Module 13 - Higher Order Systems

ME3050 - Dynamics Modeling and Controls

Mechanical Engineering

Tennessee Technological University

Topic 4 - MATLAB Simulation

Topic 4 - MATLAB Simulation

- Control Systems Toolbox
- SS function and Dynamic System Object
- Step, and Impulse Function
- Example: 1DOF Simulation

Control Systems Toolbox

Control System Toolbox provides algorithms and apps for systematically analyzing, designing, and tuning linear control systems. You can specify your system as a transfer function, state-space, zero-pole-gain, or frequency-response model.

It may have been included when you installed MATLAB but if is it not you can easily install it using the add-ins manager in the home tab.

Control Systems Toolbox

Some Toolbox Functions (full list)

- ss
- tf
- pid
- initial
- step
- impulse

The Toolbox also contains specialized blocks for Simulink

SS function and Dynamic System Object

Create a dynamic system object with the **ss()** function

```
[SYS]=ss(A,B,C,D)
```

- Input 1: A - System Matrix
- Input 2: B - Input Matrix
- Input 3: C - Output Matrix
- Input 4: D - Control Matrix
- Output 1: SYS - dynamic system object

SS function and Dynamic System Object

SYS - dynamic system object

Step, and Impulse Function

Simulate the step response of dynamic system with the **step()** function

```
[Y,T]=step(SYS,TSPAN,OPTIONS)
```

- Input 1: SYS - dynamic system object
- Input 2: TSPAN - array of time values for simulation
- Input 3: OPTIONS - created by stepDataOptions function
- Output 1: Y - simulation response as array
- Output 2: T - simulation time as array

Step, and Impulse Function

Simulate the step response of dynamic system with the **step()** function

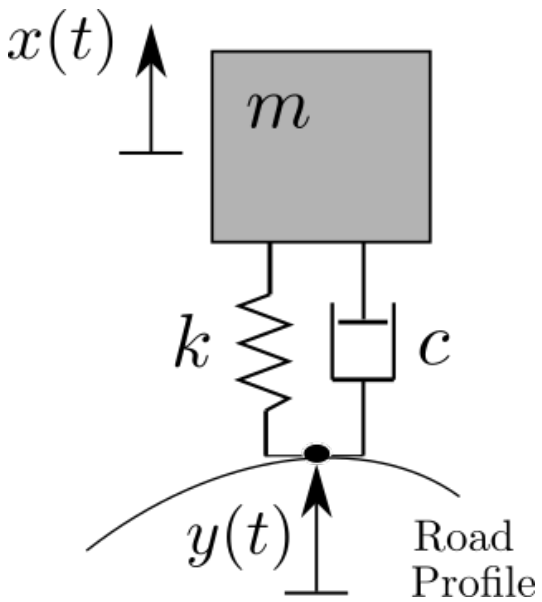
```
[Y,T]=impulse(SYS,TSPAN)
```

- Input 1: SYS - dynamic system object
- Input 2: TSPAN - array of time values for simulation

  - use IMPULSEPLOT for more options
- Output 1: Y - simulation response as array
- Output 2: T - simulation time as array

Example: 1DOF Simulation

Consider the 1DOF Quarter-Car model with displacement input from the profile of the road.



The standard EOM becomes:

$$m\ddot{x} + c(\dot{x} - \dot{y}) + k(x - y) = 0$$

The second order EOM can be decomposed into a system of first order differential equations and written as a state space model.

Example: 1DOF Simulation

The displacement input requires a *clever* substitution.

$$m\ddot{x} + c(\dot{x} - \dot{y}) + k(x - y) = 0$$

$$z_1 = x \;\; and \;\; z_2 = \dot{x} - \frac{c}{m}y$$

$$\dot{z}_2 = -\frac{c}{m}(z_2 + \frac{c}{m}y) - \frac{k}{m}z_1 + \frac{k}{m}y \;\; and \;\; \dot{z}_1 = z_2 + \frac{c}{m}y$$

Finally, write the state equation.

$$\begin{bmatrix} \dot{z}_1 \\ \dot{z}_2 \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ -\frac{k}{m} & -\frac{c}{m} \end{bmatrix} \begin{bmatrix} z_1 \\ z_2 \end{bmatrix} + \begin{bmatrix} \frac{c}{m} \\ -(\frac{c}{m})^2 + \frac{k}{m} \end{bmatrix} y(t)$$

Example: 1DOF Simulation

It is important that you keep the same substitutions when writing the output equations.

Output 1 - Position: $y_{O1} = x = z_1$

Output 2 - Velocity: $y_{O2} = \dot{x} = \dot{z}_1 = z_2 + \frac{c}{m} y$

$$\begin{bmatrix} \dot{y}_{O1} \\ \dot{z}_{O2} \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} z_1 \\ z_2 \end{bmatrix} + \begin{bmatrix} 0 \\ \frac{c}{m} \end{bmatrix} y(t)$$

References

- System Dynamics, Palm III, Third Edition -
- MATLAB-State Space handout - FIX TYPO IN HANDOUT!