Analysis of Linear Systems using MATLAB

This tutorial describes some of the MATLAB commands used to design filters and find the output of a linear system.

The aperiodic pulse shown below:

has a Fourier transform:

\[ X(j\omega) = 4 \text{sinc}(4\pi \omega) \]

As shown in MATLAB Tutorial #2, we can plot the amplitude and phase spectrum of this signal as follows:

\[
\begin{align*}
\text{>> } & f=-5:.001:5; \\
\text{>> } & X=4\times\text{sinc}(4\times f); \\
\text{>> } & \text{subplot}(1,2,1) \\
\text{>> } & \text{plot}(f,\text{abs}(X)) \\
\text{>> } & \text{subplot}(1,2,2) \\
\text{>> } & \text{plot}(f,\text{angle}(X))
\end{align*}
\]
Determining the Output of a Linear System

Assume that the pulse \( x(t) \) is applied to the low pass filter shown below.

\[
\begin{array}{c}
R \\
\text{+} \\
\hline
\text{v}_i \\
\hline
\text{C} \\
\text{+} \\
\hline
\text{v}_o
\end{array}
\]

This filter has a frequency response function:

\[
H(jf) = \frac{1}{1 + j2\pi fRC}
\]

If we assume that \( R = 150K\Omega \) and \( C = 1\mu F \), we can plot the magnitude and phase of the frequency response using MATLAB:

```matlab
>> R=1.5E6;
>> C=1E-6;
>> H=1./(1+j*2*pi*f*R*C);
>> subplot(1,2,1)
>> plot(f,abs(H))
>> subplot(1,2,2)
>> plot(f,angle(H))
```

If we apply the pulse, \( x(t) \) to this filter, the Fourier transform of the output can be found using the relationship:

\[
Y(f) = H(f)X(f)
\]
or, using MATLAB:

\[
\begin{align*}
\text{>> } & \text{Y=H.*X;} \\
\text{>> } & \text{subplot(1,2,1)} \\
\text{>> } & \text{plot(f,abs(Y))} \\
\text{>> } & \text{subplot(1,2,2)} \\
\text{>> } & \text{plot(f,angle(Y))}
\end{align*}
\]

The output waveform, \( y(t) \) can be determined by taking the inverse Fourier transform of \( Y(f) \), or:

\[
\begin{align*}
\text{>> k=0;} \\
\text{>> for } t=-2:.01:2 \\
\quad k=k+1; \\
\quad y(k)=\text{trapz}(f,Y.*\exp(j*2*\pi*f*t)); \\
\text{end} \\
\text{>> t=-2:.01:2;} \\
\text{>> plot(t,y)}
\end{align*}
\]

Note that the pulse has become distorted due to the fact that the low pass filter has removed a significant amount of the high frequency content of the signal. Decreasing the value of RC can increase the cutoff frequency of the filter. For example, if we decrease R to 150K\( \Omega \) the resulting output will be:
For $R = 15\, \text{K}\Omega$:

As expected, with decreasing $R$ the output waveform approaches the input pulse waveform.

**Designing Filters in MATLAB**

In the study of communication systems you will find that most modulators and demodulators require some type of filter. Using the Signal Processing Toolbox many sophisticated filters of varying types can be designed and implemented. Since this is not a course in filter design (that will be covered in ECE 351) we will use a simple built in filter design command in MATLAB which produces Butterworth filters. This simple method will be described here with an example a two-frequency signal:

$$x(t) = \cos(2\pi 100t) + \cos(2\pi 500t)$$
We are going to design a low pass filter to pass the 100 Hz portion of the signal and block the 500 Hz portion.

1. **Create an input vector:** In order to obtain a smooth curve from the input vector we need to have enough points in the input vector. Since the period of the highest frequency component of the input is 1/500, we will sample the input every .2 mS, or 10 times in this period. Using the following MATLAB commands:

   ```matlab
   >> t=0:2E-4:.05;
   >> x=cos(2*pi*100*t)+cos(2*pi*500*t);
   >> plot(t,x)
   ```

   we produce the following input vector:

   ![Input Vector Graph](image)

2. **Specify an ideal filter to meet your specifications:** In this example we would like to pass the 100 Hz signal and block the 500 Hz signal. Thus, we can use an ideal low pass filter with a cutoff frequency of 300 Hz as shown below.

   ![Ideal Filter](image)

3. **Design the filter using a Butterworth filter:** The following `butter` command in MATLAB designs a Butterworth low pass filter of order n and cutoff frequency wc:

   ```matlab
   [b, a] = butter(n, wc, 's');
   ```
where \( b \) and \( a \) are the numerator and denominator coefficients of the transfer function of the resulting filter. The ‘s’ denotes that an analog filter will be designed. This command can also be used to design digital filters.

\[
[b, a] = \text{butter}(5, 2\pi \times 300, \text{‘s’});
\]

will design a 5\(^{\text{th}}\) order filter with a 300 Hz cutoff frequency. The input cutoff frequency must be in rad/sec.

4. **Verify the filter design and adjust \( n \) as needed**: The filter design can be verified by plotting its frequency response as compared to the ideal filter using the \( \text{freqs} \) command. This command is in the form:

\[
H = \text{freqs}(b, a, w)
\]

where \( w \) is a vector of frequencies at which the frequency response function will be calculated. The following commands will design and plot the frequency response of a 5\(^{\text{th}}\) order Butterworth low pass filter. Note that the plot is in Hz, not rad/sec.

\[
[b, a] = \text{butter}(5, 2\pi \times 300, \text{‘s’});
\]
\[
H = \text{freqs}(b, a, 2\pi \times [0:800]);
\]
\[
\text{plot}(0:800, \text{abs}(H))
\]

We can confirm that the cutoff frequency is in fact at 300 Hz using:

\[
\text{abs}(H(301))
\]

\[
\text{ans} = 0.7071
\]
Value 301 corresponds to 300 Hz in the frequency vector. For larger values of n the filter approaches the ideal behavior more closely. Butterworth filters of order 10 and 20 are shown along with the original design below.

```matlab
>> [b,a]=butter(10,2*pi*300,'s');
>> H=freqs(b,a,2*pi*[0:800]);
>> hold on
>> plot(0:800, abs(H), 'r')
>> [b,a]=butter(20,2*pi*300,'s');
>> H=freqs(b,a,2*pi*[0:800]);
>> plot(0:800, abs(H), 'g')
```

5. **Test the filter**: We will use the filter of order 20 to remove the 500 Hz component. The `lsim` command can be used to simulate a linear time invariant system. The general form for the `lsim` command is:

\[ y = \text{lsim}(Hs, x, t) \]

where Hs is the transfer function of the system. The transfer function can be generated from the b and a vectors using the `tf` command:

\[ Hs = \text{tf}(b, a) \]

In our example the filter output is found with the following MATLAB commands:

```matlab
>> [b,a]=butter(20,2*pi*300,'s');
>> Hsys=tf(b,a);
>> y=lsim(Hsys, x, t);
>> hold off
>> plot(t,y)
```
Note that the steady state part of the output has a period of 0.01 seconds or a frequency of 100 Hz. This confirms that the 500 Hz component has been filtered out.

6. **Other types of Butterworth Filters**: The butter command can also be used to design high pass, bandpass and bandstop filters. The commands are as follows:

   \[
   \begin{align*}
   \text{[b, a]} &= \text{butter}(n, wc, \text{`high', `s'}) \quad \text{for a high pass filter} \\
   \text{[b, a]} &= \text{butter}(n, [wc1, wc2], \text{`s'}) \quad \text{for a bandpass filter with cutoff frequencies at } wc1 \text{ and } wc2 \text{ rad/sec}
   \end{align*}
   \]