Christopher Lum
lum@uw.edu

Lecture 03a
Time Domain Analysis With Matlab: Using the Linear System Analyzer

The YouTube video entitled 'Time Domain Analysis With Matlab: Using the Linear System Analyzer' that covers this lecture is located at https://youtu.be/P5fcgnaYIeQ.

Outline

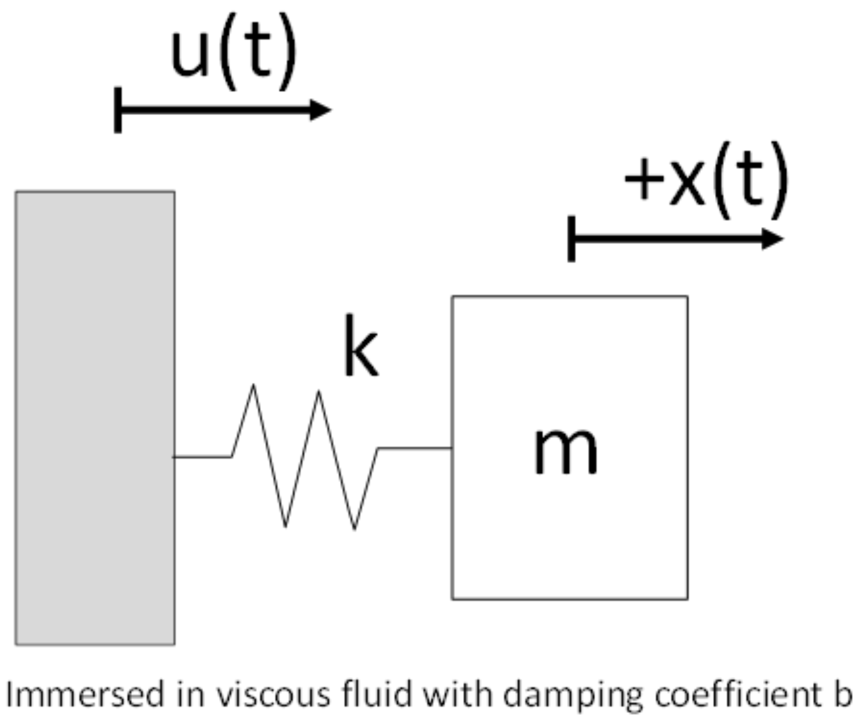
-Time Domain Analysis in Matlab

Time Domain Analysis in Matlab

Using the 'step' command
Using the 'lsim' command
Using 'linearSystemAnalyzer' app (link)

Example: Mass Spring Damper System.

Let us consider a slight variation on the mass spring damper system. Instead of inputting a force on the mass m, the position of the rod on the left is controlled directly. The deflection of the left rod is defined as the input $u(t)$. Furthermore, the mass and the rod are not connected by a damper but rather the mass is immersed in a fluid with viscous damping coefficient b.



Equations of motion of this system are given by

$$m\,\ddot{x}(t) = k(u(t) - x(t)) - b\,\dot{x}(t)$$

$$m\,\ddot{x}(t) = k\,u(t) - k\,x(t) - b\,\dot{x}(t)$$

$$\ddot{x}(t) = \frac{k}{m}\,u(t) - \frac{k}{m}\,x(t) - \frac{b}{m}\,\dot{x}(t)$$

$$\ddot{x}(t) + \frac{b}{m}\,\dot{x}(t) + \frac{k}{m}\,x(t) = \frac{k}{m}\,u(t)$$

We can calculate the transfer function

$$s^2\,X(s) + \frac{b}{m}\,s\,X(s) + \frac{k}{m}\,X(s) = \frac{k}{m}\,U(s)$$

$$\left(s^2 + \frac{b}{m}\,s + \frac{k}{m}\right)X(s) = \left(\frac{k}{m}\right)U(s)$$

$$G(s) = \frac{X(s)}{U(s)} = \frac{\frac{k}{m}}{s^2 + \frac{b}{m}\,s + \frac{k}{m}}$$

Or writing this in the standard form, we have

$$G(s) = \frac{\omega_n^2}{s^2+2\,\zeta\,\omega_n\,s+\omega_n^2}$$

```
temp = Solve[{b/m == 2 ξ ωn, k/m == ωn^2}, {ξ, ωn}];
ξ = ξ /. temp[[2]]
ωn = ωn /. temp[[2]]
```

$$\frac{b}{2\sqrt{k}\,\sqrt{m}}$$

$$\frac{\sqrt{k}}{\sqrt{m}}$$

Let us choose constants of

$$m = 2$$
$$k = 50$$
$$b = 4$$

```
m = 2;
k = 50;
b = 4;
```

So we obtain

$$\zeta = 1/5$$
$$\omega_n = 5$$

$$G(s) = \frac{25}{s^2+2\,s+25}$$

```
ξ
ωn
```

$$\frac{1}{5}$$

5

For the remainder of this analysis, we will subject the system to a step input of magnitude $A = 1.5$.

We can find the DC gain of the system, which is really the steady state response of the system with respect to a unit step input.

$$\text{DC gain } = G(0) = \frac{\omega_n^2}{\omega_n^2} = 1$$

So we see that the steady state error of this system in response to a step input will be

$$e_{ss} = (1 - G(0))\,A$$

$$= (1 - 1) A$$

$$e_{ss} = 0$$

Now, since the system is underdamped, we can find other performance characteristics

For a second order system, we previously derived various performance metrics. Luckily, this is a second order system.

For the time to the first peak, we have

$$T_p = \frac{\pi}{\sqrt{1-\zeta^2}\ \omega_n}$$

```
π / (Sqrt[1 - ξ^2] ωn) /. {ξ → 1 / 5, ωn → 5} // N
```

```
0.641275
```

For the amplitude of the first peak, we have

$$M_{P_t} = A\left(1 + e^{-\frac{\pi \zeta}{\sqrt{1-\zeta^2}}}\right)$$

```
A (1 + Exp[- π ξ / Sqrt[1 - ξ^2]]) /. {ξ → 1 / 5, ωn → 5, A → 1.5} // N
```

```
2.28993
```

For percent overshoot, we have

$$\text{PO} = 100\, e^{-\frac{\pi \zeta}{\sqrt{1-\zeta^2}}}$$

```
100 Exp[- π ξ / Sqrt[1 - ξ^2]] /. {ξ → 1 / 5, ωn → 5, A → 1.5} // N
```

```
52.6621
```

For settling time, we can use our conservative estimate of $\tilde{T}_s$. This yields (for this problem, we choose a 2% settling time so $\delta = 0.02$)

$$\tilde{T}_s = \frac{-\ln(\delta)}{\zeta\, \omega_n}$$

```
-Log[δ] / (ξ ωn) /. {ξ → 1 / 5, ωn → 5, δ → 0.02} // N
```

```
3.91202
```

Verify Predictions

We can directly implement transfer functions and perform time domain analysis directly from Matlab. For example, consider the system we just analyzed.

$$G(s) = \frac{25}{s^2+2\,s+25}$$

This system can be represented using two arrays

num = (25)
den = (1   2   25)

G = tf(num, den);

Using 'step' command

Go to Matlab, show how you can right click on figure > characteristics (then select which parameters you want to view).

Note that $\tilde{T}_s$ appears to be smaller than T_s computed by Matlab. Because $\tilde{T}_s$ is conservative, it should be larger than numerically calculated values. This discrepancy is due to the numerical inaccuracies in the step command.

Also note that this is a step of magnitude 1, not of magnitude 1.5 so some metrics are inaccurate (like magnitude at first peak)

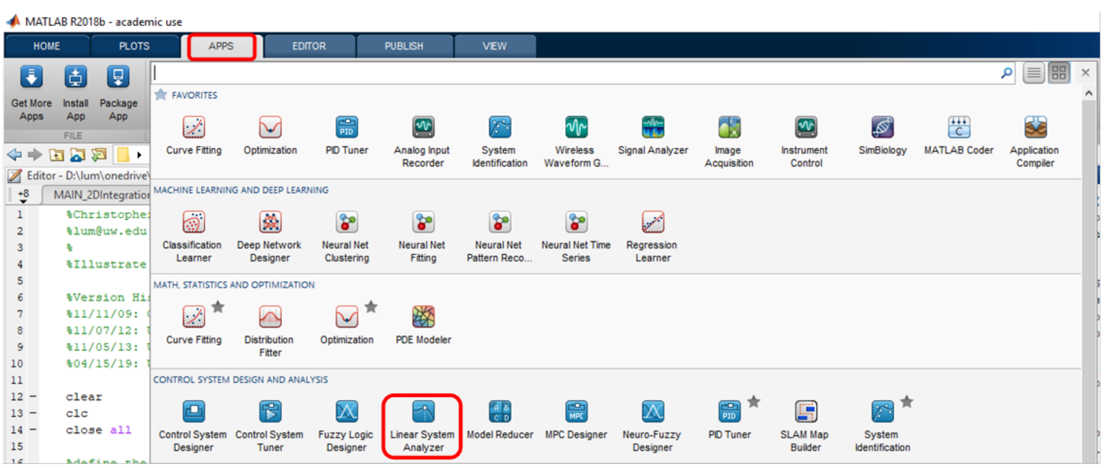
Using 'lsim' command

Show how you can use lsim to input a custom input signal such as

i. step with magnitude A = 1.5
ii. ramp with slope of 5

Using linearSystemAnalyzer

We can also use the 'linearSystemAnalyzer' (link) to analyze the system.



<u>Import Model</u>

File > Import > Workspace > choose system

<u>View Performance Metrics</u>

Right click on plot > Characteristics

Change Performance Metrics
File > Toolbox Preferences > Options
Note that you need to close the app and restart it for the changes to take effect

Add Multiple Analysis Types
Edit > Plot Configurations

Show how you can use the linear simulation plot type to input a specific type of input (like A = 1.5 or a ramp input)

Show how you can change parameters of the model and then view multiple ones simultaneously.

Higher Order Systems

Consider adding a pole of

$$\frac{1/2}{s+1/2}$$

Compare higher order systems with predictions from 2nd order systems. Show that predictions are not exact if it is not a 2nd order system. This is a teaser for the discussion on **dominant poles**.