Modeling Mechanical System using SIMULINK

Mechanical System

We will consider a toy train consisting of an engine and a car as shown in Figure 1. Assuming that the train only travels in one direction, we want to apply control to the train so that it has a smooth start-up and stop, along with a constant-speed ride. The mass of the engine and the car will be represented by m_1 and m_2, respectively. The two are held together by a spring, which has the stiffness coefficient of k. F represents the force applied by the engine, and μ represents the coefficient of rolling friction.

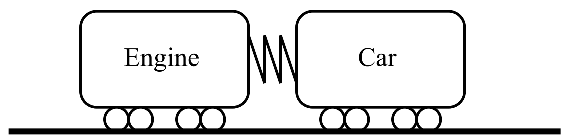


Figure 1: Train system.

Free Body Diagram and Newton's Law

The system can be represented by the free body diagram shown in Figure 2.

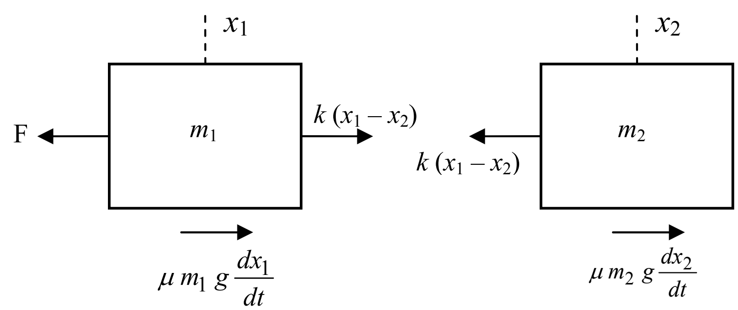


Figure 2: Free body diagram of Example given in Figure 1.

It is known from Newton's law that the sum of forces acting on a mass equals the mass times its acceleration. In this case, the forces acting on m_1 are the spring, the friction and the force applied by the engine. The forces acting on m_2 are the spring and the friction. In the vertical direction, the gravitational force is canceled by the normal force applied by the ground, so that there will be no acceleration in the vertical direction. We will begin to construct the model from the following expression

	$\sum F_1 = m_1 a$ $\sum F_2 = m_2 a$	(1)

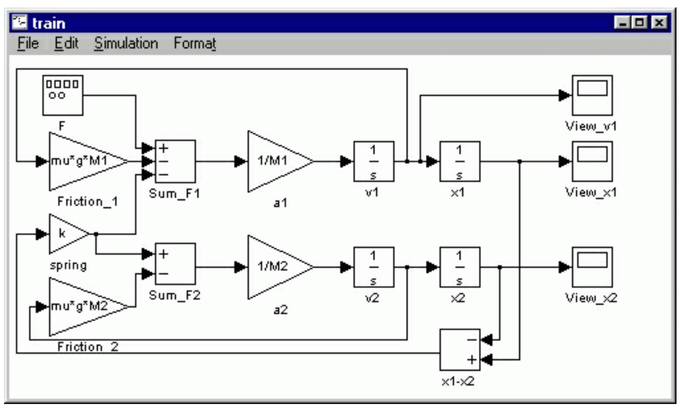
The equations of motion in the horizontal direction are as follows:

	$m_1 \frac{d^2 x_1}{dt^2} = F - k(x_1 - x_2) - \mu m_1 g \frac{dx_1}{dt}$ $m_2 \frac{d^2 x_2}{dt^2} = F - k(x_1 - x_2) - \mu m_2 g \frac{dx_2}{dt}$	

Building up the Model Using SIMULINK

1. Open a Simulink new model window (Simulink library browser).
2. *Math Operations -> Sum blocks*, drag two of them, label as **Sum_F1** and **Sum_F2**.
3. *Commonly Used Blocks -> Gain*, drag two of them, attach each one with a line to the outputs of the Sum blocks, label as **a1** and **a2**.
4. Double click the Gain blocks and change the variables to 1/M1 for a1, and 1/M2 for a2.
5. Resize the Gain blocks by dragging the small squares at the corners to stretch the block.
6. Re-align the blocks to modify the display.
7. *Commonly Used Blocks -> Integrator*, drag two for each of the accelerations, label as **v1**, **x1**, **v2** and **x2**.
8. *Sinks -> Scopes*, drag two of them, one for each integrator, label as **View_x1** and **View_x2**.
9. Add in one more force input to Sum_F1, double click the Sum block, change list of signs to be “+++”.
10. *Souces -> Signal Generator*, drag one and connect to Sum_F1, label as **F**.
11. *Commonly Used Blocks -> Gain*, drag one and connect to Sum_F1 (sign of the input of Sum_F1 should be “**-**”), label as **Friction_1**, change the variable to be “**mu*g*M1**”.
12. *Math Operations -> Sum blocks*, drag one of it at the bottom, label as **x1-x2**.

13. Right click the block, select *Format -> Flip Block*, change the signs to "**-+**".
14. Connect x2 to the negative input, and connect x1 to the positive input. (No dot means no actual crossing.)
15. *Commonly Used Blocks -> Gain*, drag one and connect to the output of the x1-x2, label as **spring**, and change the variable to be **k**.
16. Connect the output of the spring to the third input of the Sum_F1, and change the signs to "**+--**".
17. Connect the output of the spring to the first input of the Sum_F2.
18. *Commonly Used Blocks -> Gain*, drag one and connect to Sum_F2 (sign of the input of Sum_F2 should be "**-**"), label as **Friction_2**, change the variable to be "**mu*g*M2**".
19. *Sinks -> Scopes*, drag one and connect to v1, label as **View_v1**.
20. Save as "**train.mdl**".



4. Running the Model

Before running the model, we need to assign numerical values to each of the variable used in the model. For the train system, let

- M1 = 1 kg
- M2 = 0.5 kg
- k = 1 N/sec
- F= 1 N
- u = 0.002 sec/m
- g = 9.8 m/s^2

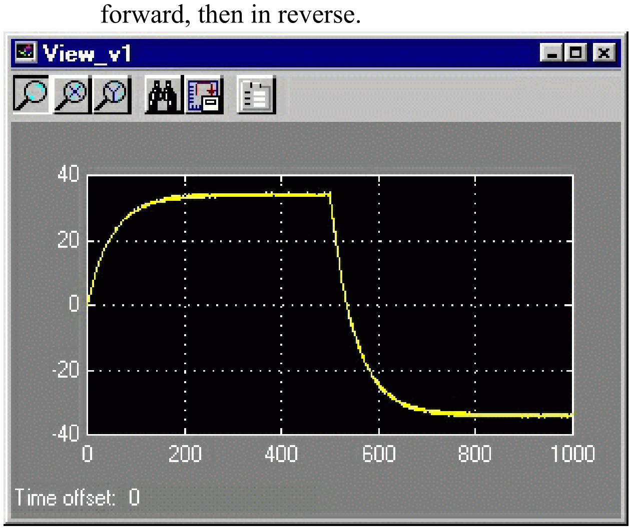
1. Create a new m-file and enter the following commands.

```
M1=1;
M2=0.5;
```

```
k=1;
F=1;
mu=0.002;
g=9.8;
```

2. Save as “variables.m”

3. Execute your m-file by entering the command **variables.m** in the main MATLAB command window. Simulink will recognize MATLAB variable for use in the model.
4. Give an appropriate input to the engine. Double-click on the function generator (F block). Select a square wave with frequency .001Hz and amplitude -1.
5. Select an appropriate simulation time. Select from menu *Simulation -> Parameters*, change the Stop Time field to 1000. (Simulate for 1000 seconds.)
6. Run the simulation and open the View_v1 scope to examine the velocity output (hit autoscale). The input was a square wave with two steps, one positive and one negative. Physically, this means the engine first went forward, then in reverse.

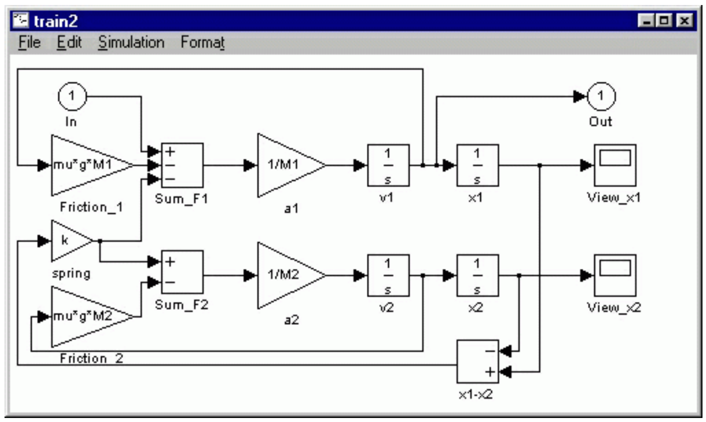


5. Obtaining MATLAB Model

We can now extract a MATLAB model (state-space or transfer function) from out SIMULINK model.

1. Delete the View_v1 scope, *Connections -> Out Block*, drag and replace.

2. Delete the F function Generator block, *Connections -> In Block*, drag and replace.
3. Save as “**train2.mdl**”.



4. Extract the model into MATLAB. Enter the following command at the MATLAB command window to extract a state-space model.

```
[A,B,C,D]=linmod('train2')
```

Output of a state-space model should be seen as below:

```
A =

  -0.0196         0    1.0000   -1.0000
        0   -0.0196   -2.0000    2.0000
        0    1.0000         0         0
   1.0000         0         0         0


B =

     1
     0
     0
     0


C =
```

```
     1     0     0     0


D =

     0
```

5. Obtain a transfer function model, enter the following command at the MATLAB command prompt.

```
[num,den]=ss2tf(A,B,C,D)
```

Output of a transfer function of the train system should be seen as below:

```
num =

         0    1.0000    0.0196    2.0000    0.0000


den =

    1.0000    0.0392    3.0004    0.0588    0.0000
```