Out: January 21, 2019, 8:00
Deadline: February 3, 2019, 20:00

NTNU – Trondheim
Norwegian University of
Science and Technology

Assignment 2

TTK4130 Modeling and Simulation

Problem 1 (Network modelling of motor with two elastic loads, Simulink vs. Modelica. 30 %)

NB: This is a computer exercise, and can therefore be solved in groups of 2 students. If you do so, please write down the name of your group partner in your answer.

In this problem, we will attempt to use a "network modeling" approach in Simulink, that is, try to use physically motivated model interfaces, even though Simulink has no built-in mechanisms to support this[1]. The system we will model is a rotary motor with two elastic loads, that is, a mechanical system which is natural to divide into three parts.

A rotary motor has some device for setting up a motor torque T_m on a rotary shaft that rotates with angular velocity ω_m. The equation of motion for the shaft is

$$J_m \dot{\omega}_m = T_m - T_L, \tag{1}$$

where T_L is the load torque acting on the shaft. Assume the inertia is $J_m = 1\text{kg} \cdot \text{m}^2$.

(a) Implement the motor (equation 1) in Simulink. As illustrated in figure 1, T_m and T_L should be the inputs and ω_m should be the output of the Simulink sub-system. Choose yourself if you want to hardcode the parameter J_m, or if you want to make it a mask parameter.

Add a figure that shows the implemented block diagram to your answer.

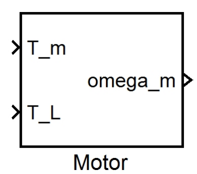

Figure 1: Simple motor implemented in Simulink: The mask.

From an energy-flow (network) point of view it is the power delivered to the motor that makes the motor run. This power is $P = T_m \omega_m$, and from this perspective natural inputs are T_m and ω_m. Similarly, the power delivered from the motor to a load is $P = T_L \omega_m$, which means natural outputs from the motor model is T_L and ω_m. However, in block-oriented (signal-flow oriented) tools like Simulink which has *unilateral interconnections*, the choice of inputs and outputs must be based on the way the model is solved/implemented computationally.

We will extend the model with a number of elastic loads, each with the following model (see Section 1.4.4 in the book):

$$J_i \dot{\omega}_i = T_{i-1} - T_i \tag{2a}$$

$$\dot{\theta}_e = \omega_{i-1} - \omega_i \tag{2b}$$

$$T_{i-1} = D_i \left(\omega_{i-1} - \omega_i\right) + K_i \theta_e \tag{2c}$$

where θ_e is the difference in rotor angles between the driving rotor and the elastic load rotor, T_{i-1} and ω_{i-1} are the torque and rotational speed on the driving rotor, and T_i and ω_i are the torque and rotational speed on the elastic load rotor.

(b) Based on the equations in 2, what are the natural signal-flow (computational) inputs and outputs, and why? What are the natural energy-flow inputs and outputs?

Hint: See next question.

[1] Simulink has an extension, Simscape, which has such features, but we will not be using Simscape in this problem.

(c) Implement a generic elastic load as a Simulink sub-system, as shown in Figure 3. Either hardcode $J_i = 1\text{kg}\cdot\text{m}^2$, $K_i = 0.5\text{kg}\cdot\text{m}^2/\text{s}^2$ and $D_i = 0.01\text{kg}\cdot\text{m}^2/\text{s}$, or use mask parameters.

Add a figure that shows the implemented block diagram to your answer.

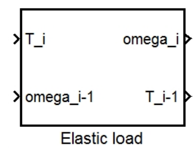

Figure 3: Elastic load implemented in Simulink: The mask.

(d) Put together the motor and two elastic loads. The last elastic load should not have an external load connected ($T_i = 0$). Let the motor torque be given by a step with size of your choosing.

Simulate and comment on the behavior of the rotational speed of the last load.

Add a figure that shows the complete system to your answer.

(e) Finally, we will look at the Bode plot from input motor torque to output rotational speed on the last load. We will let Simulink help us. The following recipe works at least for Matlab version R2018a, which is available at NTNU's programfarm:

1. Right-click the signal-flow line/arrow you want as input, choose 'Linear Analysis Points' and 'Open-loop Input'. Do correspondingly for the signal-flow line/arrow you want as output.
2. Then, in the menu, choose 'Analysis' → 'Control Design' → 'Linear Analysis...'. A new window will appear.
3. (Normally, we would now have to choose an operating point about which to linearize, but in this case the system is linear, so the operating point does not matter.)
4. Click on 'Bode' and wait until the bode plot appears.

Comment on the obtained Bode plot, and add it to your answer

We will now model this process using Dymola/Modelica. Instead of implementing the models from scratch, we will model by using predefined models from the Modelica Standard Library (MSL).

(f) Start Dymola. Choose 'File' → 'New' → 'Model'. Enter the name of the model (for example 'MotorWithElasticLoads'). Let the rest be empty, and press 'OK'. In the pane at the left hand side, press/unfold 'Modelica' to open the Modelica Standard Library. Open the library 'Mechanics' → 'Rotational'. Drag and drop 'Sources' → 'Torque', 'Components' → 'Inertia' and 'Components' → 'SpringDamper' to put together the motor with two elastic loads as shown in Figure 7. Thereafter open the library 'Blocks'. Drag and drop a 'Sources' → 'Step' to attach a step input to the torque.

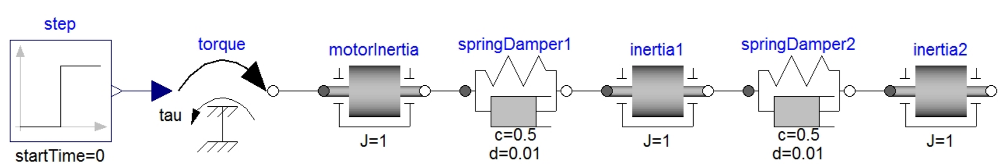

Figure 7: Dymola model of motor with two elastic loads

Simulate the model and plot the rotational speed of the last load. Compare these results with the ones obtained using Simulink/Matlab. Furthermore, comment on the amount of information contained in the graphical view of the Modelica model (Figure 7) and in the Simulink version of the overall model.

(g) Identify what variables are used by the Modelica Standard Library to connect rotating mechanical systems. Do this by going back to the "Modeling view" (press 'Modeling' tab), and open 'Mechanics' → 'Rotational' → 'Interfaces' → 'Flange_a'. Press the documentation icon (the big I) in the menu bar.

What is the difference between the variables used here and the ones used in the Simulink system?

(h) Make a Bode-plot of the model:

1. Add inputs and outputs to the model: Remove the Step-block, and connect a 'Mechanics' → 'Rotational' → 'Sensors' → 'SpeedSensor' to the last load. Add 'Blocks' → 'Interfaces' → 'RealInput' to the motor torque, and 'RealOutput' to the sensor output.
2. Go to the "Simulation view". Choose 'Simulation' → 'Linearize'.

Dymola will now export a linear(ized) model to a binary Matlab-file, called dslin.mat, in the directory where you have saved your model. Open Matlab to import this and make a Bode plot, for example by using:

```
% load output from Dymola linearize
load dslin
% ABCD is A, B, C and D matrix stacked into one matrix
% nx is number of states (dimension of the A matrix)

A = ABCD(1:nx,1:nx); B = ABCD(1:nx,nx+1:end);
C = ABCD(nx+1:end,1:nx); D = ABCD(nx+1:end,nx+1:end);

% Plot Bode response
bode(A,B,C,D)
```

Compare with the Bode plot obtained from the Simulink system.

Problem 2 (Positive real transfer functions. 42 %)

Consider the transfer functions:

1. $$H(s) = \frac{as}{1+bs}.$$
2. $$H(s) = \frac{s+a}{s^2+b^2}.$$
3. $$H(s) = \frac{s+a}{s+b}.$$
4. $$H(s) = \frac{s(s+a)}{(s+b)(s+c)}.$$
5. $$H(s) = \frac{1}{(s+a)(s+b)}.$$
6. $$H(s) = \frac{s^2+a^2}{s^2+b^2}.$$

where $a, b, c \in \mathbb{R}$ are parameters.

NB: These parameters can be zero, and factors can cancel each other out.

(a) For each transfer function, find the parameter values such that $H(s)$ is positive real.

Problem 3 (Passivity. 28 %)

(a) Let $m, d_1, d_3, k > 0$. Show that the system

$$m\ddot{x} + d_1\dot{x} + d_3\dot{x}^3 + kx = F \tag{3}$$

with input F and output $\dot{x}$ is passive.

NB: In mechanical systems, a good storage function candidate is the sum of kinetic and potential energy.

(b) Let $K_p, T_d, T_i > 0$, $\beta \geq 1$ and $\alpha \in (0, 1]$. Show that the system

$$\alpha T_d \dot{x}_1 + x_1 = (\alpha - 1)e \tag{4a}$$

$$\beta T_i \dot{x}_2 + x_2 = \frac{\beta - 1}{\alpha}(e + x_1) \tag{4b}$$

$$u = K_p \left(\frac{e + x_1}{\alpha} + x_2 \right) \tag{4c}$$

with input e and output u is passive.

(c) Consider the static system $y = f(u)$, where u is the input, y is the output and $f : \mathbb{R} \to \mathbb{R}$ is a continuous function.

Find the condition that relates the sign of u and $f(u)$, which is equivalent to the system being passive.