XMUT315 Control Systems Engineering

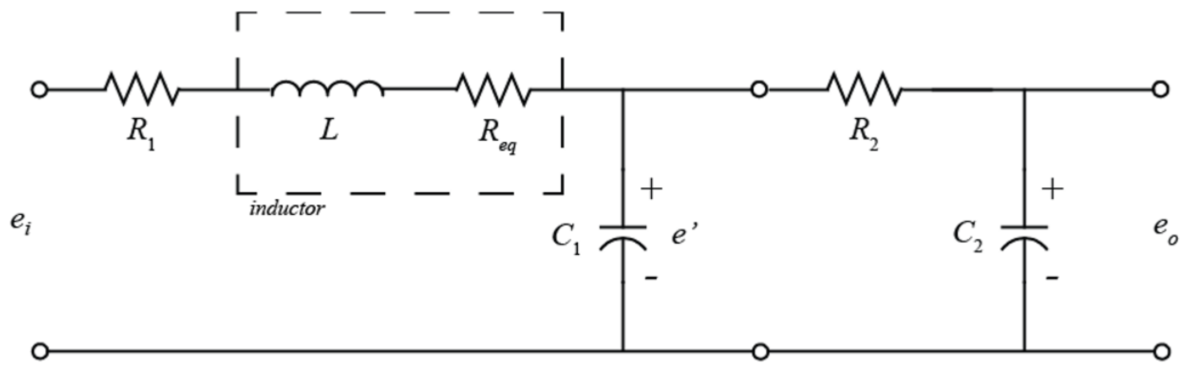
Demo 1a: Simulations in MATLAB/Simulink

A. Simulation of Control Systems with MATLAB

MATLAB is a software tool that enables modelling, analysis, and design of physical systems. Any systems, as long as their detailed descriptions and hence their equations that represents their behaviours and characteristics are available, can be modelled in MATLAB.

Experiment 1 (Simulation of Electrical Systems)

You are given the following electronic circuit that consists of series connected LRC circuit and RC circuit.

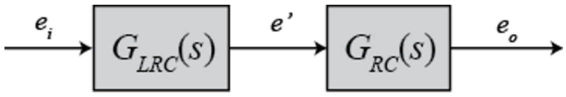


1. The following transfer function models for the LRC circuit and the RC circuit when the circuit considered separately as subsystems. Here we introduce an intermediate variable e' representing the voltage drop across the capacitor of the LRC circuit.

$$G_{LRC}(s) = \frac{E'(s)}{E_i(s)} = \frac{1}{LC_1s^2 + (R_1 + R_{eq})C_1s + 1}$$

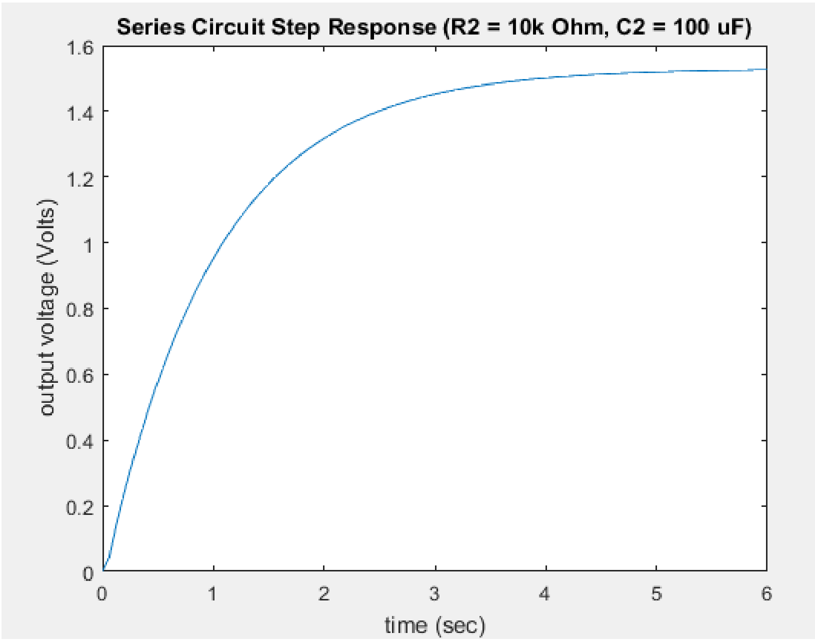
$$G_{RC}(s) = \frac{E_o(s)}{E'(s)} = \frac{1}{R_2C_2s + 1}$$

2. When modelled as transfer functions, one generally combines systems in series by multiplying their transfer functions.



3. Throughout the experiment the values of components in the LRC circuit are: $R_1 = 10\ \Omega$, $L = 1$ H, $E_{eq} = 40\ \Omega$, $C_1 = 510$ µF. For the RC circuit, we have the following component values: R_2 = 10k Ω and $C_2 = 100$ µF.

4. Use the MATLAB command step that occurs at time $t = 0$ seconds (also for step appears to occur at time equal to 1.67 seconds). The following graph shows the result of circuit simulation.



MATLAB code for the given electrical system:

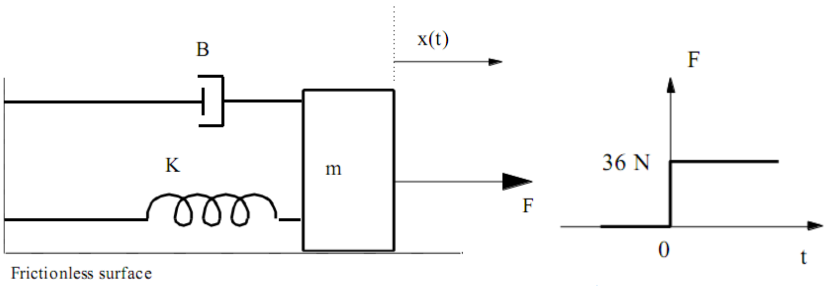
```
s = tf('s');
R1 = 10;            % resistance of resistor in LRC circuit
R2 = 10000;              % resistance of resistor in RC circuit
Req = 40;           % inductor equivalent series resistance (ESR)
L = 1;             % inductance of inductor
C1 = 510*10^-6;          % capacitance of capacitor in LRC circuit
C2 = 100*10^-6;            % capacitance of capacitor in RC circuit
ei = 1.53;               % input voltage
tstep = 1.67;           % time step occurred

G1 = 1/(C1*L*s^2 + C1*(R1+Req)*s + 1);  % LRC transfer function
G2 = 1/(C2*R2*s+1);     % RC transfer function
G = G1*G2;     % series transfer function

[y,t] = step(G*ei,6);  % model step response

plot(t,y)

xlabel('time (sec)')
ylabel('output voltage (Volts)')
title('Series Circuit Step Response (R2 = 10 kOhm, C2 = 100 uF)')
```

Experiment 2 (Simulation of Mechanical Systems)

For the mass-spring-damper system shown, determine the expression for the motion, $x(t)$ and plot it in MATLAB.

Given that K = 12 N/m, M = 4 kg, F = 36 N and the three examples of damping constant of 24.33 Ns/m, 13.8565 Ns/m, and 8 Ns/m.

Assume zero initial conditions e.g. $x(0) = 0$ and $\dot{x}(0) = 0$.



1. Assuming the equilibrium equations and apply Newton law into the mechanical system:

$$F = M\ddot{x} + Kx + B\dot{x}$$

2. Rearranging the equation to become a proper equation:

$$\ddot{x} + \frac{B}{M}\dot{x} + \frac{K}{M}x = F/M \qquad (1)$$

3. The homogenous (natural) solution:

$$\ddot{x} + \frac{B}{M}\dot{x} + \frac{K}{M}x = 0$$

4. The characteristic equation of the system by taking the Laplace transform of the above differential equation:

$$s^2X(s) + \frac{B}{M}sX(s) + \frac{K}{M} = 0$$

5. For the above equation, obtain the roots of the equation from:

$$s_{1,2} = \frac{B}{2M} \pm \frac{1}{2}\sqrt{\frac{B^2}{M^2} - \frac{4K}{M}} \qquad (2)$$

Case 1: $B = 24.33$ Ns/m (overdamped case)

6. Entering the values $B = 24.33$ Ns/m, K = 12 N/m, M = 4 kg, F = 36 N into the equation (2) as above:

$$s_1 = -5.54 \qquad \text{and} \qquad s_2 = -0.54$$

7. Substituting into equation (1) as a result:

$$x(t) = C \qquad \text{and} \qquad C = 3$$

8. Applying inverse Laplace transform:

$$x(t) = A_1 e^{-5.54t} + A_2 e^{-0.54t} + 3u(t)$$

9. With $x(0) = 0$ and $\dot{x}(0) = 0$, the two simultaneous equations are:

$$A_1 + A_2 = -3$$

$$-5.54A_1 - 0.54A_2 = 0$$

10. Solving for A_1 and A_2 yields

$$A_1 = 0.324 \qquad \text{and} \qquad A_2 = -3.324$$

11. Therefore, this is true for $t \geq 0$:

$$x(t) = 0.324e^{-5.54t} - 3.324e^{-0.54t} + 3$$

Case 2: $B = 13.8565$ Ns/m (critical damped case)

12. For value of $B = 13.8565$ Ns/m and other values as per above into the equation (2) as before.

$$s_1 = s_2 = -1.7321$$

13. As the same initial condition as per above still apply then the result of inverse Laplace transform is (for $t \geq 0$):

$$x(t) = -e^{-1.732t}(3 + 5.1963t) + 3$$

Case 3: $B = 8$ Ns/m (underdamped case)

14. For value of $B = 13.8565$ Ns/m and other values as per above into the equation (2) as before.

$$s_{1.2} = -1 \pm j\sqrt{2}$$

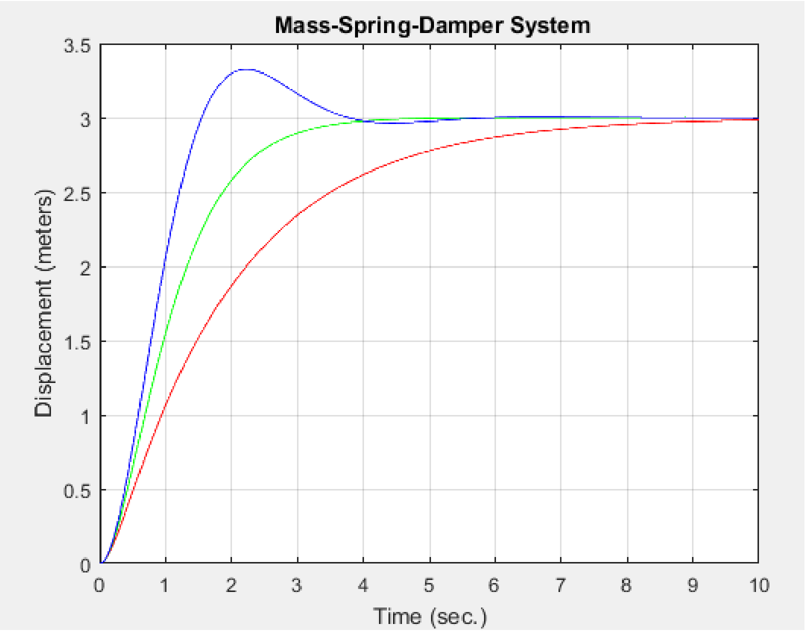
15. And the solution is

$$x(t) = e^{-t}(A_1 \cos(1.414t) + A_2 \sin(1.414t)) + 3u(t)$$

16. This leave the result of the inverse Laplace transform of:

$$x(t) = -3e^{-t}\left(\cos\sqrt{2}t + \sin\sqrt{2}t\right) + 3$$

17. Putting the three responses together is shown below along with the MATLAB code.



MATLAB code for the given mechanical system:

```
clear all

F = 36; % Newton (Applied force)
K = 12; % N/m (Spring Constant)
m = 4; % Kg (Mass)

% All cases, B is a vector:

% empty vectors for all cases of x(t)
% X1: case 1, X2: case -2, X3: case -3;

X1=[]; X2=[]; X3=[];
T=[];

for t = 0:0.01:10;
      x1 = 0.3240*exp(-5.54*t)-3.3240*exp(-0.54*t)+3;
      x2=-(3+5.1963*t)*exp(-1.7321*t)+3;
      x3=-3*exp(-t)*(cos(sqrt(2)*t)+1/sqrt(2)*sin(sqrt(2)*t))+3;

      X1=[X1 x1];
      X2=[X2 x2];
      X3=[X3 x3];
      T=[T t];
end

plot(T,X1,'r',T,X2,'g',T,X3,'b')

xlabel('Time (sec.)')
ylabel('Displacement (meters)')
title('Mass-Spring-Damper System')

grid
```

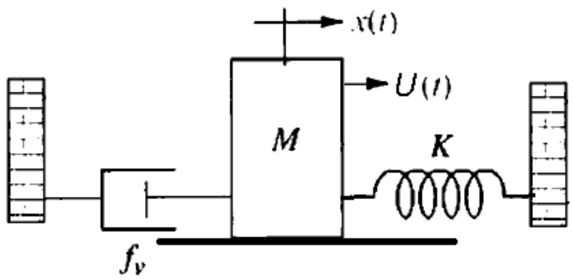
B. Simulation of Control Systems with Simulink

Simulink® is a block diagram environment for multi-domain simulation and Model-Based Design. It supports system-level design, simulation, automatic code generation, and continuous test and verification of embedded systems. Simulink provides a graphical editor, customizable block libraries, and solvers for modelling and simulating dynamic systems. It is integrated with MATLAB®, enabling you to incorporate MATLAB algorithms into models and export simulation results to MATLAB for further analysis.

Experiment 1 (Simulation of Mechanical System with Differential Equation in Time Domain)

This experiment is a basic guide to show some of the functions Simulink offers. You are encouraged to have play around with the model until you feel comfortable with adding blocks, searching the library, creating a subsystem with a mask and running a model to obtain an output.

This experiment will create a model of a damper-spring mechanical system below in the time domain in Simulink.



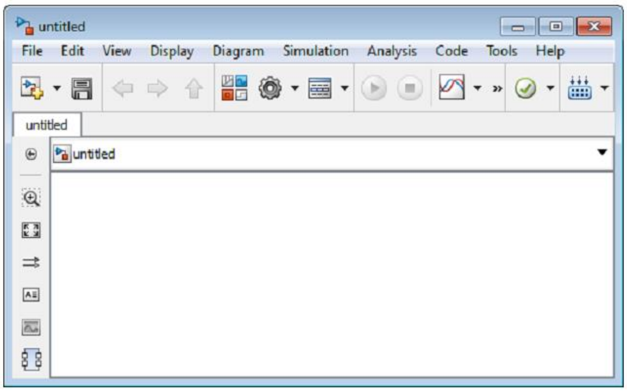
Providing that spring constant K = 2 N/m (force in the spring, $f(t) = Kx(t)$), damper constant f_v = 1 Ns/m (force in the damper, $f(t) = f_v\dot{x}(t)$), and assuming frictionless floor, the above system is represented as the following differential equation:

$$\dot{x}(t) = 2x(t) + u(t)$$

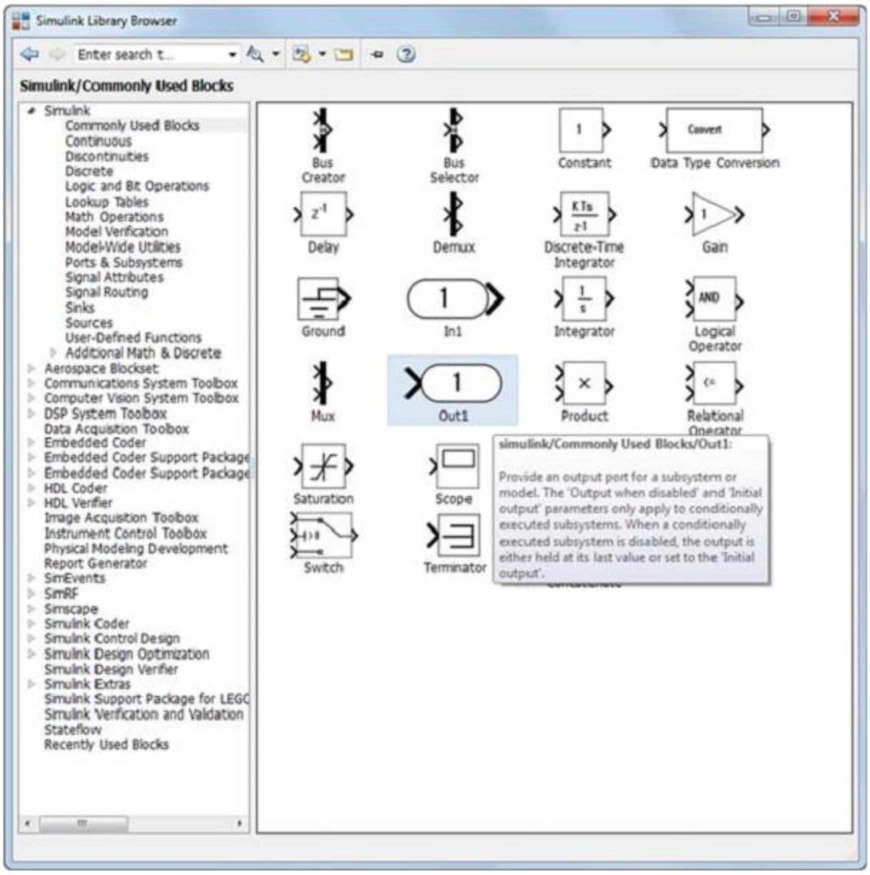
Where $u(t)$ is a square wave input with amplitude of 1 N and a frequency of 1 rad/sec

1. Open MATLAB and select Simulink from the home tab of the Toolstrip or alternatively you can type “simulink” into the command window and hit enter. This will open the “Simulink start page”.

2. In the “Simulink start page” select “blank model” and save the blank model with an appropriate name.



3. In the Toolstrip of the Simulink model there is a shortcut to “Simulink library browser” where you will find all the blocks used to create a model. The blocks are organised into sub- libraries based on the block’s functionality and/or affiliations. For this introduction, everything you need is in the “Simulink – commonly used blocks”, “Simulink - sources” and “Simulink – signal routing” sub-libraries.

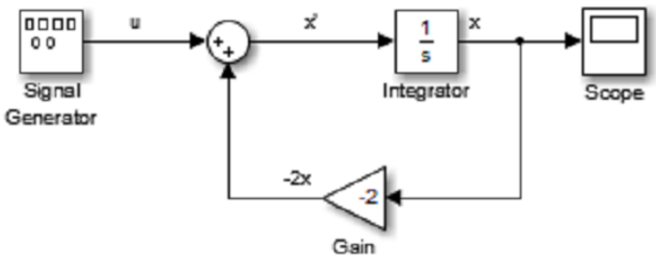


4. In the Library Browser, select an Integrator block e.g. the Integrator block integrates its input $\dot{x}(t)$ to produce $x(t)$ and other blocks needed in this model i.e. a Gain block and a Sum block.

5. Gather the blocks together.

6. To rotate the Gain block use Ctrl R (r for rotate) or to flip it horizontally, use Ctrl I (i for invert).

7. Link all of the blocks with each other as shown below.



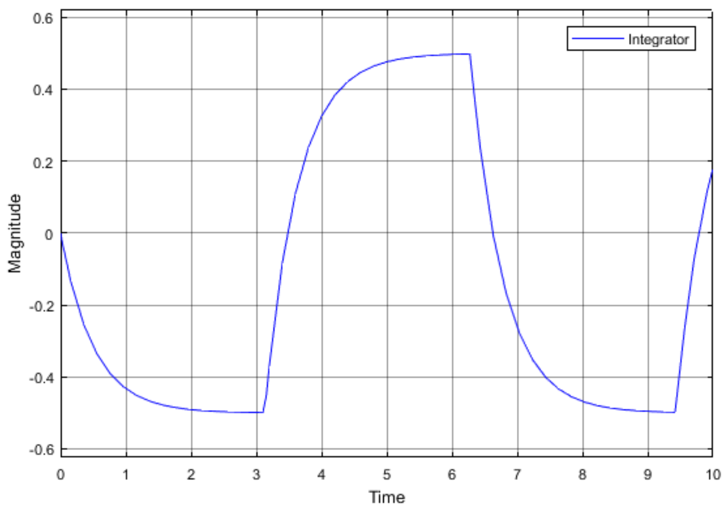
An important concept in this model is the loop that includes the Sum block, the Integrator block, and the Gain block. In this equation, $x(t)$ is the output of the Integrator block. It is also the input to the blocks that compute $\dot{x}(t)$, on which it is based. This relationship is implemented using a loop.

8. Define the gain of the system in the Gain block by double clicking it and assign value of -2.

9. To generate a square wave, use a Signal Generator block, and select the Square Wave form and change the default units to radians/sec.

10. Simulate the model by clicking the "run" button (green arrow at the top).

11. View the output of the model by double clicking on the Scope block. The Scope displays $x(t)$ at each time step. For a simulation lasting 10 seconds, the output shows as follows:



Experiment 2 (Simulation of Mechanical System with Transfer Function)

This experiment creates a model the differential equation in the frequency domain through use of transfer function. For the same system as described in the first experiment.

$$\dot{x}(t) = 2x(t) + u(t)$$

The equation you modelled previously in the first experiment can also be expressed as a transfer function. The model uses the Transfer Fcn block, which accepts u as input and outputs $x(t)$. So, the block implements $X(s)/U(s)$. If you substitute $sX(s)$ for $\dot{x}(t)$in the above equation, you get:

$$sX(s) = -2X(s) + U(s)$$

Solving for $X(s)$ this gives:

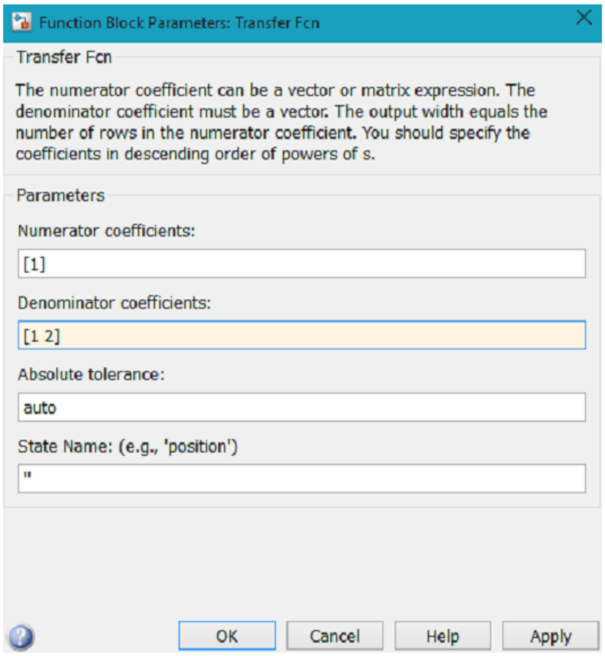
$$X(s) = \frac{U(s)}{s+2} \qquad \text{or} \qquad \frac{X(s)}{U(s)} = \frac{1}{s+2}$$

1. In the Library Browser icon, select signal generator, scope and Transfer Fcn blocks.

2. Link the blocks with each other.

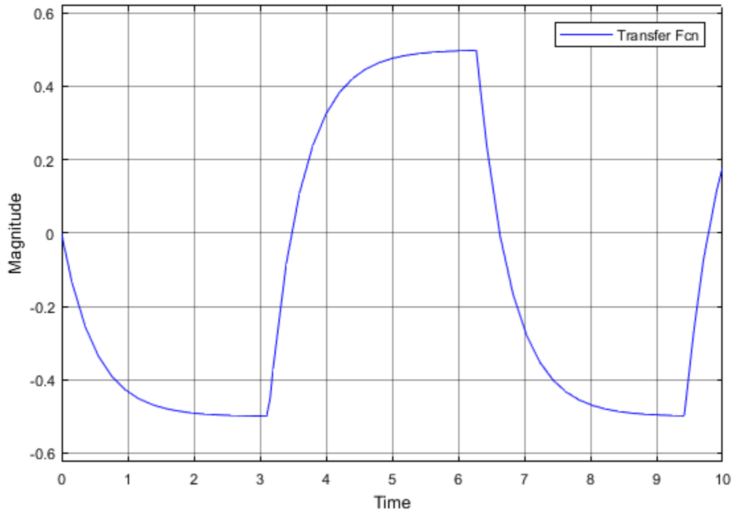
3. The block diagram of the system will look like given below.

4. The Transfer Fcn block uses parameters to specify the numerator and denominator coefficients. In this case, the numerator is 1 and the denominator is $s + 2$. Specify both terms as vectors of coefficients of successively decreasing powers of s. Double click on Transfer Fcn block, in this case assign its numerator to be [1] (or just 1) and the denominator is [1 2].



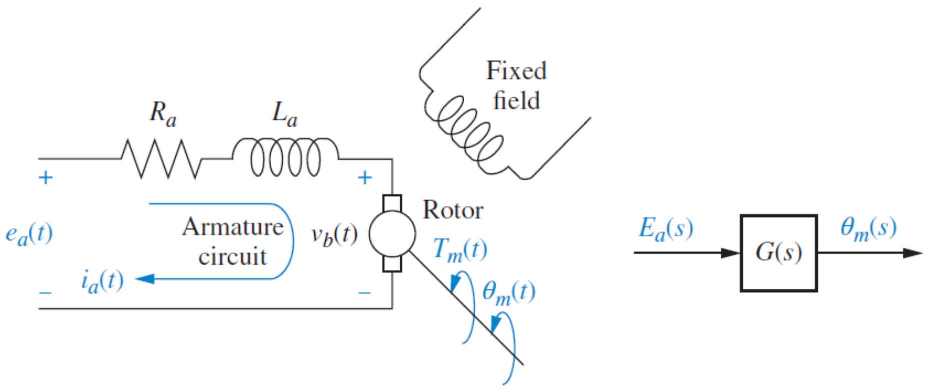
5. Run the model simulation by clicking the “Run” button (green arrow at the top).

6. Double click on the scope and the output is shown as given below (e.g. a graph of displacement vs. time).



Experiment 3 (Simulation of Electromechanical System)

This experiment will create a model of electromechanical system e.g. a brushless DC motor and its subsystems as given below. The shown motor drives a mechanical load connected to a damper system.



Note: these are the parameters of the motor: armature inductance (L_a) = 1 H, armature resistance (R_a) = 1 Ohm, torque constant (K_t) = 1 N-m/A, motor inertia (J_m) = 1, damping coefficient of motor (D_m) = 1, and back EMF constant (K_b) = 1 V-s/rad.

The equations for modelling the electric motor as shown above are as listed below:

Electrical subsystem:

$$R_s I_s(t) + L_a I_a(t) + V_b(t) = E_a(t)$$

Mechanical subsystem:

$$T_m(t) = J_m \frac{d^2\theta_m(t)}{dt^2} + D_m \frac{d\theta_m(t)}{dt}$$

Back EMF coupling:

$$V_b = K_b \frac{d\theta_m(t)}{dt}$$

Torque coupling:

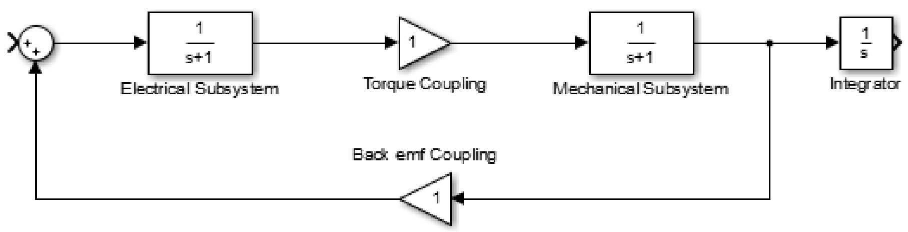
$$T_m(t) = K_t I_a(\mathrm{t})$$

1. We will be adding components to the model from the Simulink system library browser (mainly the “Commonly Used Blocks”, “Math Operations”, “Continuous”, “Sink” and “Source”).

2. Click on the Library Browser icon. We will begin by adding a “Sum” block, 2 “Transfer Fcn” blocks, 2 “Gain” blocks and an “Integrator” block.

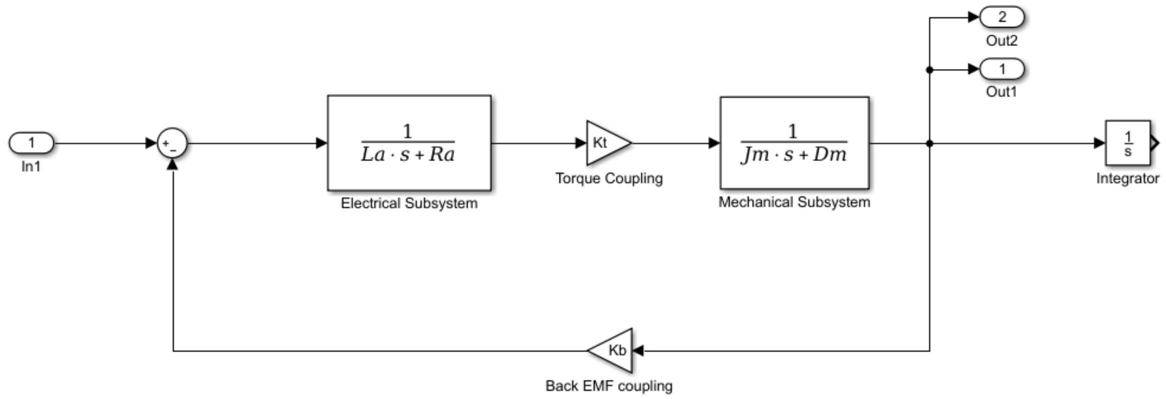
3. Arrange and name these blocks as shown in the figure below. To rotate a block use Ctrl R (r for rotate) or to flip it horizontally, use Ctrl I (i for invert).

4. Link the blocks with each other. To link, just click and drag the arrow from one point to another.
5. To add departure point i.e. from the output of mechanical subsystem to the back EMF coupling gain block, click from the block to the line (it would not work if the other way around).



6. To edit the variables in a block, double click on it. So we can easily change the variables of the system, we will add placeholder names rather than values. The variables we will use are L_a, R_a, K_t, J_m, D_m, and K_b. The figure below shows how these variables arranged.

7. To enable a negative feedback in the Sum block, double click it and change its default values from ++ to +-.



8. Now we will create a “subsystem” for the block diagram you have created. Subsystems help to keep the model tidy and allow a large number of variables edited at once. Select all the blocks in your diagram.

9. Right click and select “Create Subsystem from Selection”. Now you will have one “Subsystem” block.

10. By double clicking on the block, you can enter the subsystem containing your block diagram.

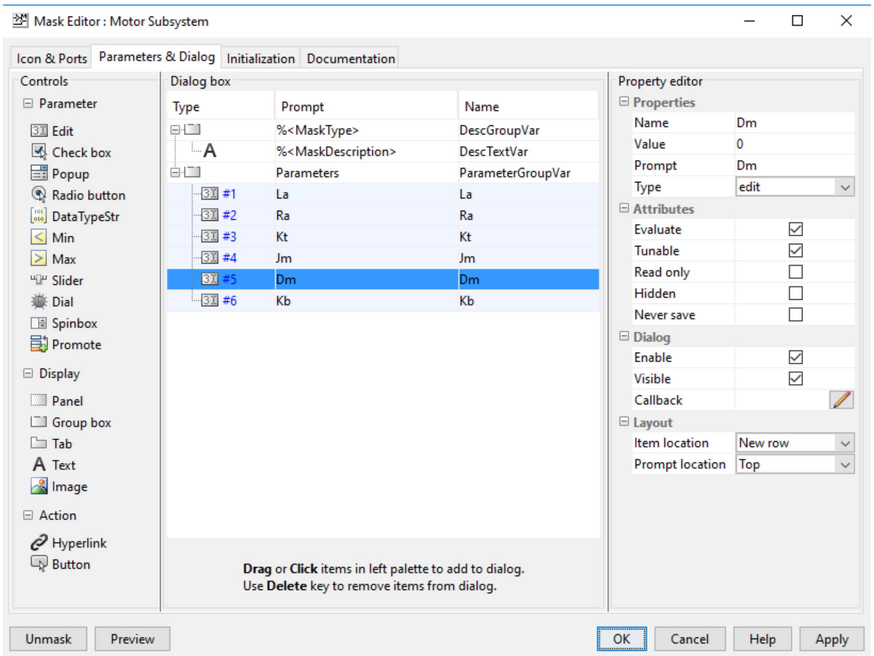
11. You can traverse tabs with the selector at the top of the working window. Name your new subsystem “Motor Subsystem”.

12. Now we will create a “mask” for our motor subsystem allowing us to store values in the variables we have inserted. Right click on the motor subsystem go to the “Mask” drop down and click “Create Mask”.

13. A new window will pop up, and we want to work in the “Parameters & Dialog” tab from the top.

14. To add a parameter, select the “edit” button on the left hand side of the screen (you will need one for each variable).

15. You can make the “prompt” something more meaningful than L_a, R_a, etc., if you like, but the write the “name” as exactly as it is in the subsystem’s blocks. The figure below shows the mask editor.



16. You will now notice an arrow on your subsystem, this will be used to enter into the subsystem as double clicking will now bring up you parameters screen.

17. Fill in the parameters of the given motor with the typical (measured) variables e.g. L_a = 1 H, R_a = 1 Ω, K_t = 1 N-m/A, J_m = 1, D_m = 1, and K_b = 1 V-s/rad.

18. Add an Input block in the subsystem and link this with +pin of the Sum block. On the other hand, add an Output block and connect it with the output side of Integrator block.

19. As we are interested in the step response of $\omega\Omega$, we will need to add a second Output block to the motor subsystem. To do so, enter the subsystem, hold Ctrl, and click on the existing Output block and drag it. This should create a second output, which you can attach to Ω. You can also give these output blocks meaningful names if you wish.

20. Now attach a "scope" block to the Ω output.

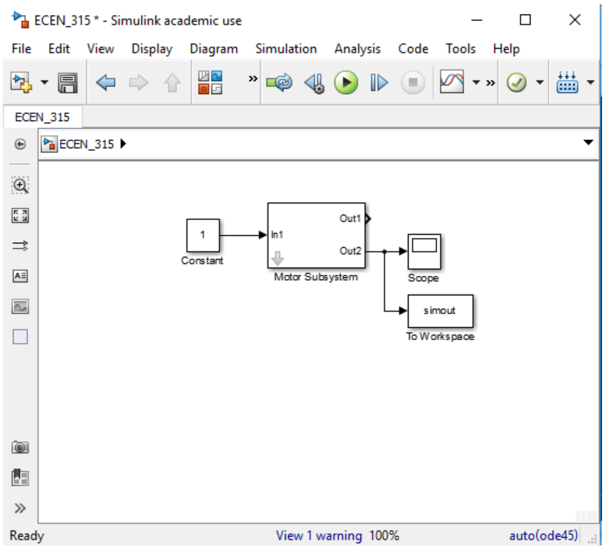
21. Double click on the scope to open it, you can use the "cog" to get "Configuration Properties" where you can change the number of inputs to the scope and give it a title.

22. Add a constant or step block to the input of the motor subsystem, change the simulation time from 10 to 2 seconds.

23. Click "Run" button (green arrow at the top) to simulate the model.

24. You can also use the "To workspace" block to add the results to the workspace for use in a script.

25. Your simulation should look like the figure below.



26. You can also perform linear analysis on your subsystem. Right click on the line between your constant block and the motor subsystem.

27. Go to “Linear analysis points” and select “Open- loop input”.

28. On the line between motor subsystem and the scope.

29. Again, go to “Linear analysis points” and select “Open-loop output”.

30. Now from the top menu, go to “Analysis”, “Control design” and “Linear analysis”.

31. From here there are a number of plots you can create including, Bode, pole/zero etc. The diagram below shows a graph of angular speed vs. time.