
for b=1:length(R_detect_new)-1
    if (R_detect_new(b)==0) & (R_detect_new(b+1)~=0)
        d=d+1;
        indext(d)= b+1;
    end
end
fs_R_deetect=length(R_detect_new)/20;
time=indext.*1/fs_R_deetect;
ind=0;
for z=1:length(time)-1
    ind=ind+1;
    time_diff(ind)=time(z+1)-time(z);
end

```



```

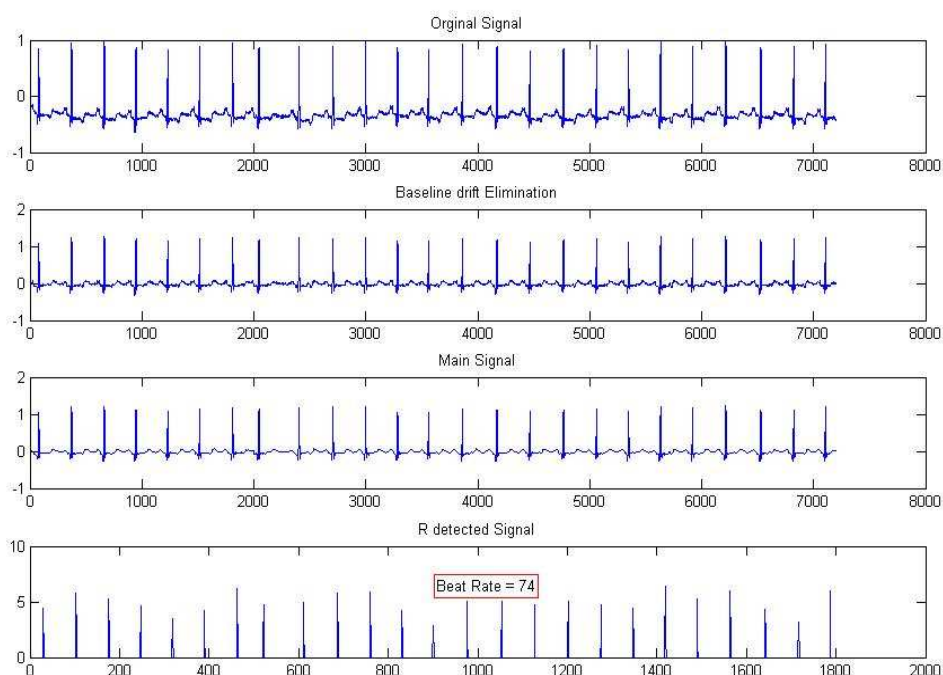
end
av_time=mean(time_diff);
Square_Number=av_time/.2;
beat_Rate=300/Square_Number;
high=max(R_detect_new);

%%%Plot the Original Signal and Eliminating Baseline Drift signal
subplot(411);plot(s1);title('Original Signal');
subplot(412);plot(s1-s2);title('Baseline drift Elimination');
subplot(413);plot(cleanecg);title('Main Signal');
subplot(414);plot(R_detect_new);title('R detected Signal');
text(length(R_detect_new)/2,high,['Beat Rate = ',num2str(fix(beat_Rate))],'EdgeColor','red');

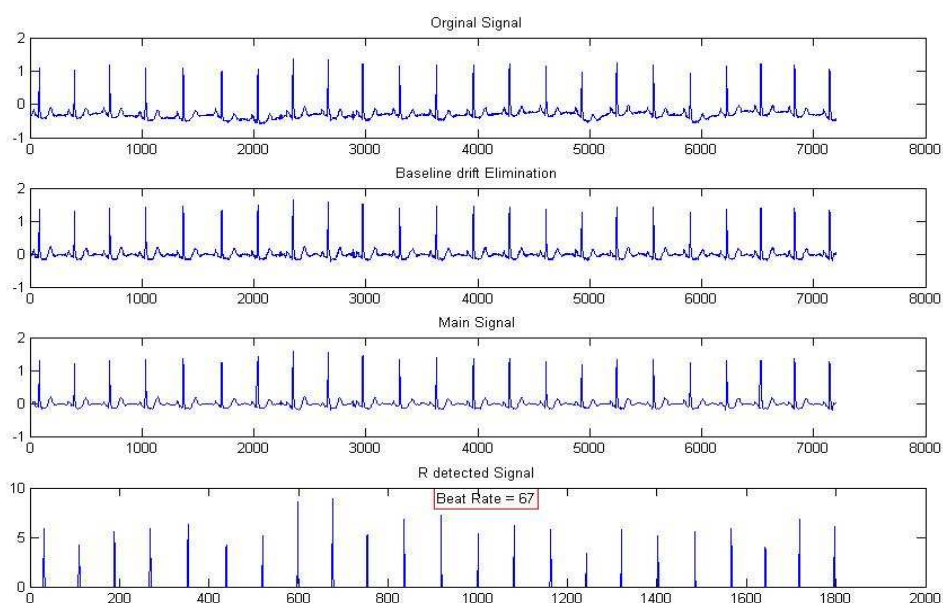
```

بخش دوم:

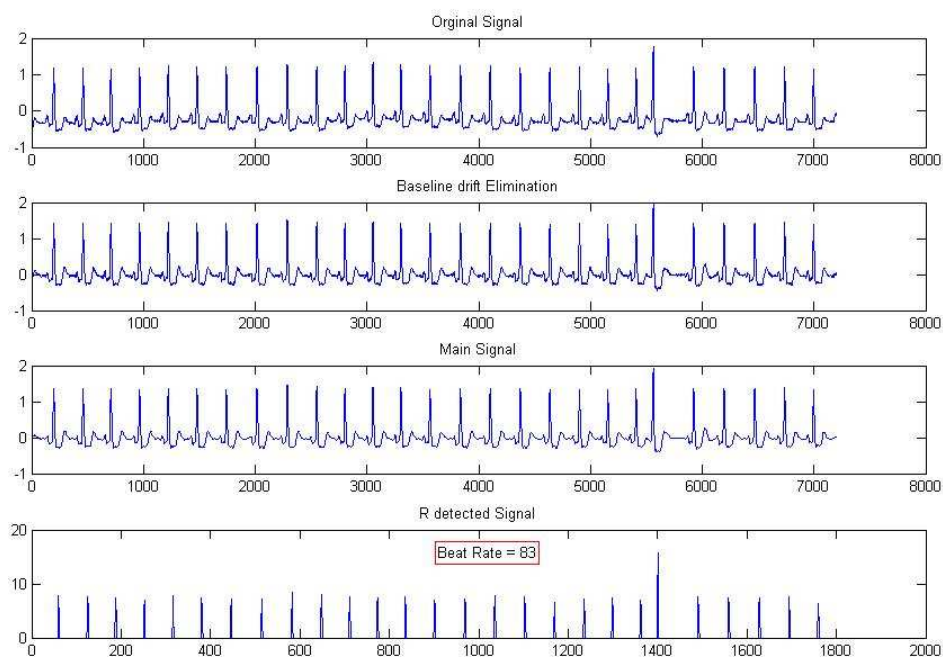
نتایج شبیه سازی برنامه)



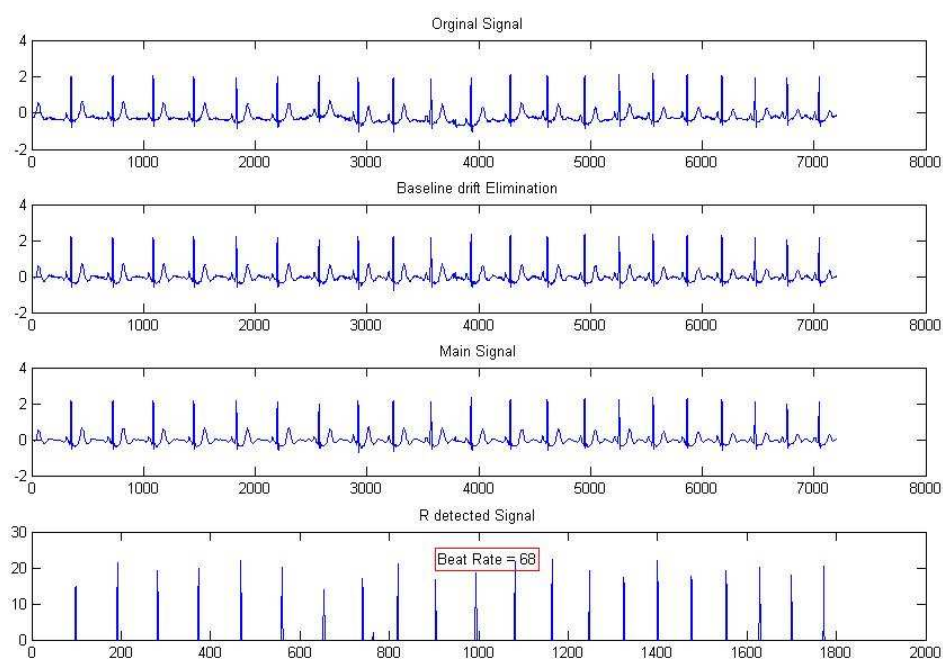
شکل ۱-۲) سیگنال ECG به همراه سیگنال *Denoise* شده و سیگنال *R detect*



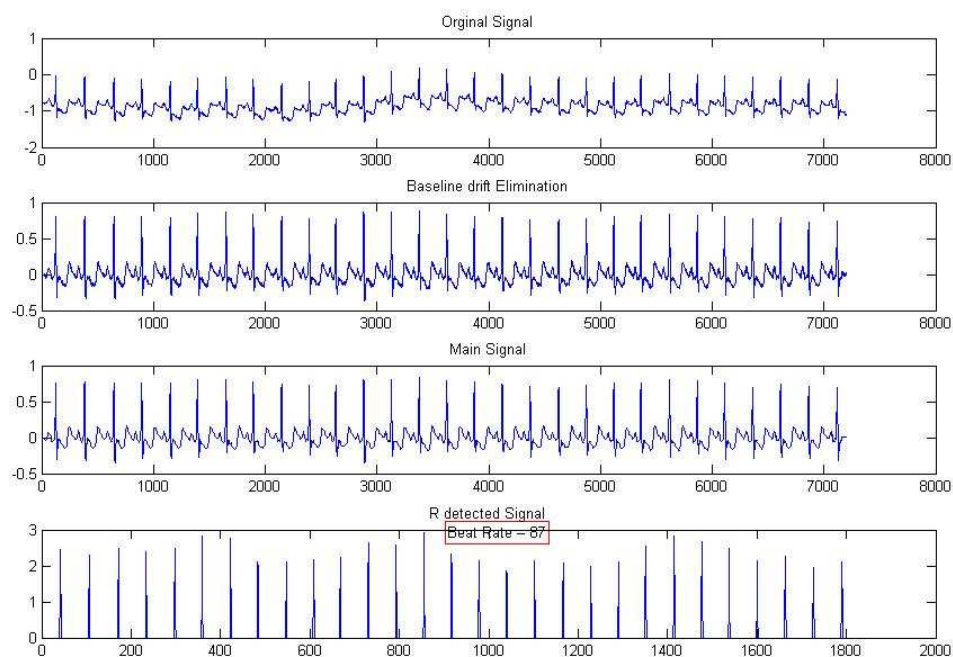
شکل ۲-۲) سیگنال ECG به همراه سیگنال *Denoise* شده و سیگنال *R detect*



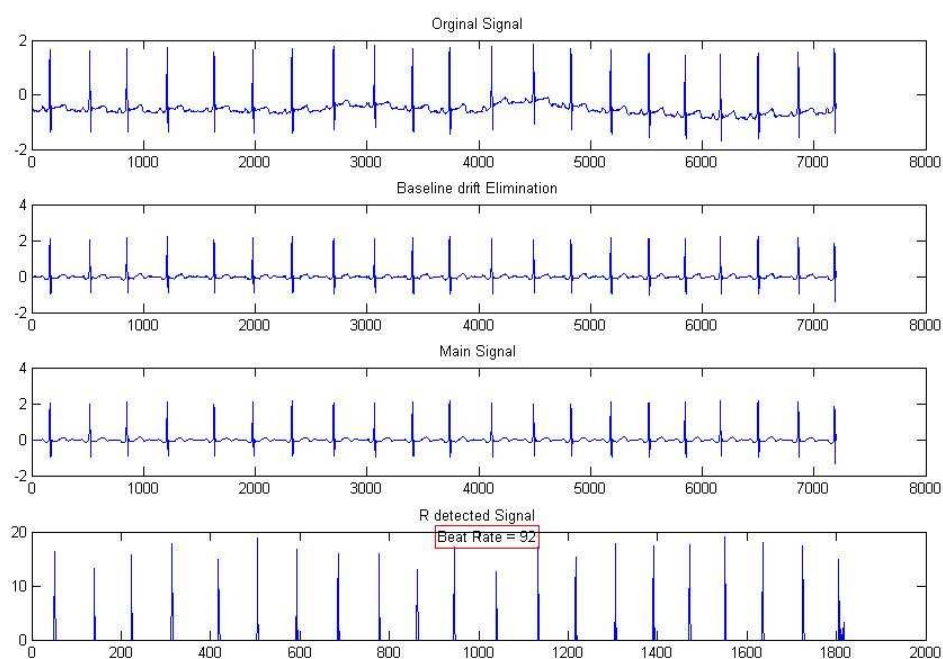
شکل ۳-۲) سیگنال ECG به همراه سیگنال Denoise شده و سیگنال R detect



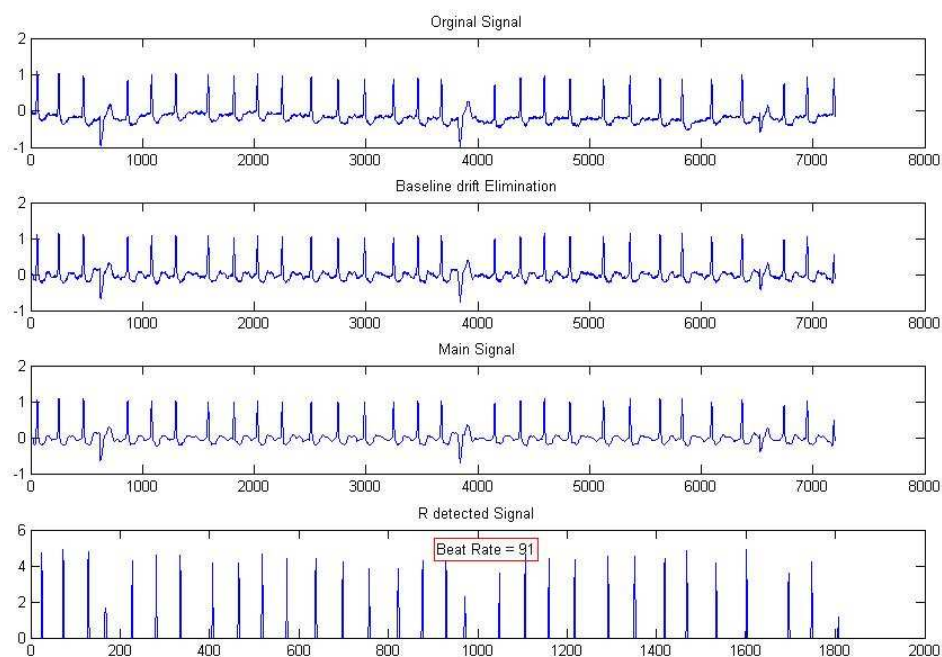
شکل ۴-۲) سیگنال ECG به همراه سیگنال Denoise شده و سیگنال R detect



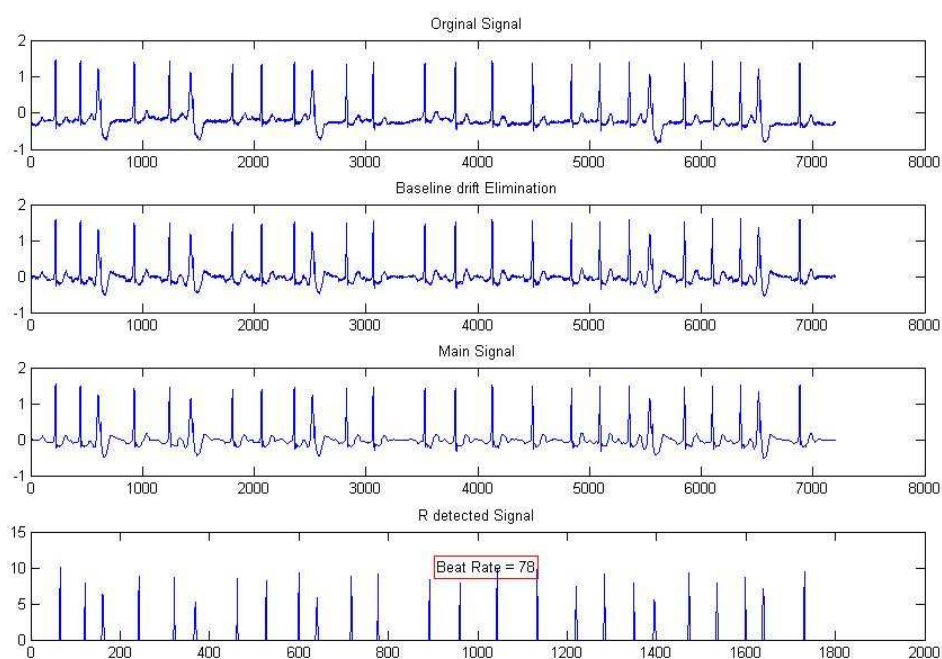
شکل ۵-۲ سیگنال ECG به همراه سیگنال *Denoise* شده و سیگنال *R detect*



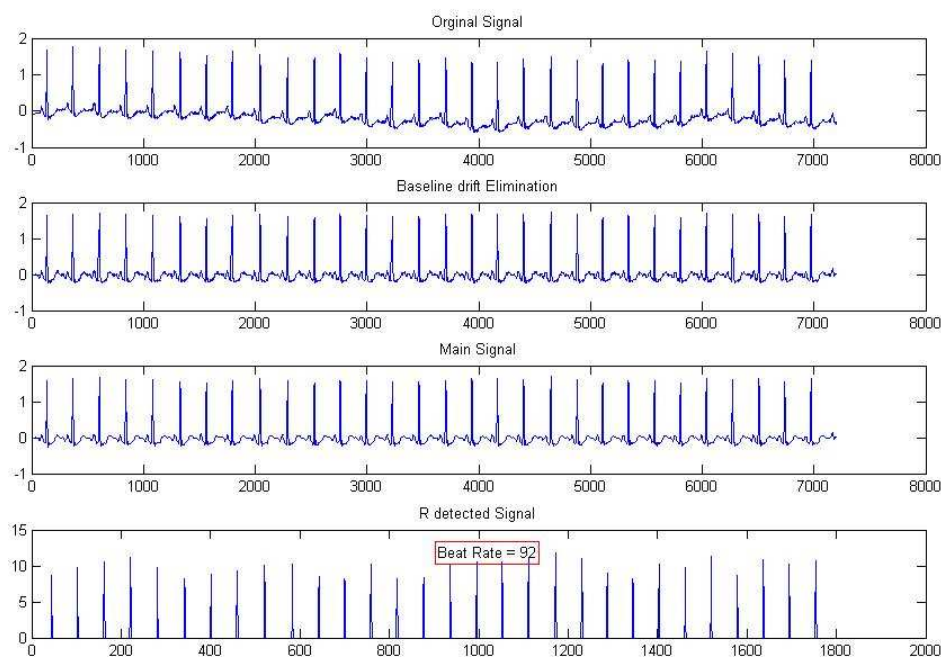
شکل ۵-۶ سیگنال ECG به همراه سیگنال *Denoise* شده و سیگنال *R detect*



شکل ۷-۲ سیگنال ECG به همراه سیگنال Denoise شده و سیگنال R detect



شکل ۸-۲ سیگنال ECG به همراه سیگنال Denoise شده و سیگنال R detect



شکل ۹-۲ سیگنال ECG به همراه سیگنال *Denoise* شده و سیگنال *R detect*

نتیجه گیری:

همانطور که از نمودارهای این بخش مشخص است با استفاده از تبدیل گسسته ی ویولت توانستیم بخش R سیگنال ECG را استخراج و از روی آن ضربان قلب شخص را بدست آوریم. مزیت این روش در این است که برای تعیین ضربان قلب دیگر نیازی به کاغذ نواری ECG نداریم یعنی در مواردی که فقط شکل سیگنال را بدون تقسیم بندی زمانی خاصی که در نوار کاغذی ECG صورت می گیرد داشته باشیم به راحتی با این برنامه می توانیم ضربان قلب را از روی سیگنال استخراج کنیم.

به عنوان افق های روشن تر برای این پروژه با اضافه کردن قسمت های دیگری به این پروژه مثل سیستم های مقایسه کننده و با استفاده از روش های بازشناسی الگو می توانیم آریتمی های بسیاری را از روی بخش های مختلف ECG تشخیص دهیم. که این هدف نیز به بازه ی زمانی و آشنائی با شبکه های عصبی نیاز دارد.