Assignment 1: Intro Simulink

1DT059: Model-based Design of Embedded Software
Uppsala University

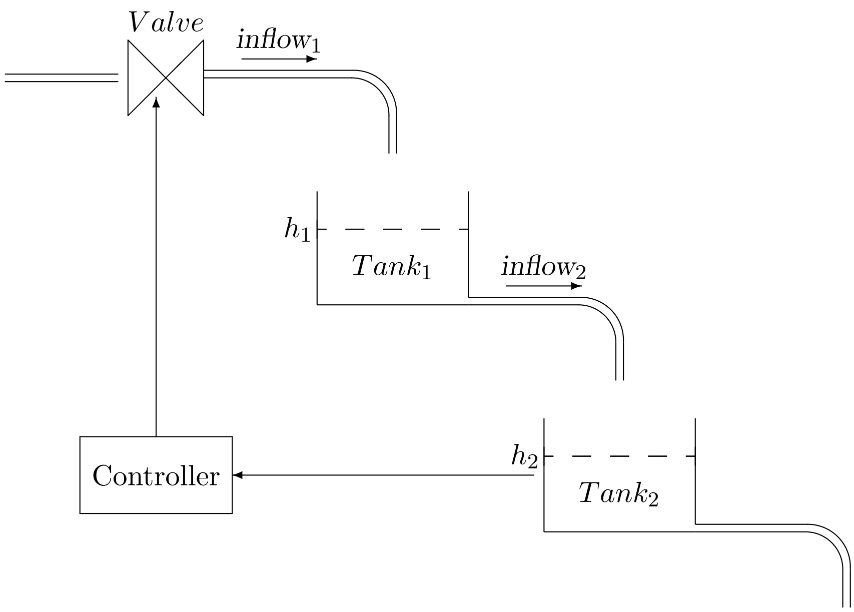
September 2, 2020

In this assignment you produce Simulink models. Each exercise should be implemented in a single .slx-file that is named after the exercise (the solution to assignment 1, exercise 2 should be in file `a1e2.slx`). Also include a file `a1.pdf` with a short high-level description of the ideas of your solutions (you do not need to include obvious things). Include your name at the top of all submitted files.

Assignments are to be solved by students individually. You can (and are encouraged to) discuss ideas and concepts with fellow students, but it is absolutely forbidden to share or copy (even parts of) solutions or lines of code. Breaking this rule will be reported accordingly.

Problem 1 Controlling Sequence of Tanks. (30p)*:*
In this problem, we consider a system of two coupled tanks, with a valve controlling the inflow to the top valve, as in the below figure. Each tank $Tank_i$ (for $i = 1, 2$) has the form of a rectangular bottom with perpendicular walls. It has a surface area which is $1m^2$. We let h_i denote the level (in m) of water in $Tank_i$. The level should of course always be nonnegative. The maximum water level in each tank is $2m$. The water level in each tank SHOULD NOT exceed $2m$. Tank $Tank_i$ has an outflow which is 0.1 times h_i (in m^3/s). The outflow from $Tank_1$ goes to $Tank_2$; the outflow from $Tank_2$ goes to the environment. $Tank_1$ has an inflow inflow_1 which is controlled by the valve $Valve$. The inflow to $Tank_2$ comes from $Tank_1$. We let h_i be the level of water in $Tank_i$ for $i = 1, 2$, and let inflow_i be the flow (in m^3/s) of water into $Tank_i$ for $i = 1, 2$.

The value of inflow_1 that goes through the $Valve$ to $Tank_1$ is controlled by a Controller, We assume that the Controller can freely control the value of inflow_1 (e.g., there are no constraints on how fast inflow_1 can change with time). The Controller can also observe the level h_2 of the second tank, through a sensor whose reading is input to Controller. Thus, the Controller must at all times provide a value for inflow_1 on the basis of past and present observations of h_2. Note that the Controller can not observe h_1.



The problems are as follows.

a) First set up equations for the system of tanks without the Controller, which describe how the levels of water evolve over time. The equations will treat $inflow_1$ as an independent input that does not depend on the other

b) Make a Simulink model for the tanks without the Controller, based on the equations in a). The model will have *inflow*$_1$ as an input.

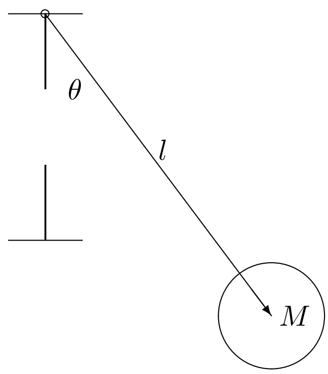
c) Now design and add a Controller to your model. The purpose of the Controller is to maintain a constant water level of around $1m$ in $Tank_2$. Design a component, block, or subsystem, respresenting the Controller in the picture, which observes h_2 (i.e., has h_2 as input)and can control $inflow_1$. The goal of the Controller is that the level of water in $Tank_2$ should reach and stay as close to $1m$ as possible (possibly after some oscillation above and below this value). You can construct the controller in any way you like; the only condition is that it can observe only h_2, and cannot directly observe h_1. Furthermore, the Controller must make sure that the level of water in $Tank_1$ or $Tank_2$ does not exceed $2m$. This can be slightly tricky, since the Controller does not "see" $Tank_1$, and since there is delay from a change to *inflow*$_1$ to its effect on h_2.

d) Simulate your system, starting from two different initial states

   1. with both tanks empty, and
   2. with both tanks full.

   Simulate until the systems seems to be reaching some steady state. Check that you avoid overflowing any of the tanks. Also check (by inspecting your simulation) that *inflow*$_1$, cannot be negative.

Problem 2 Pendulum. (30p)*:*
In this exercise, you model and simulate a simple pendulum, and use simulations to make some simple analysis. The pendulum is anchored in the ceiling. It has a mass of M at the end of a rod of length l (ignore the mass of the rod). See the picture below.



Let us set up equations that govern the pendulum dynamics. We let the state of the system be represented by the angle θ of the pendulum (where $\theta = 0$) represents the pendulum hanging vertically) and its derivative $\dot{\theta}$ (the angular velocity). To derive a model for the dynamics of the pendulum, we use Newton's second law of motion. Let us consider forces in the direction of the current movement of the mass (i.e., perpendicular to the rod). We let positive direction be in the direction of increasing θ (i.e., right-upwards in the above picture). The acceleration in this direction is $l\ddot{\theta}$ (the length of the rod times the angular acceleration). According to Newton's second law, $Ml\ddot{\theta}$ should be equal to the sum of the forces acting on the pendulum in the direction of movement. These are:

- Gravity, which in the downward direction is Mg. In the direction of movement it has magnitude $Mg\sin(\theta)$ and is directed in the direction of negative θ.
- Friction, which we just approximate as b times the velocity for some friction coefficient b. I.e., it is $-bl\dot{\theta}$ (note the sign)

Thus the sum of forces acting on the pendulum is $-Mg\sin(\theta) - bl\dot{\theta}$. As values for parameters, we can take

- $M = 1$ kg,
- $g = 9.82\ m/s^2$,
- $l = 1$ m,
- $b = 0.001$ kg$/s$.

The assignment consists of the following sub-problems

a) Make two Simulink models of the pendulum.

- One model, which exactly follows the equations above
- An approximate model, obtained by linearization around $\theta = 0$. You get such a model by approximating $\sin(\theta)$ by θ and approximating $\cos(\theta)$ by 1).

Make compact and well-structured models. Let the constants M, l, and b be parameters of the model. Check that they work by simulating them.

b) Now compare the two models to see how much the linearization around 0 affects the behavior of the model, i.e., understand how much error is introduced by the linearization. In this case, we assume (for simplicity) that the coefficient of friction b is 0. Compare the frequencies (or periods) of the two models when starting at rest from a maximal angle of $\theta = \pi/4$ (i.e., you can have initial conditions $\theta = \pi/4$ and $\dot{\theta} = 0$). How much do the frequencies or periods differ (in %)? You should instrument your model by suitable blocks, so that the difference can be seen or calculated (by post-processing) from some output signal of some block (i.e., it is not satisfactory to just simulate each model and inspect the outputs "by hand"). There are of course many ways to do this. One possibility is to add a component which observes the pendulum and counts the number of completed periods, and after some time compares the number of completed periods in both models.

c) At what maximal starting angle (with $\dot{\theta}$) do the frequencies (or periods) differ by only 1%? A satisfactory solution should not find this by "manual trial and error", but programmatically by performing a sequence of simulations to derive the answer.

As solution, you should hand in models, a report briefly describing how they are constructed, and selected simulation output.

Submission

Solutions (all files) to this assignment are to be submitted via the Student Portal by **Thursday, September 10, 2020**