Out: February 27, 2018
Recommended completion: March 05, 2018

NTNU – Trondheim
Norwegian University of
Science and Technology

Exercise 8

TTK4130 Modeling and Simulation

Problem 1 (Implementation of friction models)

In this problem we will implement different *static* and *dynamic* friction models. To test the developed models, we will use a simple *testbench* as illustrated in Figure 1. The testbench is based on a model based on Newton's law of a box of mass $m = 1$kg sliding on a table due to application of an external force F_a (assumed to be a ramp function). A friction force F_f opposes motion. We will use the parameters

$F_c = 1$	Coulomb friction
$F_s = 1.2$	Stiction (static friction)
$F_v = 0.05$	Viscous friction
$v_s = 0.1$	Characteristic Stribeck velocity

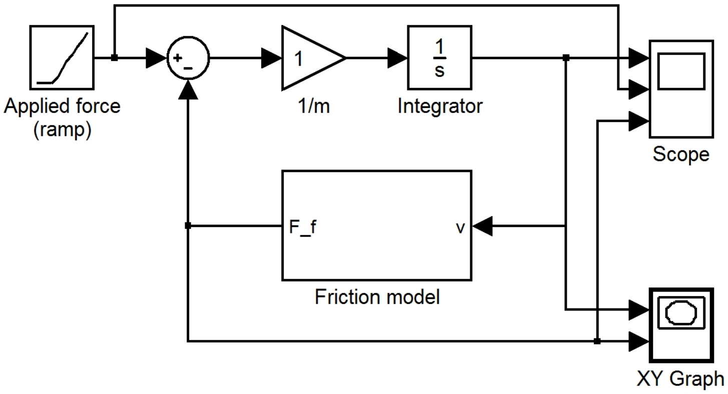


Figure 1: Testbench for friction models

(a) We start with static friction models, more specifically *Coulomb's model*, given by

$$F_f = F_c \operatorname{sign}(v), \quad v \neq 0 \tag{1}$$

where v is velocity. Implement this model in Simulink using a sign-block (do not use the built-in Coulomb friction block). Simulate the model over 10s, with ramp slope of 0.5 and 2.0. Use first a variable step solver. Explain what happens (hint: does the sign block include zero-crossing detection?). Choose the fixed-step Euler solver with sample time 0.01 instead – how does the model simulate now, for both values of the ramp?

The above (Coulomb) model has the disadvantage that it is not defined at $v = 0$. The Karnopp friction model (for Coulomb friction) remedies this by defining

$$F_f = \begin{cases} \operatorname{sat}(F_a, F_c), & v = 0 \\ F_c \operatorname{sign}(v), & v \neq 0 \end{cases}$$

where the sat-function is as defined in the book (the Simulink saturation function can be used for this saturation function by "hard-coding" the upper and lower limit at the second argument (F_c)). For this model to work properly, we must either use variable-step methods with event-detection to detect exactly when $v = 0$, or we have to use some kind of dead-zone around zero-velocity to treat the velocity as zero when it is small. Here, we will implement both of these approaches.

(b) Implement Karnopp's friction model in the setup in Figure 3, that is, fill in the two If Action Subsystems. Note that the if-block (by default) generate events when the if-clause changes. The Merge-block can be found under Signal Routing, and the If Action Subsystems under Ports & Subsystems.

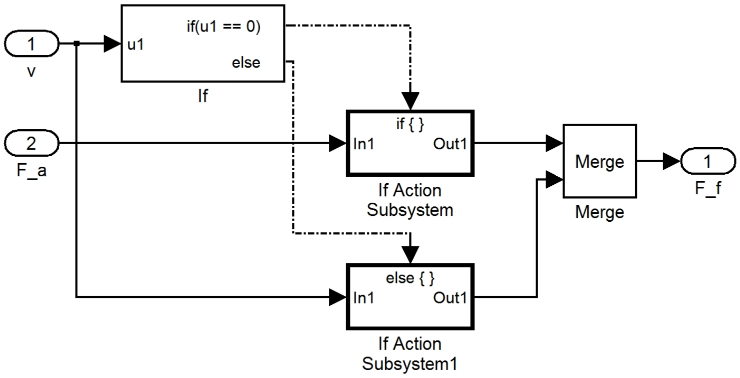


Figure 3: Setup for implementing Karnopp's friction model with event detection

Simulate using a variable step method (for example ode45) and comment. How does event-detection help? Note: It might be necessary to turn off event-detection for the sign-function inside the If Action Subsystem (why?). On the other hand, event-detection for the saturation function inside the other If Action Subsystem block should be turned on (why?).

(c) Implement Karnopp's model without relying on event-detection, by using dead-zone:

$$F_f = \begin{cases} \mathrm{sat}(F_a, F_c), & |v| \le \delta \\ F_c \,\mathrm{sign}(v), & |v| > \delta \end{cases}$$

Choose $\delta = 0.5$. Simulate using a fixed-step solver, and comment. Test also using an initial velocity of $v(0) = -1$.

(d) Extend the model in (b) with sticking, Stribeck-effect (see 5.2.6 in book) and linear viscous friction,

$$F_f = \begin{cases} \mathrm{sat}(F_a, F_s), & v = 0 \\ \left[F_c + (F_s - F_c)e^{-(v/v_s)^2}\right] \mathrm{sign}(v) + F_v v, & v \neq 0 \end{cases}$$

(note slight error in book). Compare with Coulomb friction. Use ramp slope 0.5.

(e) Finally, we shall implement the LuGre dynamic friction model,

$$\dot{z} = (v - \sigma_0 \frac{|v|}{g(v)} z),$$

$$F_f = \sigma_0 z + \sigma_1 \dot{z} + \sigma_2 v,$$

where z may represent a small displacement in the stick-zone and σ_0 "spring-stiffness" of asperities. Set $\sigma_0 = 500$, $\sigma_1 = 0$ and $\sigma_2 = F_v$, and and implement $g(v)$ to get the same steady-state solution as in (d). Simulate and compare with solution in (d). Play around with the parameters σ_0 and σ_1, and comment.

Problem 2 (Transmission line)

A lossless transmission line is described by

$$\frac{\partial u}{\partial t} = -cz_0 \frac{\partial y}{\partial x} \tag{2a}$$

$$\frac{\partial y}{\partial t} = -\frac{c}{z_0}\frac{\partial u}{\partial x} \tag{2b}$$

where c is wave velocity (speed of sound) and z_0 is line (characteristic) impedance. The wave variables are

$$a = u + z_0 y,$$
$$b = u - z_0 y.$$

(a) We first assume that the equations describe a lossless hydraulic transmission line filled with water. What physical variables are u and y in this case?

The transmission line is a pipe with circular cross section with radius $R = 10\,\text{cm}$. Water has speed of sound

$$c = \sqrt{\frac{\beta}{\rho}} = 1500\,\text{m/s} \tag{3}$$

and density $\rho = 980\,\text{kg/m}^3$. What is the bulk modulus of water, and what is (characteristic) line impedance for the transmission line? What are the units?

(b) We now assume the equations describe an electrical transmission line in the form of a high voltage line in air. The wave velocity is the speed of light $c = 3 \cdot 10^8\,\text{m/s}$ and the characteristic impedance is $z_0 = 300\,\Omega$. What are the line inductance L and capacitance C per unit length? What physical variables are u and y in this case?

Problem 3 (Passivity of transmission line)

A lossless transmission line is described by (2). We use index 1 to denote input-side and index 2 to denote output-side. That is, u_1 is input and y_2 is the output/measurement. On the output side we have a load impedance $Z_2(s)$ such that

$$U_2(s) = Z_2(s)Y_2(s). \tag{4}$$

The corresponding wave variables fulfil

$$B_2(s) = H_2(s)A_2(s) \tag{5}$$

where

$$H_2(s) = \frac{G_2(s) - 1}{G_2(s) + 1} \tag{6}$$

and

$$G_2(s) := \frac{Z_2(s)}{z_0}. \tag{7}$$

See e.g. Chapter 1.6.8 in book.

A linear system with input u and output y is passive if the transfer function $Y(s)/U(s) = G(s)$ is positive real.

(a) Show that if $G_2(s)$ is positive real, then $H_2(s)$ is bounded real (hint: see book Chapter 2.4.11).

(b) Define $H_1(s)$ by

$$B_1(s) = H_1(s)A_1(s). \tag{8}$$

Show that if $H_2(s)$ is bounded real, then $H_1(s)$ is also bounded real. Use that

$$H_1(s) = \frac{B_1(s)}{A_1(s)} = e^{-2Ts}H_2(s).$$

(c) We have the load impedance $Z_1(s)$ such that

$$Z_1(s) = \frac{Y_1(s)}{U_1(s)}. \tag{9}$$

Further, we define the transfer function

$$G_1(s) = Z_1(s)z_0. \tag{10}$$

Show that

$$G_1(s) = \frac{1 - H_1(s)}{1 + H_1(s)} \tag{11}$$

and that if $H_1(s)$ is bounded real, then $G_1(s)$ is positive real. (Hint 1: Check how Eq. 6 is found. Hint 2: $2\,\mathrm{Re}\,[G_1(s)] = G_1(s) + G_1^*(s)$).

(d) Show that if $Z_2(s)$ is positive real, then the system with input u_1 and output y_1 is passive.

Problem 4 (Coupled drives)

The purpose of the coupled drives model can be to model the process of winding some material from one spool to another, where the material might be wire, plastic strip, paper, textile yarn, or magnetic tape. The model, with some small modifications, may also represent two drives driving a conveyor belt/assembly line.

The system consists of two (electric) drives and a material belt (the material being wound). In addition to controlling the speed of the belt, it is usually necessary to control the tension in the belt. To these ends, one may use a spring mounted pulley such that the deflection of the spring indicates tension, and the pulley speed indicates belt speed. See Figure 10, which is the setup we will model in this problem.

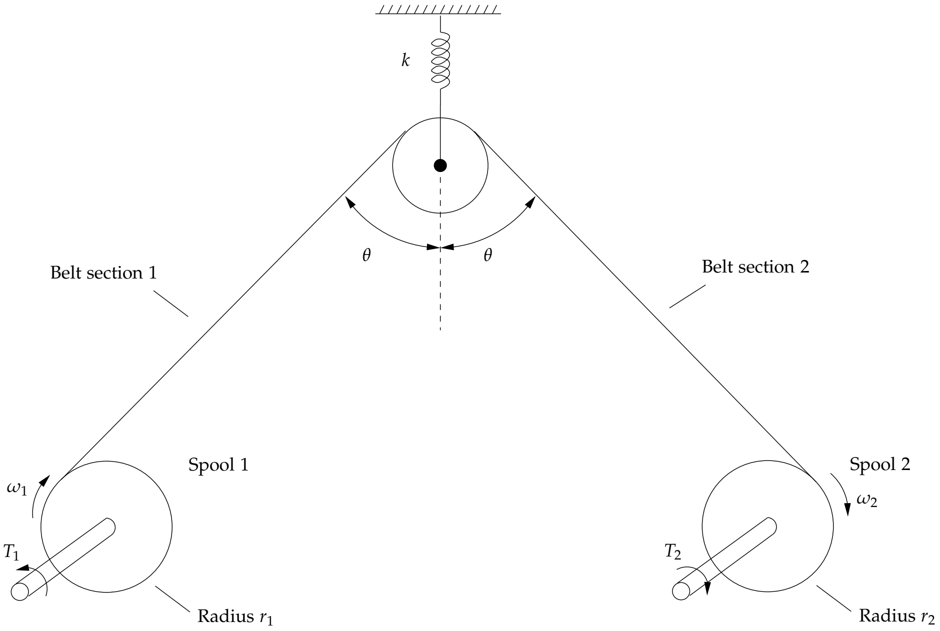
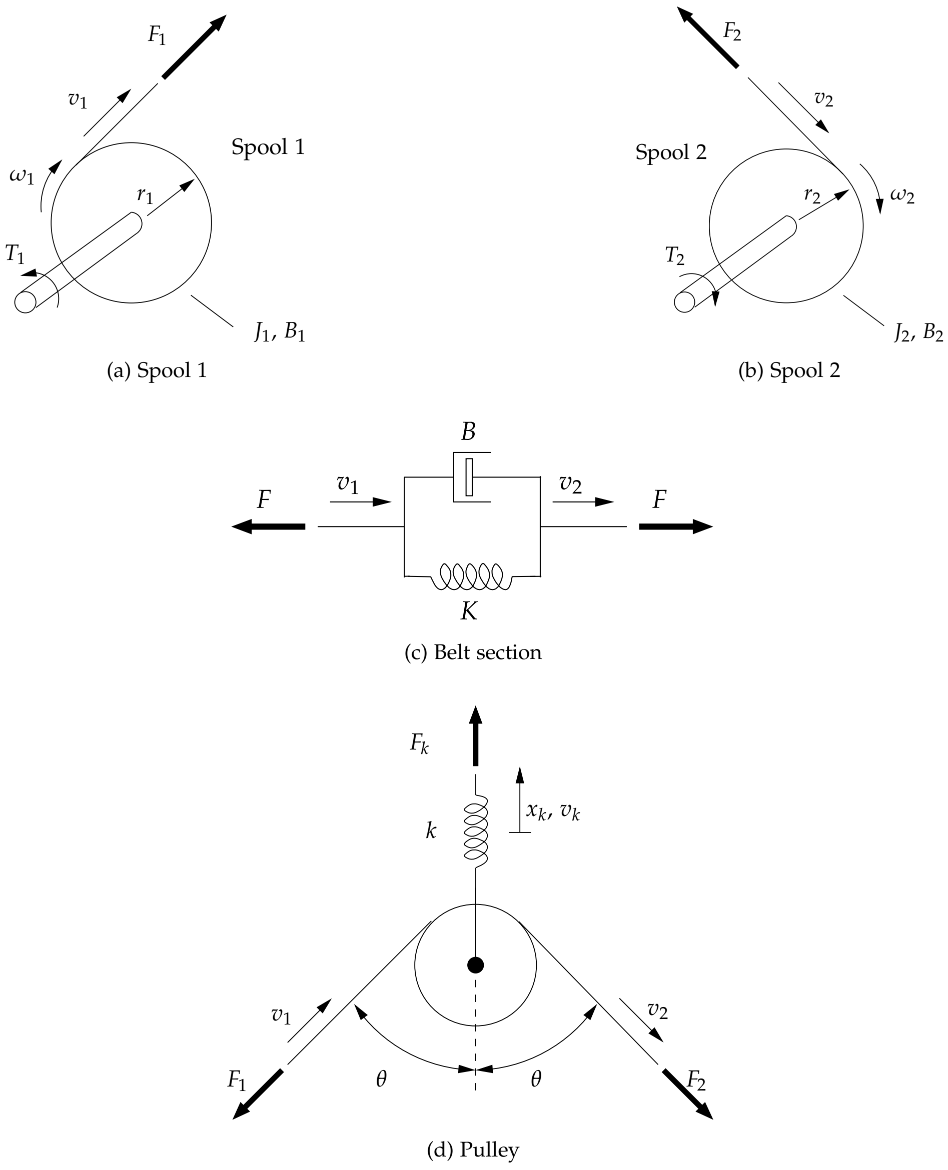


Figure 10: Coupled drives setup

We will implement the model as consisting of five parts, two spools driven by torques T_1 and T_2 (from some motors, for instance DC motors, that we will model only as a torque source), two elastic belt sections, and one "jockey pulley".

We will implement this model in Dymola. Create a package called CoupledDrives. In this package, create a model also called CoupledDrives, that will be the total model of the coupled drives system. In addition, we will create models for the drives, the belts and the pulley in the package, which we will use to put together the total model.

To help the modeling, we have drawn a "free body diagram" in Figure 11.



(a) Spool 1

(b) Spool 2

(c) Belt section

(d) Pulley

Figure 11: "Free body diagrams" illustrating forces and positive directions

(a) We start be modelling the drives. In addition to the torque input and the belt force, we assume there is "viscous friction" of the form $B_i\omega$. Let the force from the belt be denoted F_i. Write up the torque balance for Spool 1 ($J_1\dot{\omega}_1 = \ldots$) and 2 ($J_2\dot{\omega}_2 = \ldots$). Note how the driving torques has opposite positive directions, and how the belt forces work in opposite directions compared to the rotation.

We will use the Modelica Standard Library, Modelica.Mechanics.Rotational to model the drives. By switching positive direction on the torque on Spool 1, we will use the same model for both drives/spools. It might seem that we then will get a problem with the sign of the belt force on Spool 2, but we will see later in the problem how this fixes itself.

(b) Create a model called Drive in the package you created, and drag and drop components Torque, Inertia, Damper and IdealGearR2T. Relate each component to a term in the torque balance. Insert parameters $J = 1 \cdot 10^{-3}\text{kg} \cdot \text{m}^2$, $B = 0.1\text{kg} \cdot \text{m}^2/\text{s}$, $r = 0.1\text{m}$ (same for both drives[1]).

We will use this model as a sub-model, so we need inputs and outputs: Use Modelica.Blocks.Interfaces.RealInput as input, and connect to the Torque-block. Use Modelica.Mechanics.Translational.Interfaces.Flange_b as output, and connect to the IdealGearR2T. If you like, open the Icon-view of the model and draw for example a circle and rectangle to represent the drive. The drive model is illustrated in Figure 12. (A possible icon for the model can be seen in Figure 13.)

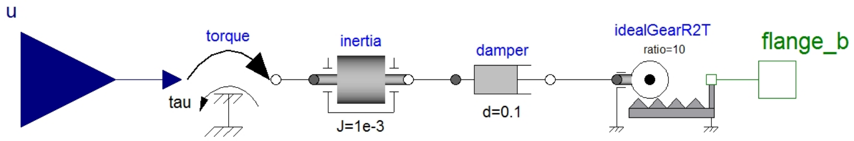


Figure 12: Dymola model of a drive/spool.

(c) We model each belt section as a massless spring + damper, see Figure 11c. Write down the equation for the force, and find a component in Modelica.Mechanics.Translational.Components that can be used for this. Let the belt stiffness be $K = 50\text{kg}/\text{s}^2$ and belt viscous friction/damping be $B = 0.5\text{kg}/\text{s}$.

(d) Lastly, we want to model the pulley, see Figure 11d. We assume the pulley is without mass. A power balance can be written

$$F_1 v_1 - F_2 v_2 = F_k v_k.$$

The spring force can be written

$$F_k = k x_k, \quad \dot{x}_k = v_k.$$

Write down a force balance in horizontal and vertical direction, and show that the pulley can be described by

$$\begin{aligned} v_1 - v_2 &= 2 v_k \cos\theta, \\ \dot{x}_k &= v_k, \\ F_k &= k x_k, \\ F_1 = F_2 &= \frac{F_k}{2\cos\theta}. \end{aligned}$$

Assume the spring stiffness is $k = 200\text{kg}/\text{s}^2$, and $\theta = 45°$. (The above equation set is not minimal – several variables can be eliminated. However, Dymola can do this for us.)

(e) To implement the pulley model, we cannot use elements from the standard library. Implement instead the model above in the following skeleton:

[1] Radius r means a ratio $1/r$ when converting to translational motion.

```
model Pulley
  extends Modelica.Mechanics.Translational.Interfaces.PartialTwoFlanges;
  import SI = Modelica.SIunits;
  SI.Velocity v1 "Velocity section belt 1";
  SI.Velocity v2 "Velocity section belt 2";
  SI.Velocity vk "Vertical velocity pulley";
  SI.Position xk "Spring deflection/vertical position pulley";
  SI.Force Fk "Spring force";
  parameter Real k "Spring constant";
  parameter Real theta "Angle";
equation
  // Implement Pulley model here
end Pulley;
```

The forces F_1 and F_2 can be taken from the connectors flange_a and flange_b, but note that the force in flange_b has opposite positive direction compared to Figure 11d. The corresponding velocities can be found by differentiating the positions in these connectors.

(f) Put together the entire model to get something like Figure 13. Note how the force on Spool 2 now has the right “sign” due to Modelica sign conventions.

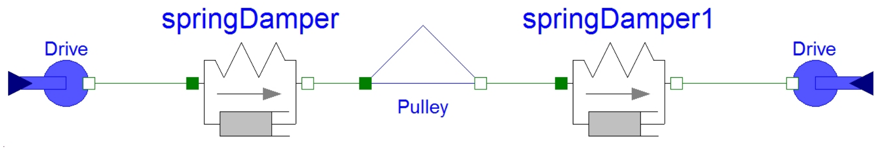


Figure 13: Dymola model of coupled drives

(g) Finally, we want to plot a Bode plot of the transfer-function from the two torques to the deflection of the pulley spring (a measurement related to belt tension).

- Add an output to the pulley model. A simple way to do this is to drag-and-drop a Modelica.Blocks.Interfaces.RealOutput into the pulley model, and add the equation `y = xk;`
- Add two Modelica.Blocks.Interfaces.Realinput to the CoupledDrives-model and connect to the torque inputs.

You should also add a sign reversal for the torque input to spool 1, cf. discussion before (b). See Figure 14. Linearize (in the simulation menu) and create a Bode-plot in Matlab, using the code

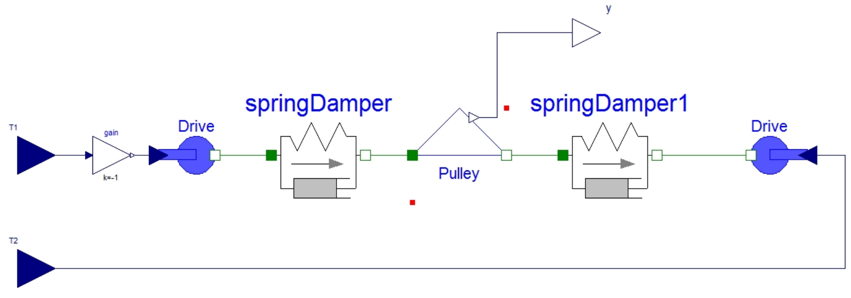


Figure 14: Dymola model of coupled drives with inputs and outputs

```
% 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);
sys = ss(A,B,C,D);

% Plot Bode response
w = logspace(-1,4,50);
bode(sys,w)
```

Comment.

(h) (Optional!) Experiment with simulations of the model, for instance by adding a step torque to the torque input of Spool 2. Try to add PI controllers for controlling belt speed and belt tension (as measured by pulley speed and spring deflection, respectively). As this is a highly coupled two-by-two system, this might be somewhat challenging[2].

[2] Adding a “decoupler” by making a input coordinate change may help.