Chapter 3 MATLAB Frequency Response Example

A couple years ago one student asked if I could put together some of the MATLAB commands I used in obtaining the discrete-time \(G(z)\) using the integration rules, and for finding the frequency response (magnitude and phase). I’ll even show a little more than that. So here it is...

1. System under Consideration. Let’s consider a second-order system with damping ratio \(\zeta = 0.1\) and natural frequency \(f_n = 5\) Hz. The low damping ratio will give us a resonant peak in the frequency response (as well as much overshoot in the step response if we were interested in that). We know that the continuous-time transfer function will be

\[
G(s) = \frac{s^2}{s^2 + 2\zeta\omega_n s + \omega_n^2} \tag{1}
\]

Note that this \(G(s)\) has a DC gain of 1.

1.1 Continuous System. First we’ll construct the continuous-time system \(G(s)\) in MATLAB (one way to do it):

```matlab
>> zeta = 0.1; % Define damping ratio
>> fn = 5; % Define natural frequency (Hz)
>> wn = 2*pi*fn; % Convert to rad/s
>> numGc = [0 0 wn^2]; % Create numerator of G(s)
>> denGc = [1 2*zeta*wn wn^2]; % Create denominator of G(s)
>> Gc = tf(numGc,denGc) % Create transfer function G(s); call it Gc
```

Transfer function:

\[
\frac{987}{s^2 + 6.283 s + 987}
\]

1.2 Discrete System. Next we’ll find the trapezoidal discretization; that one is supported by MATLAB. But before that we have to select a sampling frequency. An old “rule of thumb” is to sample 6–20 times faster than the natural frequency, so let’s choose \(f_s = 50\) Hz (\(T = 0.02\) sec).

```matlab
>> fs = 50; % Sampling frequency (Hz)
>> T = 1/fs; % Sampling period (sec)
>> Gtrap = c2d(Gc,T,'tustin') % Find trapezoidal (Tustin) discretization
```

Transfer function:

\[
\frac{0.08497 z^2 + 0.1699 z + 0.08497}{z^2 - 1.552 z + 0.8918}
\]

Sampling time: 0.02

You might be interested in the poles and zeros of the \(G(z)\); the \texttt{zpk()} function helps a little:

```matlab
>> zpk(Gtrap)
```

Zero/pole/gain:
\[
\frac{0.084971 \ (z+1)^2}{(z^2 - 1.552z + 0.8918)}
\]

Sampling time: 0.02

So this \(G(z)\) has two zeros at \(z = -1\), but we still don’t know where the poles are! The \texttt{zpk} function won’t factor a complex denominator (or numerator)! I have no idea why not, but what you must do is extract the numerator and denominator using the \texttt{tfdata()} function, then use function \texttt{roots()} to find the roots (poles). Like this:

\[
\text{>> } [\text{numtrap}, \text{dentrap}] = \text{tfdata}(\text{Gtrap},'v') \quad \text{The 'v' MUST be present or you get a cell array}
\]
\[
\text{numtrap} = 0.0850 \quad 0.1699 \quad 0.0850
\]
\[
\text{dentrap} = 1.0000 \quad -1.5519 \quad 0.8918
\]

\[
\text{>> } \text{poles} = \text{roots}(\text{dentrap}) \quad \text{% Find the roots of the denominator (poles)}
\]
\[
\text{poles} = 0.7760 + 0.5382i
\]
\[
0.7760 - 0.5382i
\]

You don’t really have to find the poles in this case, but that’s how.

2. Frequency Response. Let’s find the frequency response (mag, phase) at 75\% of the natural frequency, which is \(f = 3.75\) Hz. The magnitude \(|G|\) is the steady-state amplitude ratio, while the phase \(\angle G\) is the phase shift. MATLAB returns magnitude as a “pure” number (not dB), and returns phase in degrees. We’ll convert magnitude to dB, and we’ll do this for both the original continuous system \(G(s)\) and the discrete system \(G(z)\).

2.1 Continuous system. Here we work with \(G(s)\):

\[
\text{>> } f = 3.75; \quad \text{% Frequency of interest (Hz)}
\]
\[
\text{>> } w = 2\times\pi\times f; \quad \text{% Convert to rad/s}
\]
\[
\text{>> } [\text{magc, phasic}] = \text{bode}(\text{Gc}, w) \quad \text{% Find the frequency response at } f = 3.75 \text{ Hz}
\]
\[
\text{magc} = 2.1622 \quad \text{% Magnitude of } G(s) \text{ as a pure number, somewhat greater than 1 (DC value)}
\]
\[
\text{phasic} = -18.9246 \quad \text{% Phase lag of } G(s) \text{ is around 19 degrees}
\]

\[
\text{>> } \text{magcdB} = 20\times\text{log10}(\text{magc}) \quad \text{% Remember that log10 is the base 10 log function}
\]
\[
\text{magcdB} = 6.6978 \quad \text{% Magnitude of } G(s) \text{ is around 7 dB (DC value is 0 dB)}
\]

2.2 Discrete system. Here we work with \(G(z)\):

\[
\text{>> } [\text{magtrap, phasetrap}] = \text{bode}(\text{Gtrap}, w) \quad \text{% Same function, but } G(z) = G\text{trap}
\]
\[
\text{magtrap} = 2.2563 \quad \text{% Both the magnitude and phase are}
\]
\[
\text{phasetrap} = -20.1731 \quad \text{% pretty close to } G(s)
\]

\[
\text{>> } \text{magtrapdB} = 20\times\text{log10}(\text{magtrap})
\]
\[
\text{magtrapdB} = 7.0681 \quad \text{% Magnitude of } G(z) \text{ in dB...pretty close}
\]

So that’s how to perform the discretization and find the frequency response at a single frequency point.
However, we usually want to make Bode plots of the frequency response data. This can be done with the same \texttt{bode} function in MATLAB.

The simplest way is to let MATLAB select the frequency range (we'll use \(G(s)\) here):

\[
\texttt{>> bode}(Gc); \texttt{grid}; \% I like to put a grid on the plot}
\]

This will produce the plot shown below.

![Bode Diagram](image)

This plot is nice (note that the magnitude is plotted in dB) but if you wish to select the frequency range, you must pass a vector of frequencies to the \texttt{bode} function. The vector of frequencies should be formed using the \texttt{logspace()} function to get logarithmic spacing. The \texttt{logspace()} function accepts powers of 10 to determine this range, as well as a number of points. There's also the issue of Hz vs rad/s (Problem 3c).

To construct a frequency range of \(1 \leq f \leq 1000\) Hz with 200 points (just as an example), then convert it to rad/sec and pass it to \texttt{bode()} you would do the following (I'm not going to show the plot):

\[
\texttt{>> f = logspace}(0,3,200); \% Construct vector of frequencies (Hz) from 10^0 \rightarrow 10^3 \text{ w/200 pts}
\texttt{>> w = 2*pi*f; \% Convert to rad/s}
\texttt{>> bode}(Gtrap,w); \% Find Bode plots of }G(z)\ldots\text{this might look a little strange!}
\]

3. Conclusion. So there is an example of the MATLAB functions used to find and plot the frequency response characteristics of both a continuous system \(G(z)\) and its discrete equivalent \(G(z)\). Hope this helps!

\textbf{NOTE:} If you use the \texttt{bode()} function with returned arguments, like
>> [mag,phase] = bode(Gtrap,w);

you will need to remove the “singleton” extra dimension from returned arrays mag and phase. Like this:

>> mag = squeeze(mag); % Remove singleton dimension
>> phase = squeeze(phase); % Here, too...

This didn’t used to be necessary, but that’s progress.