4. ECG – Removing 60 Hz Interference

The recording of an electrocardiogram (ECG) is susceptible to 60-Hz power-frequency interference because of the use of exposed electrodes placed on the chest. The 60-Hz interference can be removed with the help of a notch filter.

The transfer function of a 2nd order notch filter with notch frequency f_0 and 3-dB width Δf (both in Hz) is given by,

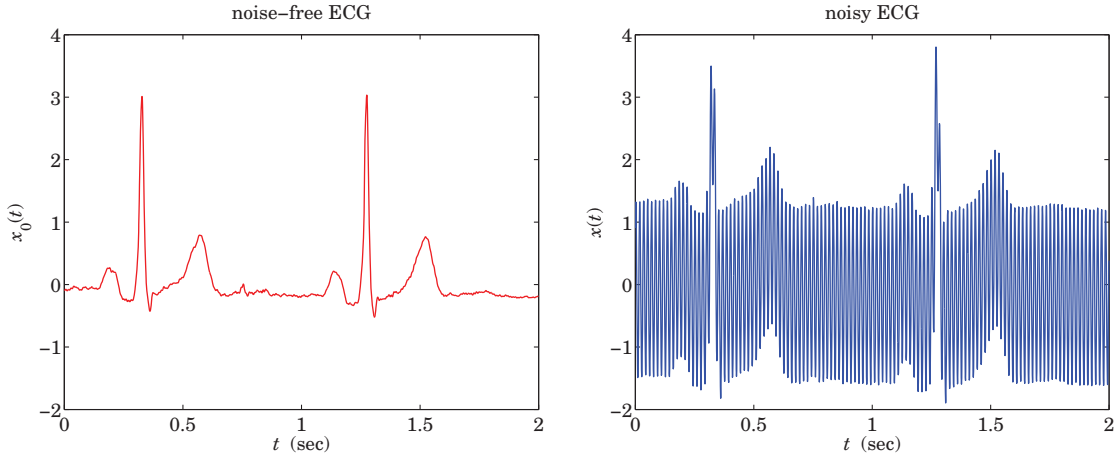
$$\boxed{H(s) = \frac{s^2 + \omega_0^2}{s^2 + as + \omega_0^2}} \tag{4}$$

where $\omega_0 = 2\pi f_0$ is the notch frequency in rads/sec, and, $a = 2\pi\Delta f$. The corresponding impulse response is given by,

$$h(t) = \delta(t) - ae^{-at/2}\left[\cos(\omega_r t) - \frac{a}{2\omega_r}\sin(\omega_r t)\right]u(t)\,, \quad \text{where} \quad \omega_r = \sqrt{\omega_0^2 - \frac{a^2}{4}} \tag{5}$$

Assuming that $\omega_0 > a/2$, as is typical for narrow notch filters, we see from Eq. (5) that the impulse response is a decaying sinusoid with a time-constant of $\tau = 2/a = 1/\pi\Delta f$, which is inversely proportional to the 3-dB width. Thus, the more narrow the filter, the longer its transient response and time constant. This tradeoff is observed in the graphs of the filtered ECGs below for the cases $\Delta f = 1$ and $\Delta f = 3$ Hz.

The included files, **ece0.dat**, **ecg60.dat**, represent a two-beat recording of a noise-free ECG and an ECG contaminated with 60 Hz interference, respectively, and are shown below.



The following MATLAB code does the following: (a) loads the ECG data, (b) defines (as transfer function classes) two notch filters, with 3-dB widths of $\Delta f = 1$ and $\Delta f = 3$ Hz, (c) computes and plots the filtered ECG outputs using the built-in function **lsim**, and (d) computes and plots, versus frequency f in Hz the corresponding magnitude-square filter responses obtained by setting $s = j\omega$ in Eq. (4), that is, the quantity,

$$|H(\omega)|^2 = \left.\frac{(\omega^2 - \omega_0^2)^2}{(\omega^2 - \omega_0^2)^2 + a^2\omega^2}\right|_{\omega=2\pi f} \tag{6}$$

The left and right 3-dB frequencies f_1, f_2 are given in Hz by,

$$\begin{aligned} f_1 &= -\frac{\Delta f}{2} + \sqrt{f_0^2 + \left(\frac{\Delta f}{2}\right)^2} \\ f_2 &= +\frac{\Delta f}{2} + \sqrt{f_0^2 + \left(\frac{\Delta f}{2}\right)^2} \end{aligned} \quad \Rightarrow \quad \Delta f = f_2 - f_1 \tag{7}$$

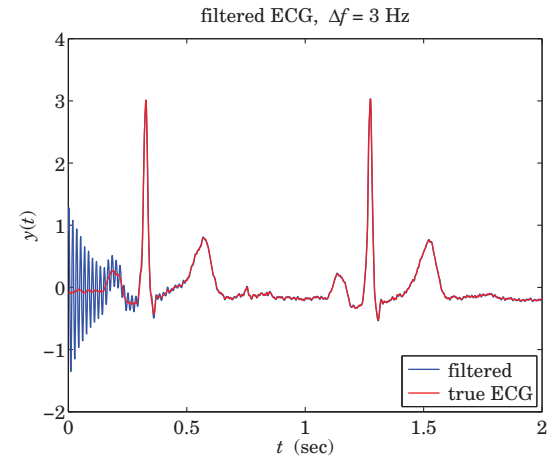
They are not quite symmetrically placed about f_0, only approximately so. They are obtained by finding the positive solutions (in $\omega = 2\pi f$) of the 3-dB condition (i.e., at half power),

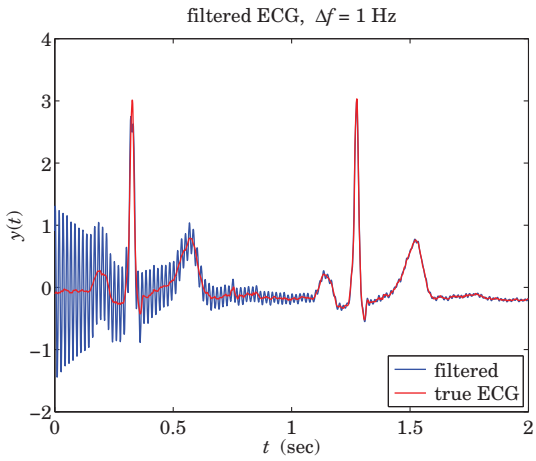
$$|H(\omega)|^2 = \frac{(\omega^2 - \omega_0^2)^2}{(\omega^2 - \omega_0^2)^2 + a^2\omega^2} = \frac{1}{2}$$

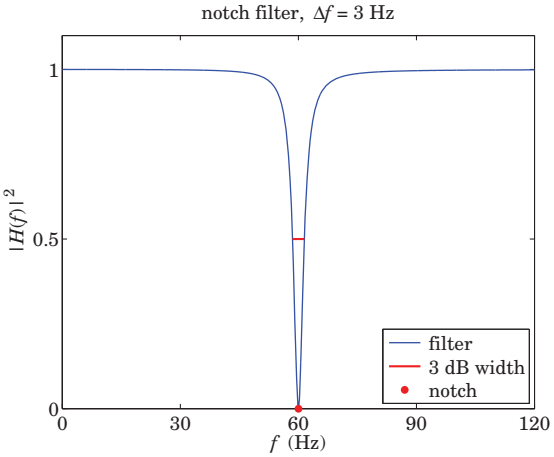
In the magnitude response graphs below, the 3-dB frequencies are indicated by the short red horizontal lines drawn at the half-power levels.

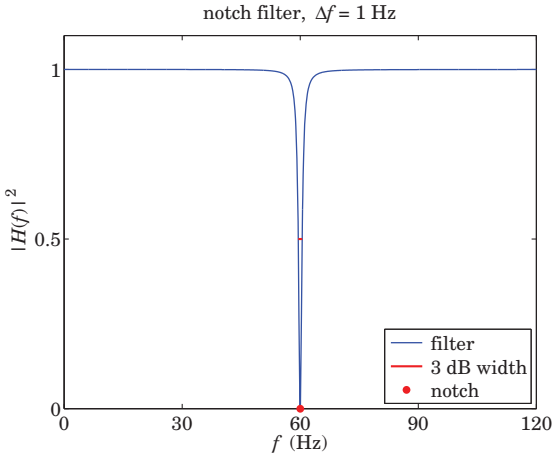
```
x0 = load('ecg0.dat');       % true ECG - for comparison only
x = load('ecg60.dat');       % noisy ECG

fs = 1000; T = 1/fs;         % sampling rate

N = length(x);
t = (0:N-1)*T;

f0 = 60;                     % Hz
w0 = 2*pi*f0;                % rad/sec

f = linspace(0,120,2401);  % frequency range for plotting |H(f)|^2
w = 2*pi*f;

figure; plot(t,x,'b');
title('noisy ECG');
xaxis(0,2,0:0.5:2); yaxis(-2,4,-2:1:4);
xlabel('{\itt}  (sec)'); ylabel('{\itx}({\itt})');

for Df = [3,1]                           % 3-dB widths in Hz
    Dw = 2*pi*Df;
    f1 = -Df/2 + sqrt(f0^2 + Df^2/4);    % left 3dB bandedge
    f2 = +Df/2 + sqrt(f0^2 + Df^2/4);    % right 3dB bandedge

    s = tf('s');                         % transfer function variable
    H = (s^2+w0^2)/(s^2 + Dw*s + w0^2);  % transfer function object
    y = lsim(H,x,t);                     % run with zero initial state

    figure; plot(t,y,'b', t,x0,'r-');
    title(['filtered ECG,  \Delta{\itf} = ',num2str(Df),' Hz']);
    xaxis(0,2,0:0.5:2); yaxis(-2,4,-2:1:4);
    xlabel('{\itt}  (sec)'); ylabel('{\ity}({\itt})');
    legend(' filtered', ' true ECG','location','se');

    H2 = (w.^2-w0^2).^2 ./ ((w.^2-w0^2).^2 + Dw^2*w.^2);

    figure; plot(f,H2,'b-'); hold on
    title(['notch filter,  \Delta{\itf} = ',num2str(Df),' Hz']);
    xaxis(0,120, 0:30:120); yaxis(0,1.1, 0:0.5:1);
    xlabel('{\itf}  (Hz)'); ylabel('|{\itH}({\itf})|^2');
    plot([f1,f2],[0.5,0.5], 'r-', 'linewidth',2)
    plot(f0,0,'r.','markersize',22)
    legend(' filter', ' 3 dB width', ' notch','location','se');
    hold off
end
```

In future sets, we will discuss how to discretize second- and higher-order systems using a variety of discretization schemes such as the zero-order hold, forward and backward Euler methods, and the trapezoidal/bilinear transformation method. It should be noted that the built-in function **lsim** uses either a zero-order or a first-order hold, whichever is more appropriate for the input at hand.


filtered ECG, $\Delta f = 3$ Hz
$y(t)$
t (sec)
filtered
true ECG


filtered ECG, $\Delta f = 1$ Hz
$y(t)$
t (sec)
filtered
true ECG


notch filter, $\Delta f = 3$ Hz
$|H(f)|^2$
f (Hz)
filter
3 dB width
notch


notch filter, $\Delta f = 1$ Hz
$|H(f)|^2$
f (Hz)
filter
3 dB width
notch