Basic Analog Electric Filters: A Handshake

Prepared for ECE 301
Fall Semester, 2002

Prepared By
Walter L. Green
Professor Emeritus, Electrical Engineering
University of Tennessee
Knoxville, TN

Basic Concepts of Analog Filters:

A Handshake

Background: Four filter shapes are normally defined in basic electric filters. These are shown in Figure 1: they are called ideal filters.

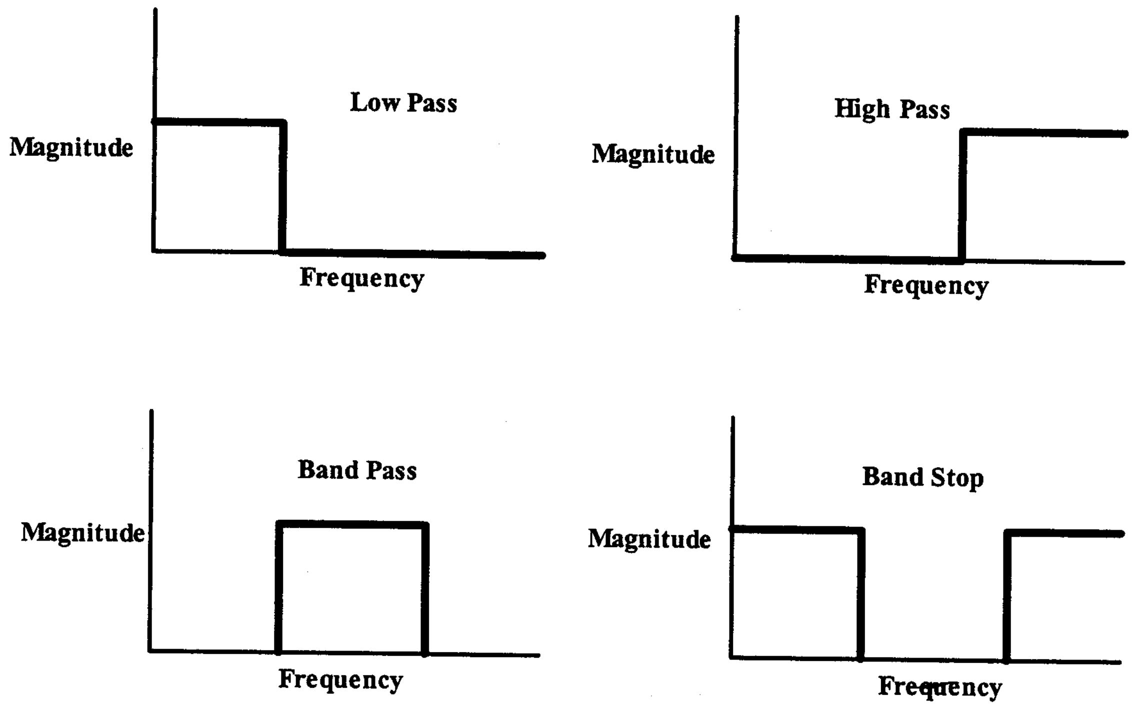


Figure 1: Profile of ideal filters.

Ideal filters are not realizable. Their shape can be approached as the order of the filter becomes very large. Normally, one accepts a filter that is not ideal and has a manageable order. Profiles of more realistic filters are shown in Figure 2.

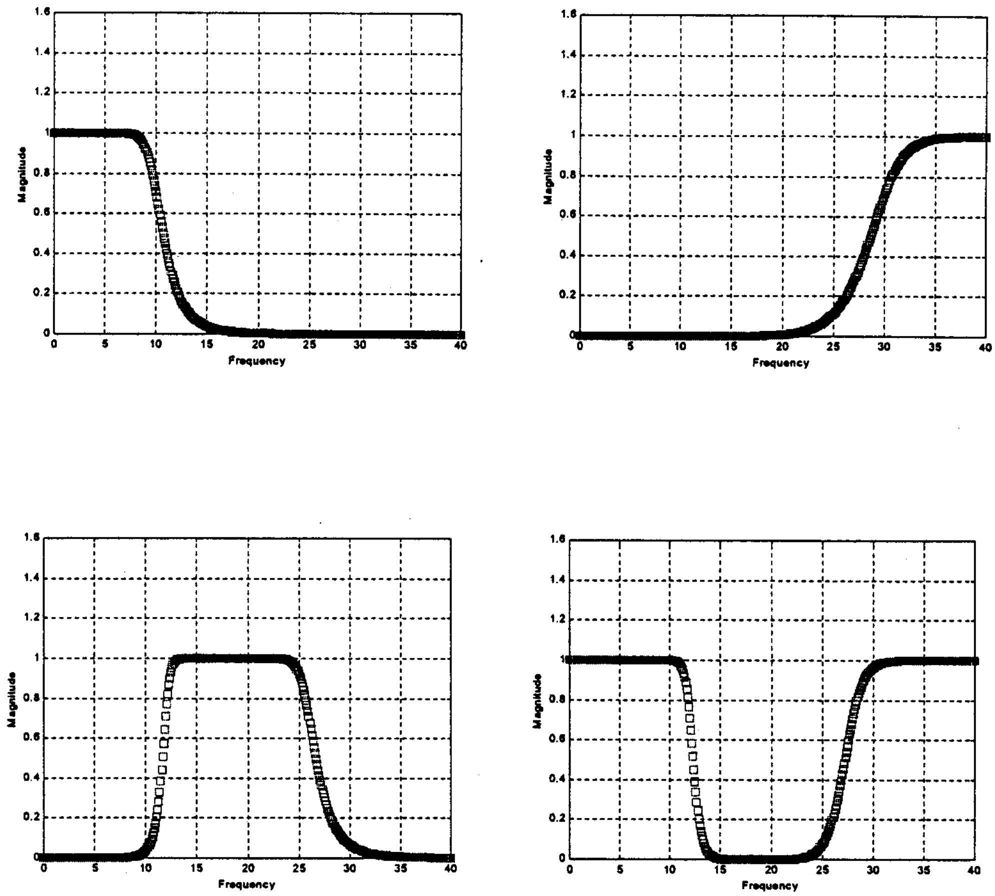


Figure 2: Typical filter profiles: 8th order Butterworth. See the Appendix.

Analytically, filters are defined from transfer functions. A transfer function expressed in factor form might appear as in Equation (1).

$$\frac{Y_0(s)}{Y_{in}(s)} = \frac{K(s - z_1)(s - z_2)\ldots(s - z_m)}{(s - p_1)(s - p_2)\ldots(s - p_n)} \qquad \text{Eq (1)}$$

The roots of the numerator are called zeros while roots of the denominator are called poles. In this case there are m zeros and n poles. Equation (1) may also be expressed in polynomial form. This form is shown in Equation (2).

$$\frac{Y_0(s)}{Y_{in}(s)} = \frac{K[s^m + a_{m-a}s^{m-1} + ... + a_1 s + a_o]}{s^n + b_{n-1}s^{n-1} + .. + b_1 s + b_o} = G(s) \qquad \text{Eq (2)}$$

The order of the denominator polynomial of Equation (2) is said to be the order of the transfer function (nth order in this case) and also called the order of the filter.

To find the frequency response corresponding to G(s) in a laboratory setting, the function $Y_{in}(s)$ (actually $y_{in}(t)$) becomes the output signal from a signal generator that can supply sinusoidal waveforms. For each frequency input, the response (magnitude and phase) are recorded. A display of the magnitude vs frequency and phase vs frequency becomes the frequency response.

To find the frequency response with a computer, one replaces s by $j\omega$ and typically runs a program to determine the absolute value of $G(j\omega)$ and the angle of $G(j\omega)$ as $0 < \omega < \infty$.

As an example, a second order low pass Butterworth filter with a cutoff of w = 10 rad/sec is given by the expression,

$$\frac{Y_0(s)}{Y_{in}(s)} = \frac{100}{s^2 + 14.14s + 100} \qquad \text{Eq (3)}$$

The MATLAB function H = freqs(Num,Den,w) can be used to obtain the frequency response. This response, for both the magnitude and phase is shown in Figure 3 of the following page.

Equation (3) may be rearranged to the following form;

$$(s^2 + 14.14s + 100)Y_0(s) = 100Y_{in}(s) \qquad \text{Eq (4)}$$

Equation (4) can be expressed as a differential equation as below.

$$\frac{d^2 y_0(t)}{d^2 t} + 14.14\frac{dy_0(t)}{dt} + 100y_0(t) = 100y_{in}(t) \qquad \text{Eq (5)}$$

Let,

$$y_{in}(t) = A\cos(wt) \qquad \text{Eq (6)}$$

To solve for the steady state solution (particular solution) of Equation (5) for i sinusoidal inputs, we assume $y_{in}(t) = y_{P_i}(t)$. An input of the following form shown in Equation (7) is assumed.

$$y_{P_i} = B_{1i}\cos(w_i t) + B_{2i}\sin(w_i t) \qquad \text{Eq (7)}$$

This leads to the solution,

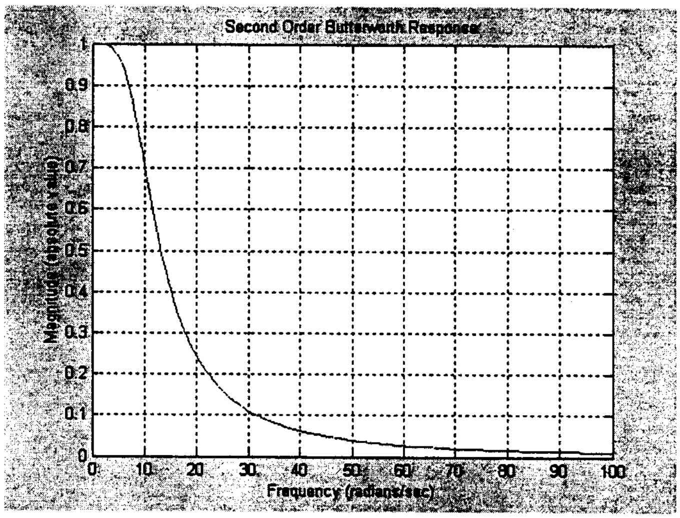
$$y_{Pi}(t) = C_i\cos(w_i t + \theta_i) \qquad \text{Eq (8)}$$

$$y_{P_T}(t) = \sum_{i=1}^{i=i_{max}} y_{P_i}(t) = \sum_{i=1}^{i=i_{max}} C_i\cos(w_i t + \theta_i)$$

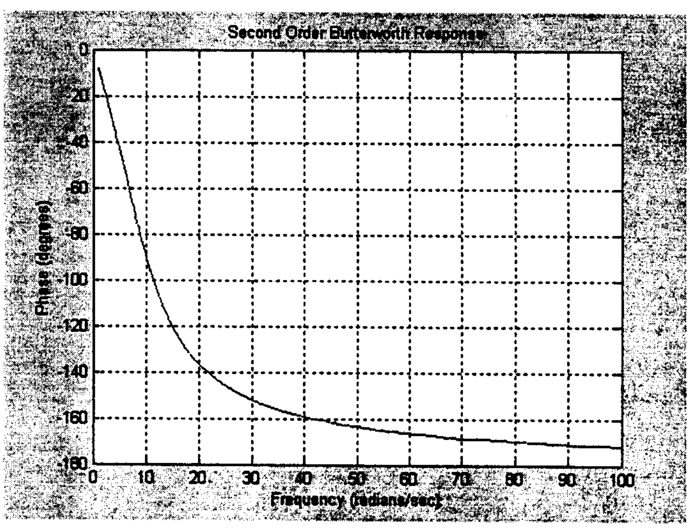
For every $w = w_i$, we get a solution for $y_{P_i}(t)$. The frequency response is a display of the C_i and the θ_i. The C_i are magnitudes and represent the magnitude profile of what we call a filter. The θ_i are the phase angle of the filter. Thus, the following statement can be made.

An analog frequency response (filter) is a graphical solution of a linear differential equation which has sinusoidal forcing functions, $(A_i)\cos(w_i t)$, and output $(C_i)\cos(w_i t + \theta_i)$ $1 < i < \infty$.

The C_i are the filter magnitude and the θ_i are the filter phase.



(a) magnitude of response



(b) phase of response

Figure 3: Magnitude and phase for a filter. 2nd order analog Butterworth filter.

We now present an example that shows how this definition is used.

Example 1:

The signal $y_i(t) = 4 \cos 10t$ is applied to the filter shown in Figure 4. Determine $y_0(t)$.

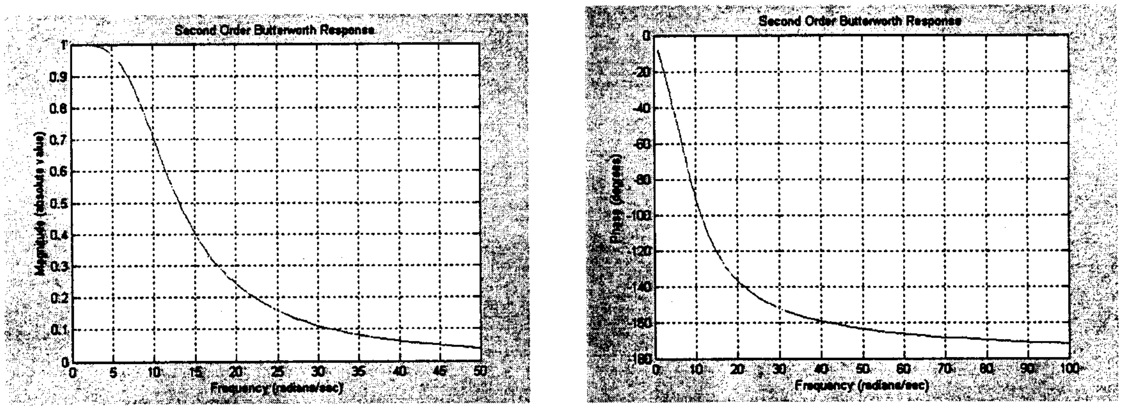
Solution: From the filter magnitude and phase one sees that at $\omega = 10$ rad/sec, the magnitude from the filter response is 0.71 and the phase angle is $-90°$. This means that

$$4x(0.7)\cos(10t - 90°) \qquad \text{Eq (9)}$$

One seldom ever writes that this is the solution for a differential equation for $\omega = 10$ rad/sec (but that is exactly what it is). Rather, one thinks in terms of what happens to various signals, with frequency ranging from $0 < \omega < \infty$, that are applied to the filter. The filter response of Figure 4 gives the answer as to what happens.

The next level is learning how to design filters that give desired shapes. An enormous amount of work has been done in the filtering area. This work was developed for analog filters but is easily extended to digital filters. At any rate, we usually rely on information provided by standard methods available from,

Butterworth Design	Chebyshev I, II Design
Elliptic Design	Park-McCellan Design



(a) magnitude (b) phase

Figure 4: Frequency response information for a filter.

As a simple example on designing a low pass filter of the Butterworth form, we consider the following example.

Example 2:

Design a Butterworth filter that has the property that

(i) no signal less than 10 Hz is attenuated more than 3 dB

(ii) all signals above 25 Hz are attenuated by at least 20 dB

Give the filter transfer function and show the frequency response.

Solution: When given specifications, such as the above, it is a good idea to sketch a specification diagram. This diagram has been prepared and is shown in Figure 5. The program used for designing the filter, entirely within MATLAB, is shown in Figure 6. The function **buttord** is used to determine the order of the filter needed to meet the specifications. In this case a 3 order filter was required.

Next, the function **butter** was used to determine the transfer function, in this case, Num, Den. Num, Den are used as the input for **freqs** to determine the frequency response of the filter.

The frequency response resulting from the design is shown in Figure 7. The specification diagram lines have been superimposed on the plot to show that the filter meets specifications.

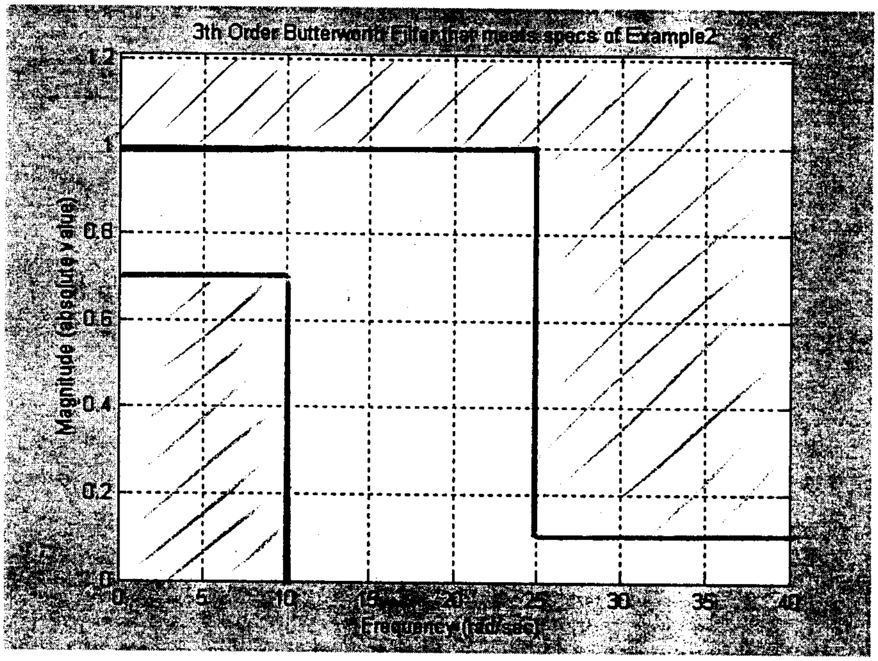


Figure 5: Specification diagram for Example 2.

```
% example to determine the order of a filter
% to meet specifications and show the filter
% response:  program butter_search.m  ECE 301,
% Fall 2002,  wlg, office computer

 N = buttord(10,25, 3, 20,'s');
[Num,Den] = butter(4,10,'s');

 w = 0:.05:60;
[H,w] = freqs(Num,Den,'s');
Hmag = abs(H);
plot(w,Hmag)
grid
axis([0 40 0 1.21])
title('3th Order Butterworth Filter that meets specs of Example2')
ylabel('Magnitude (absolute value)')
xlabel('Frequency (rad/sec)')
```

Figure 6: MATLAB used for designing filter for Example 2

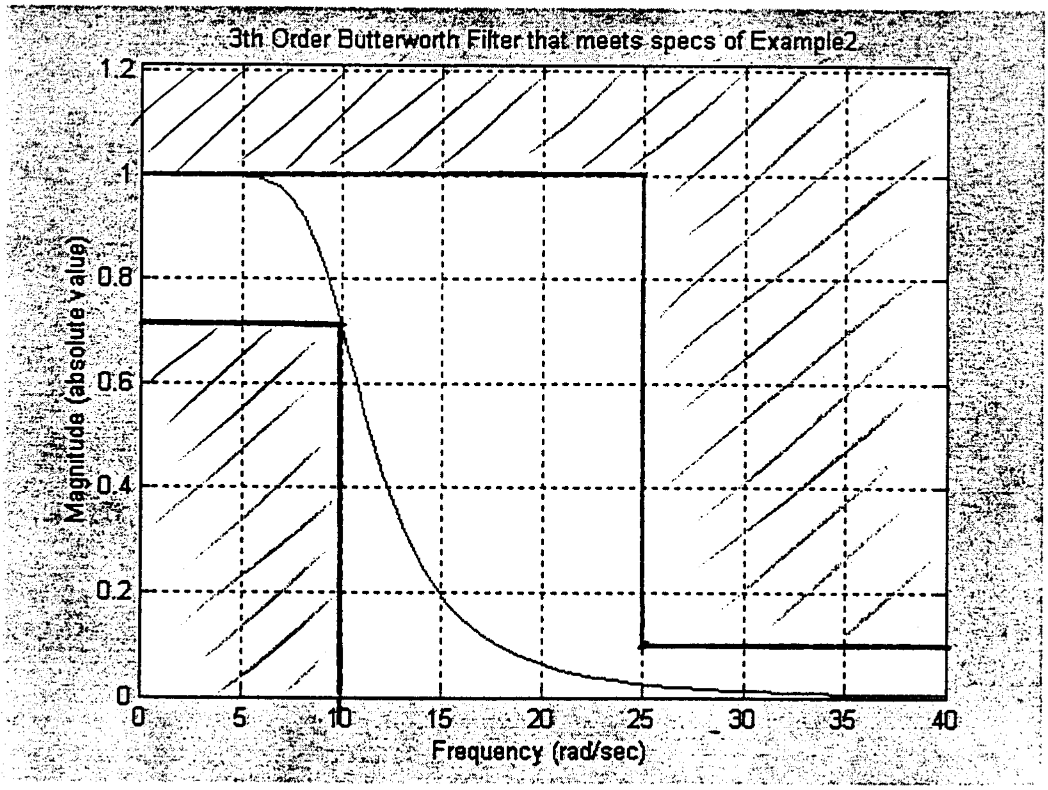


Figure 7: Filter response for Example 2.

<u>Transfer Functions and Filters</u>: The "building" of filters from given transfer functions is not an easy task, at least not in the general case. People have devoted their lives to this subject, under the heading of ***<u>Network Synthesis</u>***. We will only concern ourselves with two of the simplest filters and build a circuit for each. The two filters are first order low pass and high pass.

<u>A Low Pass Filter</u>: Consider the RC circuit show in Figure 8. We first derive an expression for the transfer function of $V_0(s)/V_{in}(s)$ using the voltage

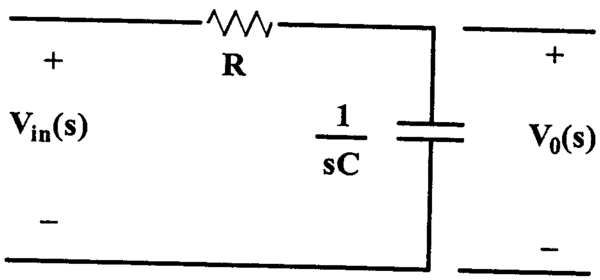


Figure 8: A basic first order low pass filter.

divider rule of basic circuits. We have,

$$\frac{V_0(s)}{V_{in}(s)} = \frac{1/sC}{R + 1/sC} = \frac{1}{1 + sRC} = \frac{1/RC}{s + 1/RC} \qquad \text{Eq (10)}$$

The magnitude profile of this filter fits that of a low pass filter. The cut-off point (-3dB point) is located at $\omega = 1/RC$. One only must select the cut-off frequency and satisfy $\omega = 1/RC$. The normal case is for C to be specified and then R can be calculated. This is a simple design. This first order low-pass filter is important in signal processing. Many digital filters today are aliasing, one must provide an analog guard filter on the incoming data stream to the computer. The first order low-pass filter plays this role.

A High Pass Filter: Interchanging the positions of the resistor and capacitor in the previous circuit give the filter of Figure 8. A little though will show that the response of this filter fits that of a high pass response.

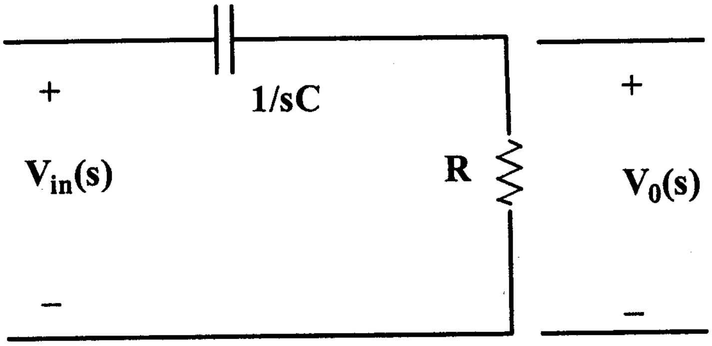


Figure 7: A basic first order high pass filter.

The transfer function of the filter is,

$$\frac{V_0(s)}{V_{in}(s)} = \frac{R}{R+1/sC} = \frac{sRC}{sRC+1} = \frac{s}{s+1/RC} \qquad \text{Eq (11)}$$

The filter cut-off is a w = 1/RC. One selects the desired high frequency cutoff point and solves for R when C is given (C is usually the given parameter).

These notes give just a brief introduction to analog filtering, ***a handshake***. For the most part, filtering today is done digitally. However, it is not difficult to take the start presented here and venture on to digital filters. Good luck.

Appendix

The MATLAB programs used for producing the profile of typical filters is given below.

```
% basic filter profiles
% program name: filter_profile.m

%[nhp,dhp]=butter(12,30,'high','s');
%w=0:.1:40;
%Hhp = freqs(nhp,dhp,w);
%Hmaghp = abs(Hhp);
%plot(w, Hmaghp,'s')
%grid
%axis([0 40 0 1.6])
%ylabel('Magnitude')
%xlabel('Frequency')

%[nlp,dlp]=butter(8,10,'s');
%w=0:.1:40;
%Hlp = freqs(nlp,dlp,w);
%Hmaglp = abs(Hlp);
%plot(w, Hmaglp,'s')
%grid
%axis([0 40 0 1.6])
%ylabel('Magnitude')
%xlabel('Frequency')

%[nbp,dbp]=butter(8,[12,26],'bandpass','s');
%w=0:.1:40;
%Hbp = freqs(nbp,dbp,w);
%Hmagbp = abs(Hbp);
%plot(w, Hmagbp,'s')
%grid
%axis([0 40 0 1.6])
%ylabel('Magnitude')
%xlabel('Frequency')

[nsp,dsp]=butter(8,[12,28],'stop','s');
w=0:.1:40;
Hsp = freqs(nsp,dsp,w);
Hmagsp = abs(Hsp);
plot(w, Hmagsp,'s')
grid
axis([0 40 0 1.6])
ylabel('Magnitude')

xlabel('Frequency')
```