7/25 ME132 Simulink!

Larry Hui

Simulink® - Introduction

- Graphical front-end to MATLAB®
- Programming environment for dynamic system modeling, control design, simulation, and analysis.
    - used in industry so you should learn how to use it!
- Hierarchical block diagrams
    - How a model is organized
    - How parts interact
- Automatic code generation for embedded deployment

For documentation, click on this link!

Simulink® - Configuration (Optional)

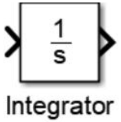
- “Simulation” → “Model Configuration Parameters”
  - “Solver” menu
    - Set “Solver Options” → “Type” to **Variable-step**
    - Set “Additional Options” → “Max Step Size” to **2e-3,** to ensure sufficient resolution of simulation data
  - “Data Input/Export” menu
    - Set “Save to workspace” → “Format” to **Array**
    - Uncheck “Additional parameters Save options” → “Limit data points to last: ” to allow Simulink® to record as many data points as necessary
- Again, this is **optional.** You do not need to do this!
- “File” → “Simulink Preferences” → “Configuration Defaults”

Simulink® - Component Blocks

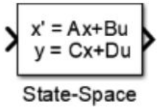
- All components are listed in the library browser! Here are some common blocks

1. Continuous

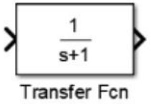
Numerical derivative of a signal



Continuous-time integration of a signal



Add a system block in state-space form



Add a system block in transfer function form

2. Discontinuous

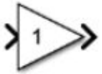
Limit input signal to specific upper and lower saturation values

Simulink® - Component Blocks

3. Math Operations

Abs

Absolute value



Gain

Constant gain. Can do matrix multiplication by changing the type to “Matrix (K*u)”

Math Function

Math functions such as exp, log, etc. Can choose from Function drop-down menu

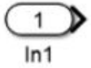
Add/substrct two or more signals. Can change signs or add/remove input nodes

Trigonometric Function

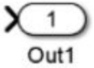
Used to place non-linear trig elements such as sin, cos, tan, and their inverse

Simulink® - Component Blocks

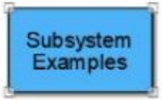
4. Ports and Subsystems



Add an input to a subsystem



Add an output to a subsystem

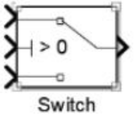


Create a user-defined subsystem with variable number of inputs and outputs.

5. Signal Routing

Split up a bus of multiple signals into its individual signals

Combine multiple signals into a single bus

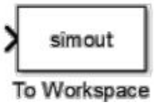


2nd input is compared against threshold and passes either 1st input or 3rd input.

Simulink® - Component Blocks

6. Sinks/Output Visualization

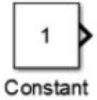
View system signals during simulation. Can plot multiple signals at once



Store signals into MATLAB workspace as specified variable name

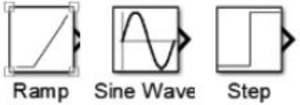
7. Sources

Clock signal returns time values used in simulation



Constant

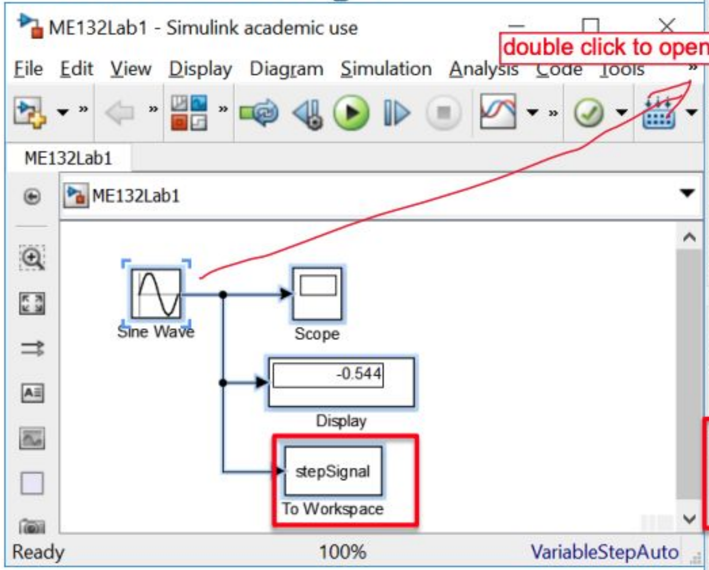
Periodic pulse signal

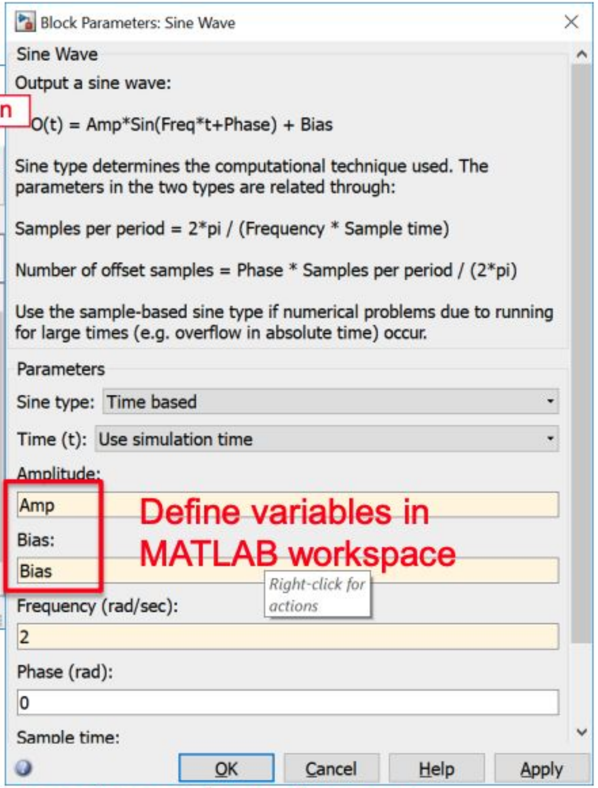


Ramp/Sine wave/Step signal with specified parameters

Simulink® - Example 1 (Output)

- **Sine to Scope**



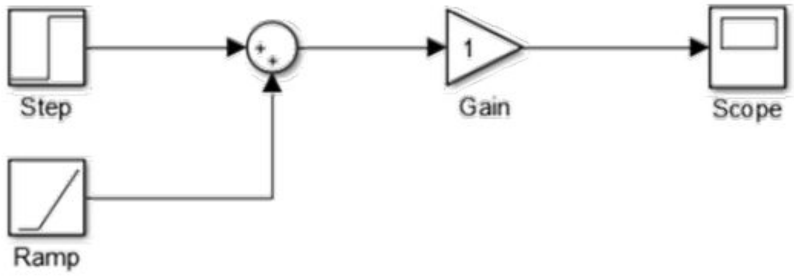



```
Amp = 1;
Bias = 1;
sim('ME132Lab1.slx')
```

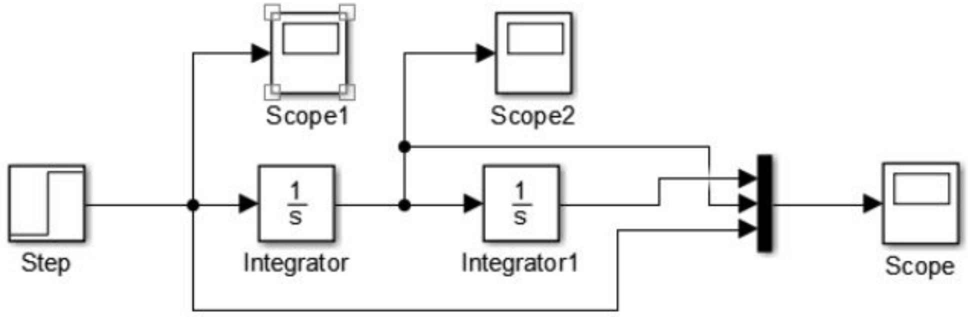
variable `stepSignal` in workspace

Simulink® - Example 2 (Block Diagrams)

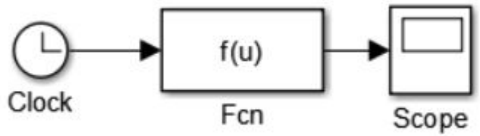
1. Math Operations: Gain, Summing Junction

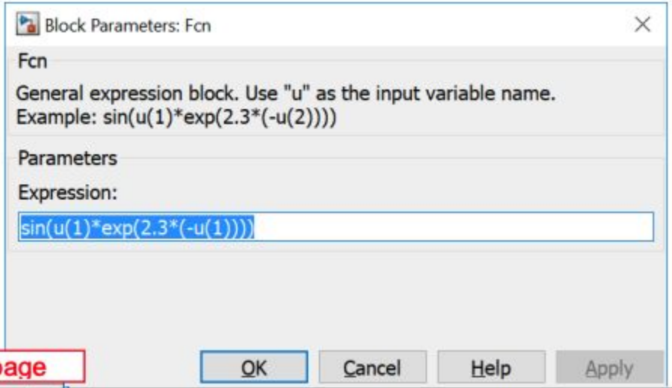


2. Continuous: Integrator



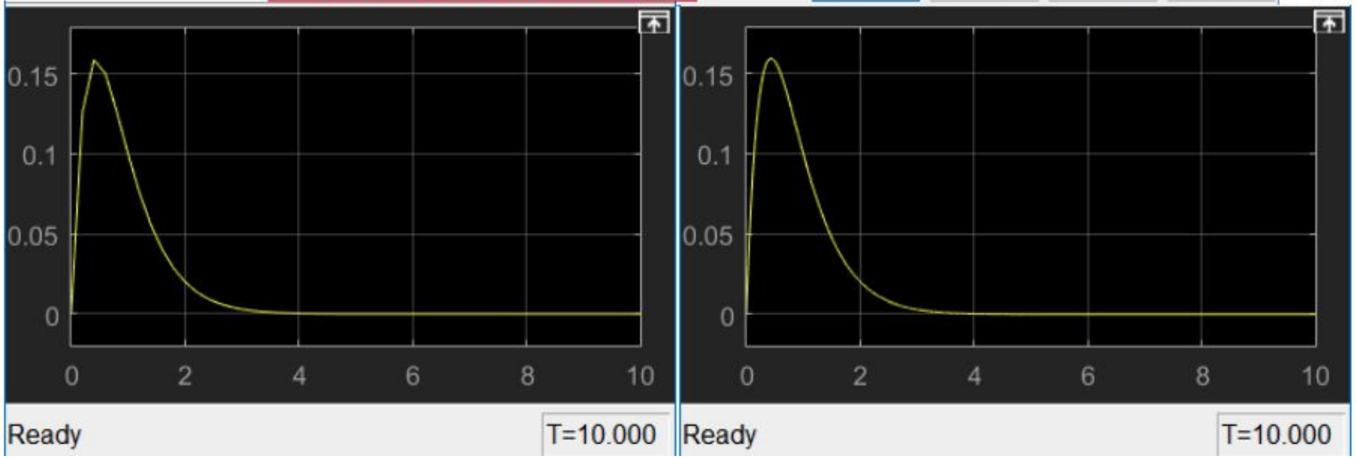
Simulink® - Example 3 (User-Defined Function)





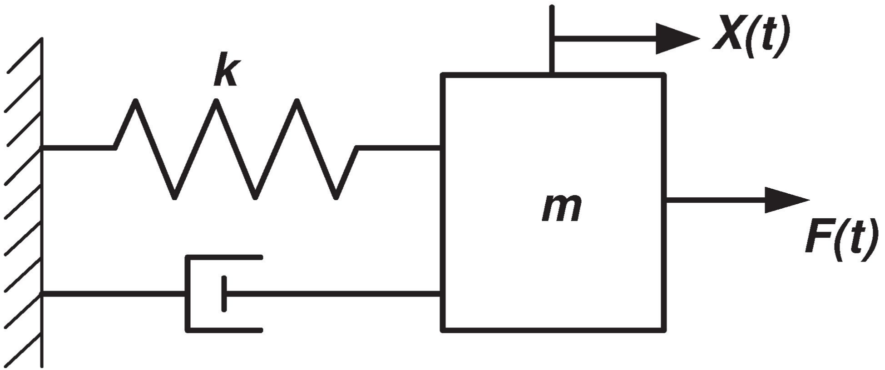
change **Max step size**

see Simulink - configuration page



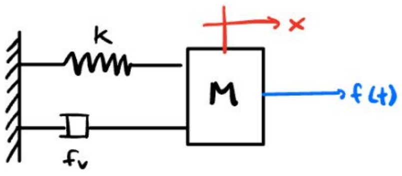
Simulink® - Walkthrough

- Since most people have not been exposed to Simulink®, I'll be going over some examples in this video.
  - Basic sine wave gain, Spring-Mass Damper w MATLAB script, PID Control
- We will model a spring-mass damper and seeing it's response
- Video will be posted under the Discussion Summer 2025 > Larry's Discussion > Simulink Supplement



Simulink® - Spring-Damper Setup

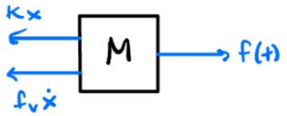
Spring - Damper - Mass System



$f(t) = u(t) = 1 \quad , \quad t \geq 0$

f_v, k, M are all unity $\Rightarrow$ all 1 !

FBD



$\Sigma F = M\ddot{x}$

$f(t) - kx - f_v\dot{x} = M\ddot{x}$

$\boxed{M\ddot{x} + f_v\dot{x} + kx = f(t)}$ EoM !

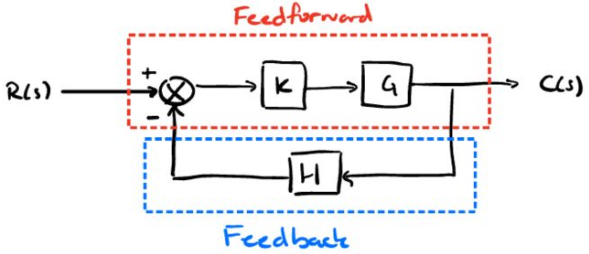
$\ddot{x} = \frac{1}{M}\left(f(t) - f_v\dot{x} - kx\right)$

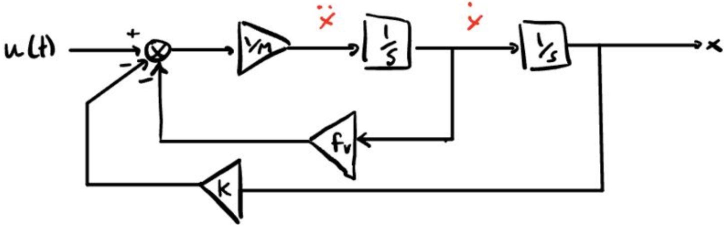
integrate twice

$\iint \ddot{x} = \frac{1}{M}\left(\iint f(t) - \iint f_v\dot{x} - \iint kx\right)$

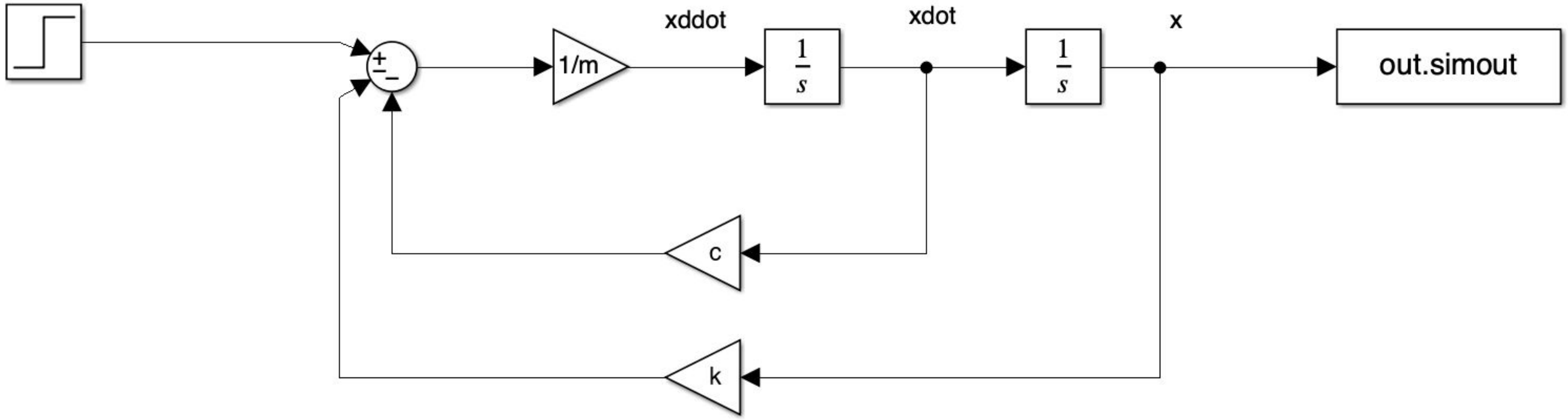
$x = \frac{1}{M}\left(\iint f(t) - kx - \int f_v\dot{x}\right)$

output; gain for feedforward path; input





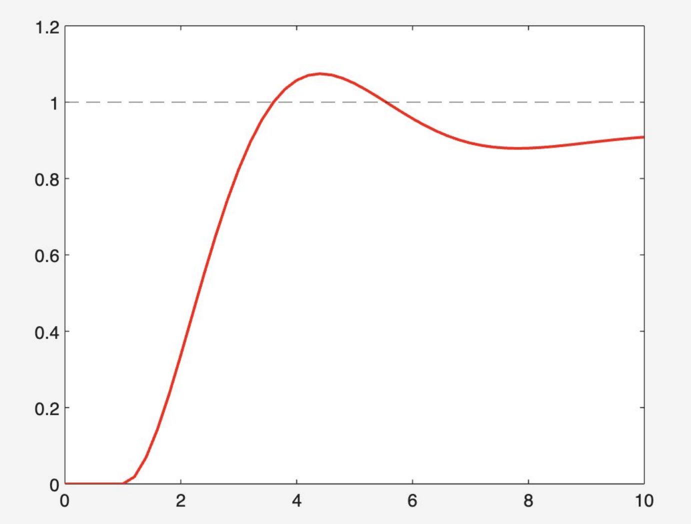
Simulink® - Spring-Damper Block Diagram



test.slx

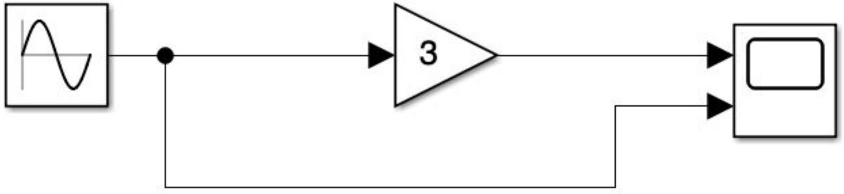
Simulink® - Spring-Damper Block Diagram Code

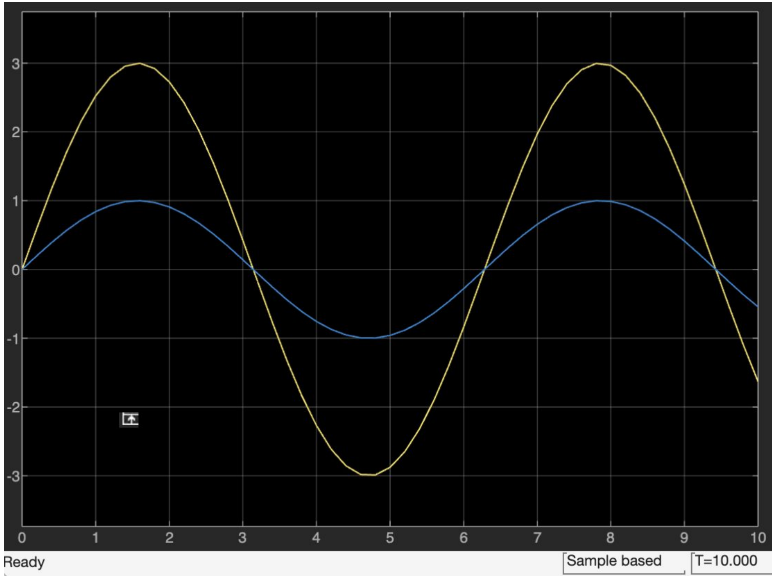
```
%% Open Loop control
clear, close all, clc

% intialize our variables
m = 1; c = 1; k = 1.1;

% configure simulation
smd_var = sim('test.slx');

% plot system response
plot(smd_var.simout.Time, smd_var.simout.Data, 'r', 'LineWidth', 1.5);
hold on
yline(1, '--')
```



disc5ex1.slx

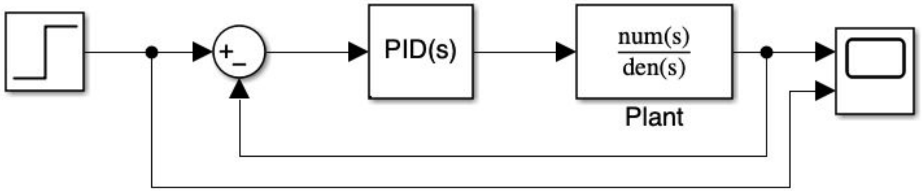
Simulink® - Sine Gain + Scope

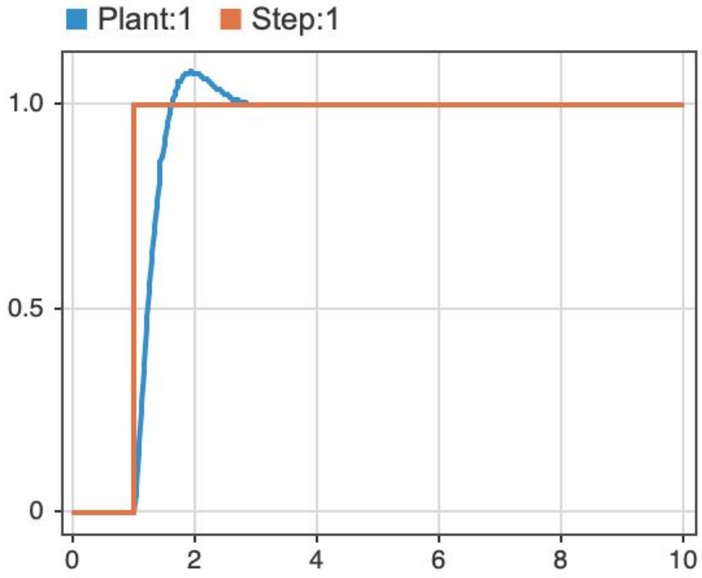




sinegain.slx

Simulink® - PID Control





example2.slx