

Noise Corrupted Signals and Signal Processing using MATLAB

Cleaning a Noise Corrupted ECG Signal

Jennifer Estrada

4/23/2009

While taking ECG signal data, a noise signal was introduced. In order for the data to be useful for detecting any abnormalities in the subject's heart, the signal must be refined to eliminate the noise. In order to fix this problem, identifying and designing the appropriate filter to reconstruct the original ECG signal. In most cases, noise corruption of a signal when sampling data is common and unpreventable. For more accurate readings, filtering of sampled signals for common noise corruption such as the power line hum would be necessary.

Statement of Problem and Assumptions

While data was taken, a noise was introduced and corrupted the original signal. In order to get rid of the noise, a proper filter must be designed. Electrocardiogram (ECG) signal is some index of the functionality of the heart. For example, a physician can detect arrhythmia by studying abnormalities in the ECG signal. Since very fine features present in an ECG signal may convey important information, it is important to have the signal as clean as possible. Figure 1 shows a clean ECG signal, labeled with some important features doctors look to identify abnormalities. Cables carrying ECG signals from the examination room to the monitoring equipment are susceptible to electromagnetic interference of the power frequency by ubiquitous supply lines and plugs, sometimes enough to totally mask the signal by this type of noise/interference.

Results and Recommendations

The Electrocardiograph signal taken from a patient was corrupted by an external noise introduced while the data was taken.

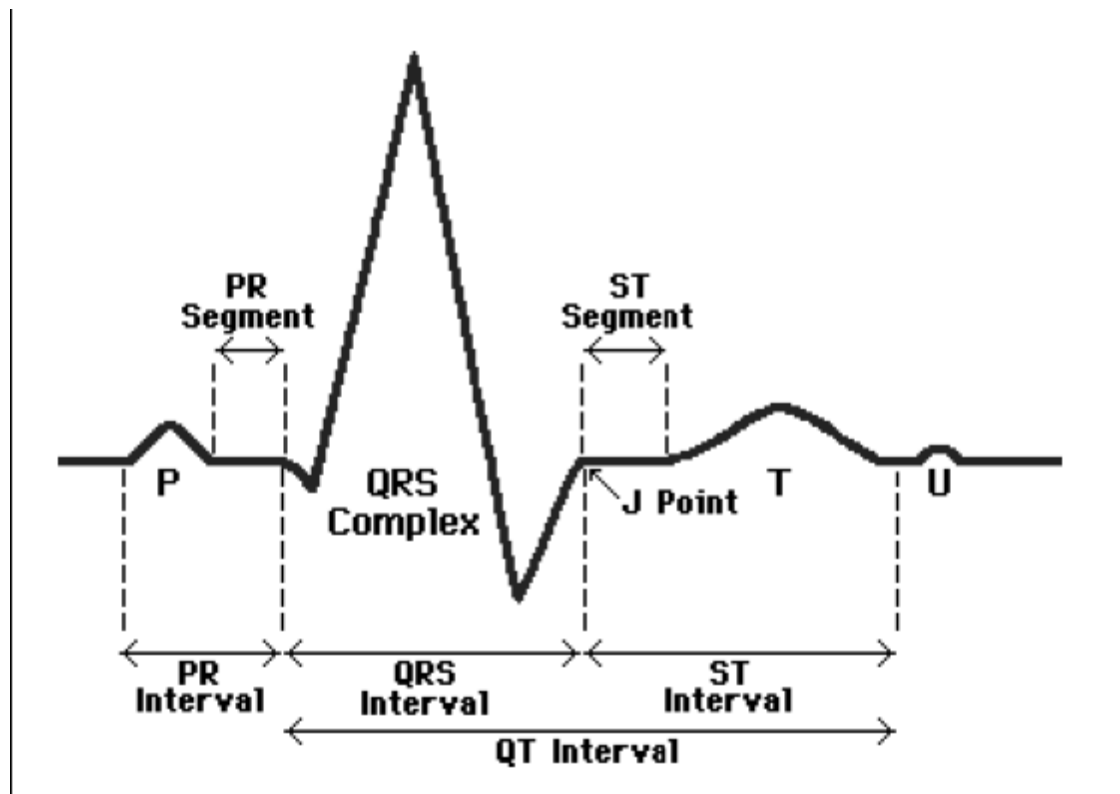


Fig.1. *Diagram of Actual ECG signal with identification*

Figure 2 depicts the corrupted signal that must be filtered.

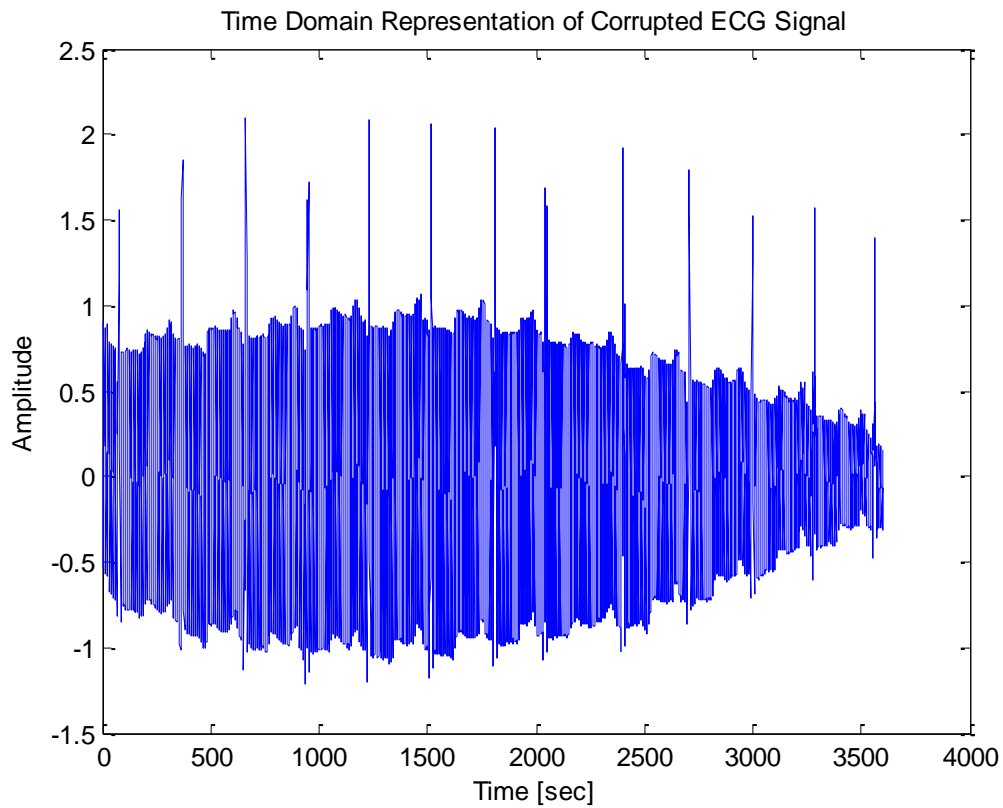


Fig.2.*Corrupted ECG Signal*

Transforming the corrupted signal into the frequency domain, the frequencies that belong to the external noise can be easily detected. As seen in figure 3, the noise is present at 60 Hertz. This noise is known as a power line hum, caused by any electrical appliances and lights.

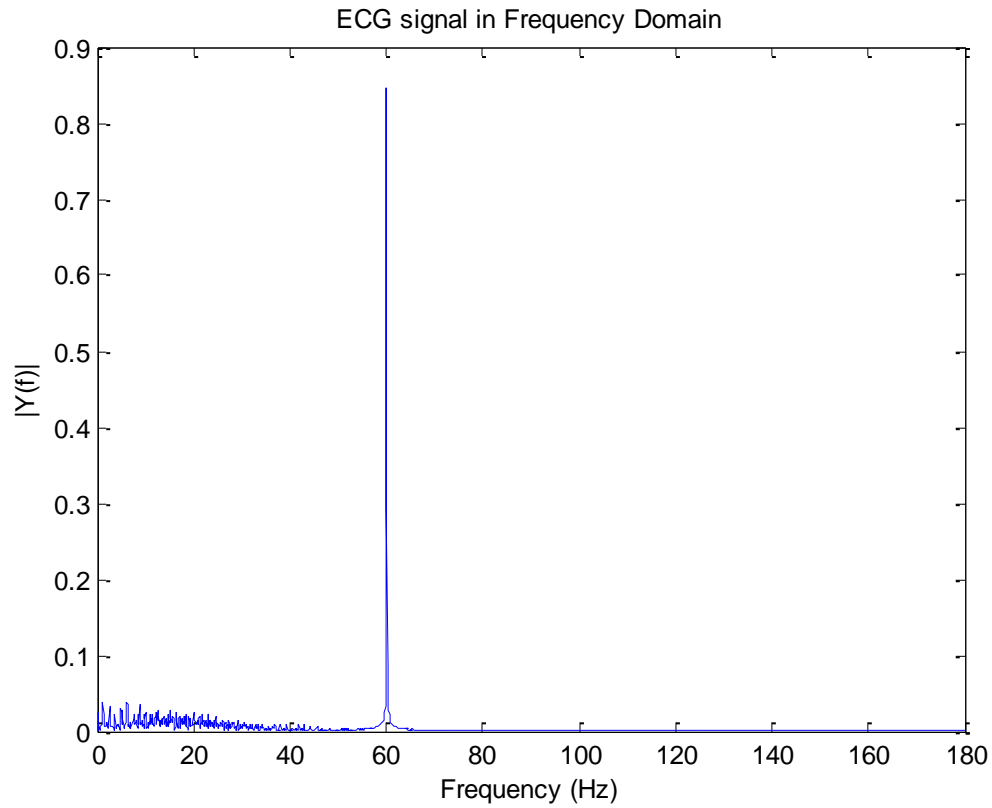


Fig.3. *Corrupted ECG Signal in the Frequency Domain*

Now that the proper frequency of the noise has been identified in the frequency domain, the proper filter can be created to clean the corrupted signal. Since the signal has only one frequency that must be eliminated, a Notch Filter would be ideal for this application with $F_0 = 60$ Hertz.

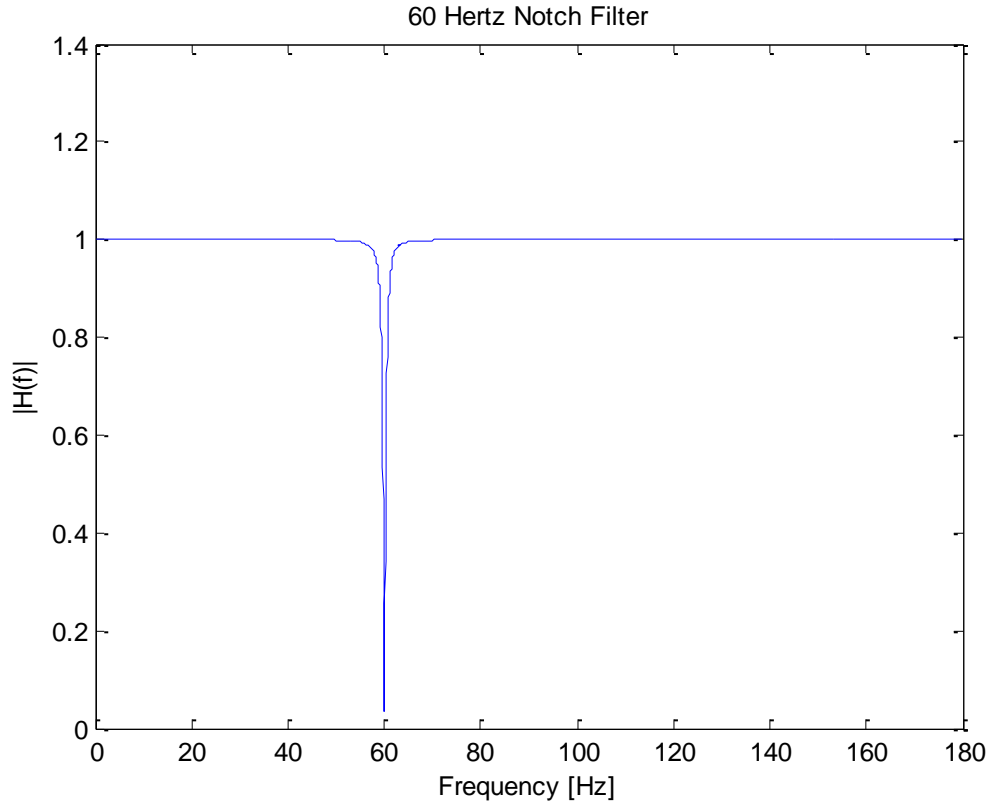


Fig.4. *Notch Filter Magnitude Response*

Applying the Notch filter to the noise corrupted signal and returning to the time domain, the recovered signal looks significantly more like an accurate reading of an ECG signal. Placing a zero at the point on unit circle that corresponds to the notch frequency F_0 can ensure the filter completely attenuates signals at the notch frequency. Placing a pole at an angle of $\theta_0 = 2\pi F_0 T$ with $r < 1$ can control the 3-dB bandwidth of the stopband, both poles and zeros appear in complex conjugate pairs to ensure real coefficients. The transfer function of a Notch Filter is:

$$H_{notch}(z) = \frac{b_0(z - e^{j\theta_0})(z - e^{-j\theta_0})}{(z - re^{j\theta_0})(z - re^{-j\theta_0})}, \quad r \approx 1 - \frac{\Delta F \cdot \pi}{f_s} \quad b_0 = \frac{|1 - 2r \cos(\theta_0) + r^2|}{2|1 - \cos(\theta_0)|} \quad [1]$$

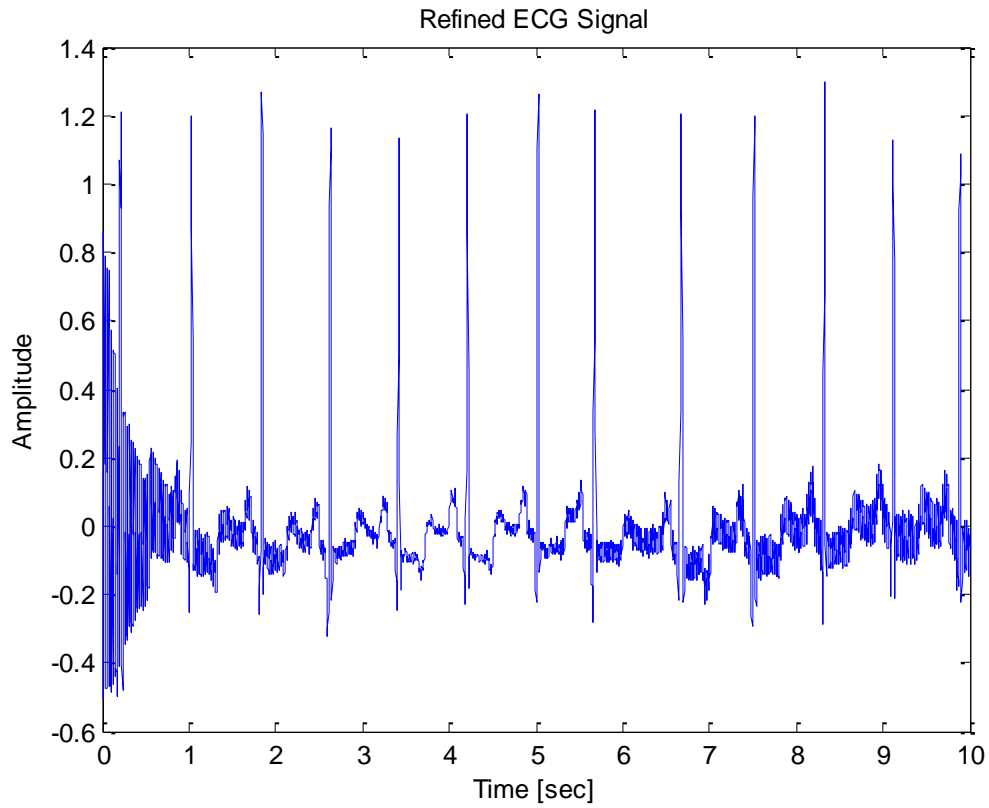


Fig.5. *Refined ECG Signal with $\Delta F = 1$*

From equation 1, the only variable that can be fine tuned is ΔF . Thus, manipulation of the parameter ΔF shows different results, getting cleaner with a higher value, until the effect levels off since must be $r \ll 1$.

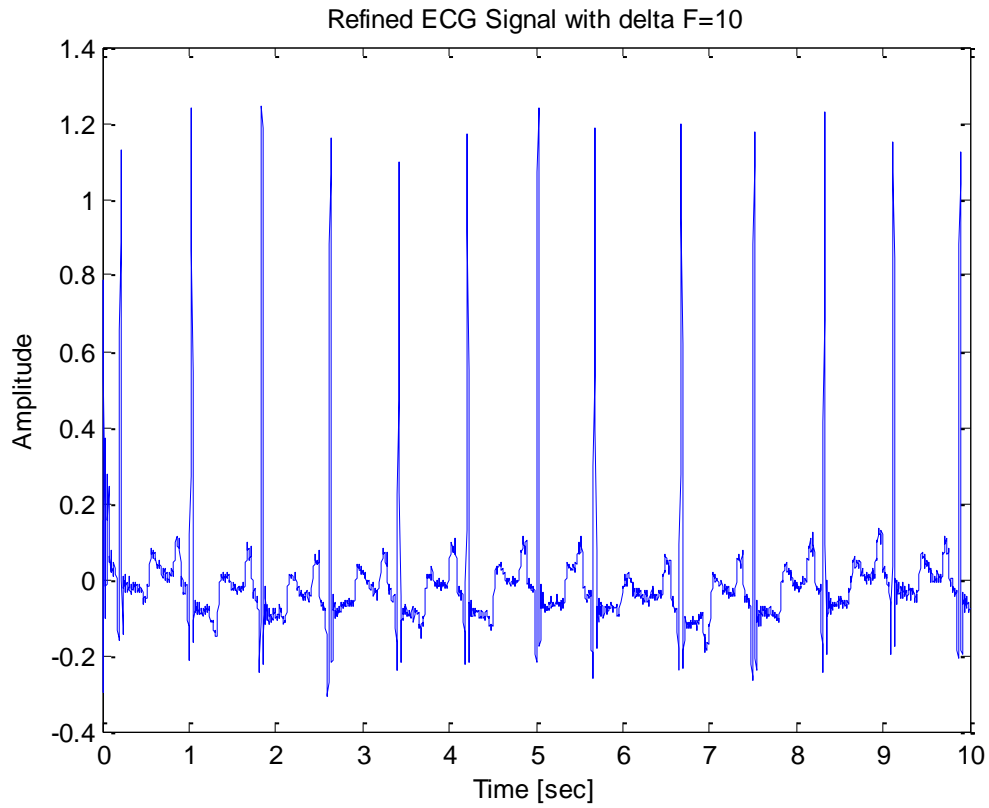


Fig.6. *Refined ECG Signal with $\Delta F = 10$*

With ΔF becoming greater, the signal seems to have less noise corruption. This continues until a threshold, where the value of ΔF ceases to make an impact on the refinement of the ECG signal.

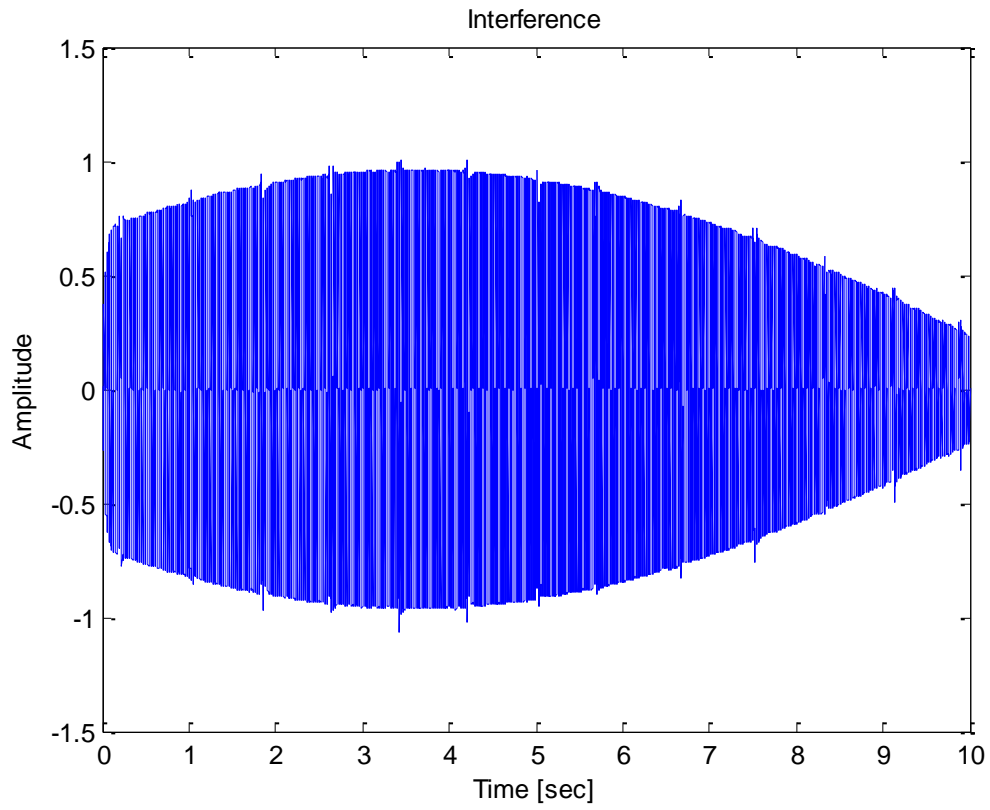


Fig.7.*Interference*

Subtracting the refined signal from the corrupted signal, the interference is plotted in figure 7.

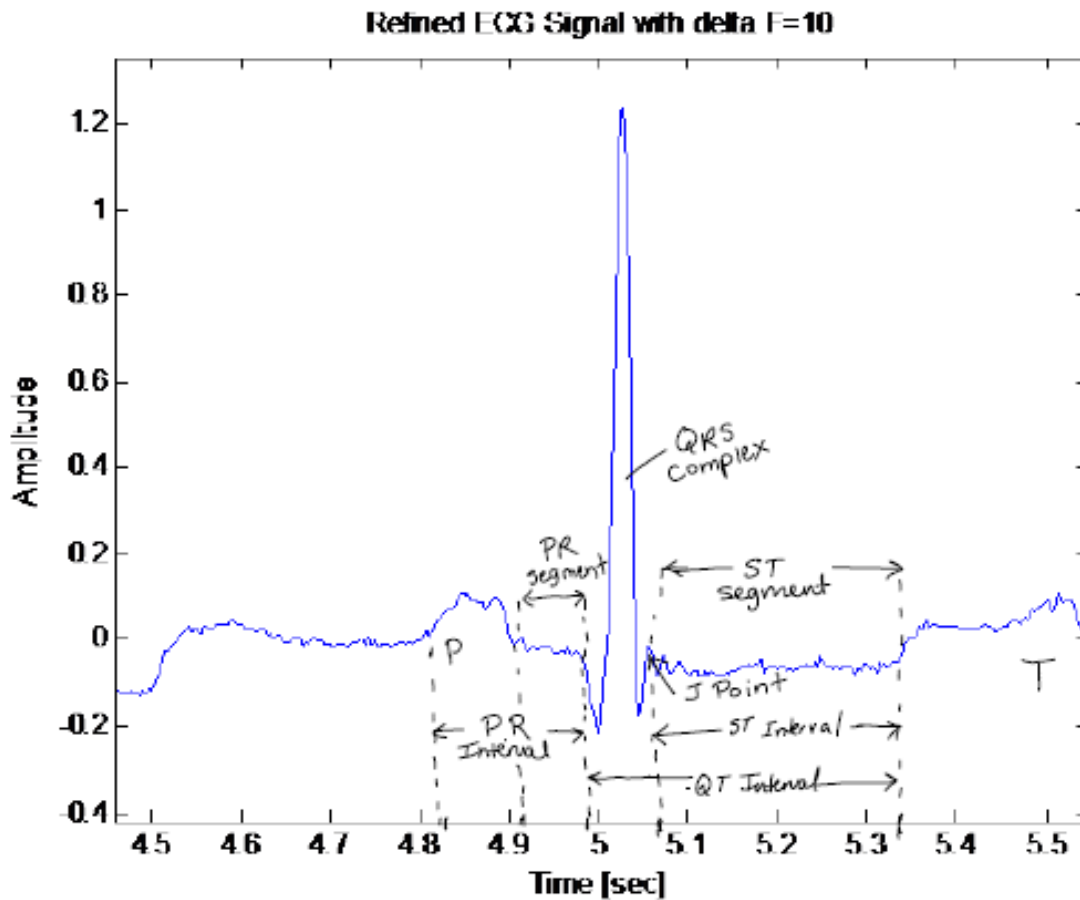


Fig.8. *Identifying the Signal portions from the refined signal*

Figure 8 is a close up of one period of the signal. The details of the ECG signal are easily discernable after the filter has been applied to a physician. PRSTQR Intervals are identified and clearly labeled in the figure.

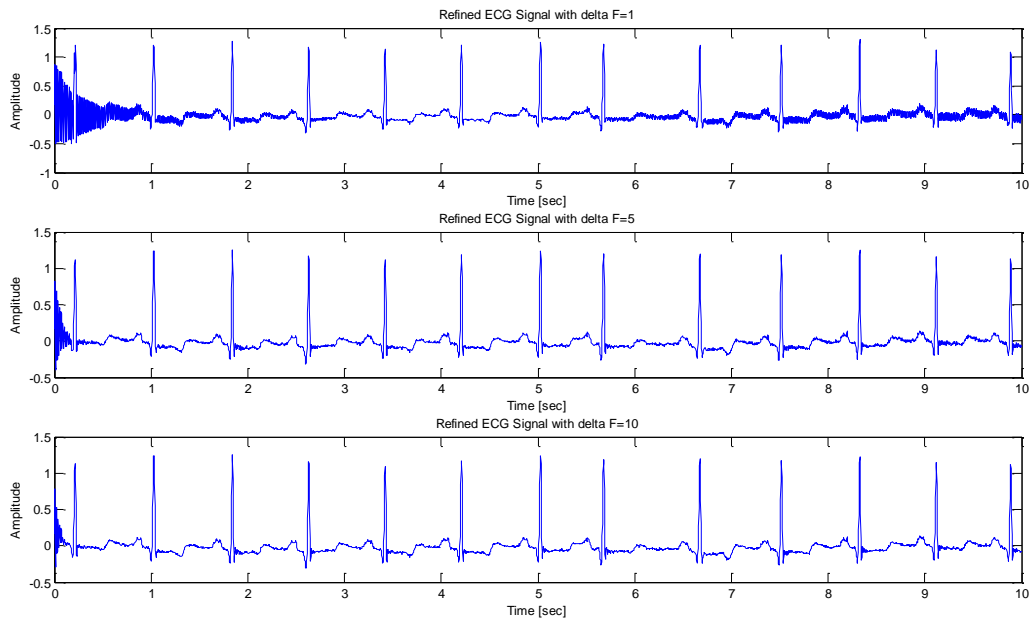


Fig.9. *Refined ECG Signal with $\Delta F = 1, 5$ and 10*

Increasing ΔF has a better effect for cleaning up the ECG signal, as figure 8 clearly illustrates.

Electromagnetic Interference is common in electrical appliances and electronics. Within the medical field, the accuracy of the information from data collected from patients can be altered enough by this interference to cause misdiagnosis.

References

- [1] R. Schilling and S. Harris. **Fundamentals of Digital Signal Processing using MATLAB**. Canada: Thompson Publishing Co. 2005.
- [2] D. Klotz. "Guide to Report Writing and Grading." **GE Advanced Courses** (1992) : pp. 3-9.

Appendix:

MATLAB Code-Main Code

```
%Data Importation and Extraction
input=load('input.mat'); %10 second signal
fs=360; %Sampling Frequency (Hertz)
N=length(input);

t = ((0:length(input)-1)./fs); %Time Interval
t=t';
figure(1)
plot(t, input)
title('Noise Corrupted ECG Signal')
xlabel('Time [sec]')
ylabel('Amplitude')
%-----

Ys = fft(input)/N;
equal_space=linspace(0,.5,N/2); %adjusting the interval of frequency range
freq = fs*equal_space;
Ys = Ys(1:ceil(N)/2);
plot(freq,2*abs(Ys))
title('ECG signal in Frequency Domain')
xlabel('Frequency (Hz)')
ylabel('|Y(f)|')

F_0=60;
Delta_F=1;
[b,a] = f_iirnotch (F_0,Delta_F,fs);%creates coefficient vectors a and b
refined=filter(b,a,input); %filter signal
[H Hf] = f_freqz(b,a,N); %returns transfer function
amp=abs(H);
figure(2)
plot(Hf*fs,amp)
title('60 Hertz Notch Filter')
xlabel('Frequency [Hz]')
ylabel('|H(f)|')

%-----
%Attempting different values of delta F
subplot(3,1,1)
plot(t,refined)
title('Refined ECG Signal with delta F=1')
xlabel('Time [sec]')
ylabel('Amplitude')

F_0=60;
Delta_F=5;
[b,a] = f_iirnotch (F_0,Delta_F,fs);%creates coefficient vectors a and b
refined=filter(b,a,input); %filter signal

subplot(3,1,2)
```

```

plot(t,refined)
title('Refined ECG Signal with delta F=5')
xlabel('Time [sec]')
ylabel('Amplitude')

F_0=60;
Delta_F=10;
[b,a] = f_iirnotch (F_0,Delta_F,fs);%creates coefficient vectors a and b
refined=filter(b,a,input); %filter signal

subplot(3,1,3)
plot(t,refined)
title('Refined ECG Signal with delta F=10')
xlabel('Time [sec]')
ylabel('Amplitude')

%-----
%interference

interference=input-refined;
plot(t, interference)
title('Interference')
xlabel('Time [sec]')
ylabel('Amplitude')

```

Notch Filter Code (Made by Prof. Schilling)

```
function [b,a] = f_iirnotch (F_0,Delta_F,fs)

%F_IIRNOTCH: Design an IIR notch filter
%
% Usage: [b,a] = f_iirnotch (F_0,Delta_F,fs)
%
% Inputs:
%     F_0      = notch frequency
%     Delta_F  = radius of 3dB bandwidth (Delta_F << fs)
%     fs       = sampling frequency
% Outputs:
%     b = 1 by 3 numerator coefficient vector
%     a = 1 by 3 denominator coefficient vector
%
% See also: F_IIRES, F_IIRCOMB, F_IIRINV

% Initialize

fs = f_clip (fs,0,fs);
F_0 = f_clip (F_0,0,fs/2);
Delta_F = f_clip (Delta_F,0,fs/4);

% Find the parameters

r = 1 - (Delta_F*pi/fs);
theta_0 = 2*pi*F_0/fs;
b_0 = abs(1 - 2*r*cos(theta_0) + r^2)/(2*abs(1-cos(theta_0)));

% Find the coefficient vectors

b = b_0*[1,-2*cos(theta_0),1];
a = [1,-2*r*cos(theta_0),r^2];
```