Laboratory 5:

**System Modelling**

In this laboratory you experiment to model and simulate a mechanical dynamic system (single and coupled pendulum).

1. Let us first determine the differential equation describing the movement of a single suspended pendulum:

θ(t)

l

m

***Figure 1 Single Suspended Pendulum***

Physical Law:

Set up equation:

Linearisation:

Differential equation: 

Block diagram:

1. Let us now consider the effect of friction in the rotation point. This can be modelled as a rotation damping, causing a torque equals:



Determine the new differential equation and draw the corresponding block diagram:

Differential equation:

Isolate highest derivate:

Block diagram:

1. Let us now build and simulate this block diagram in Matlab using Simulink. Open the script ***runSimulink \_single\_pendulum.m*** and check for the section:  
   “(C) Run Simulink Simulation” which initialize parameters and open the Simulink model ***single\_pendulum.slx*** , and work through the following steps:

* Complete the Simulink model according to the equation calculated above (with attenuation). Use the parameters defined in the Matlab file to describe the gain values;
* Insert the initial condition ***thetaA\_0*** in the corresponding integrator box;
* Check the logging parameters of the scope block (to import data in Matlab workspace);
* Check the set up in menu >Simulation >Model Configuration Parameters;
* Then carry on the simulation and visualize the results.   
  Hint: the command sim(<modelname>) starts the simulation.

1. Another possibility is to solve the differential equation directly in Matlab. We will do this by applying the hypothesis that an exponential function is a suitable solution. You will see that this hypothesis matches quite well, once you replace it in the differential equation.

Solution hypothesis: 



Replacing in the differential equation:

Resulting algebraic equation:

Back to the solution hypothesis:

1. Let us calculate numerically in Matlab the solution of the 2nd order algebraic equation. And plot the corresponding solution:  time function.

Work in the Matlab script in the section   
“(E) Checking numerical solution of Diff Equation”

1. We also want to compare both solution methods (Simulink and Matlab- via algebraic equation). In order to do so, import the solution from Simulink in the Matlab workspace and plot both solutions in the same graphic.

Work in the Matlab script in the section   
“(F) Compare Simulink and Diff-Equation numerical solution“

1. What could be the cause for the differences you can observe?

How are the time steps specified in Simulink?

Force a smaller time steps in Simulink and test the comparison once more.

1. How does this method of the exponential solution (and conversion from a differential equation to an algebraic equation) relates to the representation with Fourier Transformation?
2. We would like now to use the same approach to model a coupled pendulum.

Open the Simulink model ***coupled\_pendulum.slx*** and apply “backwards engineering” to read out the two differential equations describing the coupled pendulum.





…



θA



…



θB

***Figure 2 Block Diagram coupled pendulum***

1. If you isolate  in the 1st equation, and replace  ,  and  in the 2nd equation, you get a single differential equation describing the complete coupled pendulum. Determine which order would this differential equation have (but without carrying out the replacement calculation).

How does this order relate to the number of integrator boxes in the model?

1. If you apply the same method of testing an exponential solution for the coupled pendulum, which solution (values for s) do you expect to find?

How do these values relate to the response of the pendulum that you can observe in the time domain?

Open and execute the Matlab script Trial\_4\_roots.m and explain the relationship between the 4 roots, applied to the exponential solution and the resulting graphic.

1. Make a copy of your previous Matlab script, and name it ***runSimulink \_coupled\_pendulum.m*** . Define the required constants and add the call for the Simulink simulation of the coupled pendulum.

For the spring used to connect the two pendula you can use:

k = 2.5; % Spring constant in [N/m]

l1 = 0.7; % Length Pendel Rod in [m]

l2 = 0.4; % Torque Arm Spring in [m]

In case you are curious, find next a more detailed description of the coupled pendulum.

The structure diagram of the system is drawn below. It consists of a metal frame with two rotation points (pivot), where two poles with masses in the lower end are attached. The pendulums are coupled via a translation spring. You may neglect the mass of the poles, but you need to consider the friction at the rotation points.

*Figure 1: coupled pendulums with translational spring*

Vocabulary:

pole, rod: der Stab, die Stange

pivot point: der Drehpunkt

spring: der Feder

friction: die Reibung

disc: die Scheibe

torque: das Drehmoment

m

m

k

l1

l2

***Figure 3 shows the proposed notation for the constants and variables of the model:***

* The output signals: deflection angles θA and θB in (rad), which are measured with rotation sensors;
* The position of the spring end points: xA und xB in (m) (system internal states);
* The spring constant k in (N/m) ;
* The mass of the pendulums m in (kg) ; (given m= 1,56 kg )
* The damping constant ct in ( N.m/(rad/s) ) ;
* The length of the pendulums l1 , and the length of the spring fixing-point l2 in (m) .

There is no external input signal, but you can start the pendulum’s movement by giving a non-zero initial condition, for example deflecting one pendel out of its steady state.