C7 Constrained Pendulum – MATLAB/SIMULINK

Single model using tangential velocity

Simulator: MATLAB is a widely used software tool based on numerical vector and matrix manipulation. SIMULINK is a MATLAB-toolbox for graphical modelling and numerical simulation of dynamic systems, offering some extensions for modelling hybrid systems.

Model: The approach in this solution uses a single full-parameterised model that can solve all tasks of the comparison. Model parameters are set via MATLAB scripts with the `set_param()` function for the specified blocks (constant block, memory block, hit-crossing block and switch block).

Using the tangential velocity $v = \dot{\varphi} \cdot l$ instead of the angular velocity $\dot{\varphi}$ prevents discontinuous integrator results due to switching the length of the pendulum on hitting or leaving the pin, i.e. the formula $\dot{v} = -g \cdot \sin\left(\int \frac{v}{l}\right) - \frac{d}{m} \cdot v$ is implemented via SIMULINK blocks.

To resolve the algebraic loop for calculating the actual pendulum length, the memory block `pLen` and the triggered subsystem `Subsystem1` is used. The hit-crossing block is necessary to catch also short hits on the pin.

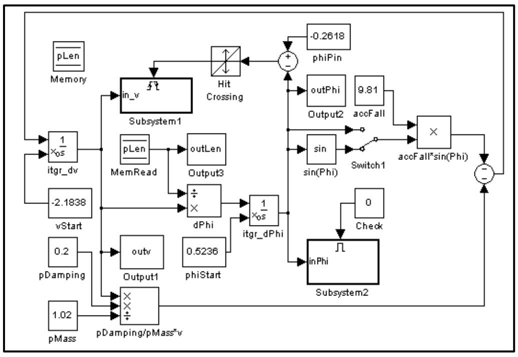

Figure 1: Pendulum model

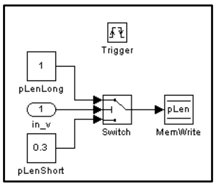

Figure 2: Subsystem1

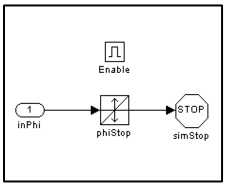

Figure 3: Subsystem2

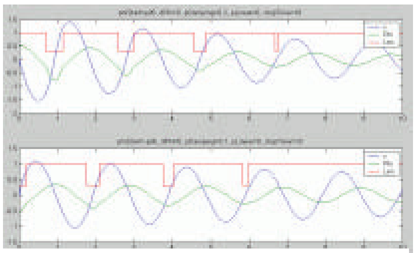

Figure 4: Result of Task a

Task a: Simulation in the time domain. The model is just executed via a MATLAB script that sets the parameters and plots the results. The constant block `Check` disables Subsystem2 and the switch Switch1 selects the nonlinear model.

Task b: Comparison of linear and nonlinear model. The model is simulated two times. After the first run, the switch `Switch1` toggles to the linear system, i.e. the `sin` block is shorted. To compare the results, interpolation is necessary because the output vectors are of different size.

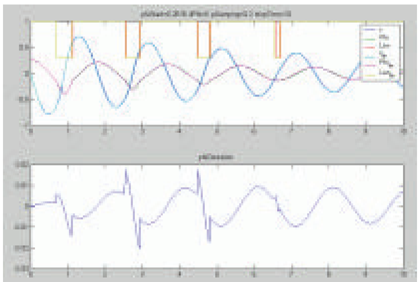

Figure 5: Result of Task b

Task c: Boundary value problem. To calculate the start velocity, the pendulum is started at the given end position using the negative damping d=-0.2 . The subsystem `Subsystem2` (activated via the constant block `Check`) stops the model on reaching the given start position (the hit-crossing offset in `phiStop` equals the start angle). The negative velocity, when the simulation stops, is the required start velocity.

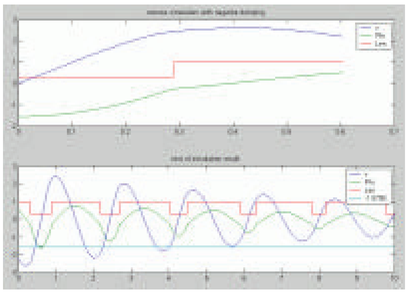

Figure 6: Result of Task c

The second plot in figure 6 shows in a further simulation the correctness of the previous result. The position where the shortened pendulum reaches the maximum angle is marked with an x.

Alexander Kittenberger
e9725395@student.tuwien.ac.at