Control Systems Lab (SC4070)
Rotational Pendulum Experiment

Description

The rotational pendulum is in fact a two-link rigid manipulator operating in a vertical plane. The first (lower) joint, is actuated by a torque introduced by a DC motor. The second (upper) joint is free to rotate. Both links (arms) can rotate the full 360 degrees. The objective is to control the motor such that the joints are stabilized at some desired angles. The easiest task is to control the links around the stable equilibrium (both links down). More difficult tasks are when the second link is in its unstable equilibrium (pointing upward). The schematic diagram in Figure 1 shows the system including all the relevant parameters and variables. Positive directions of variables are indicated by arrows.
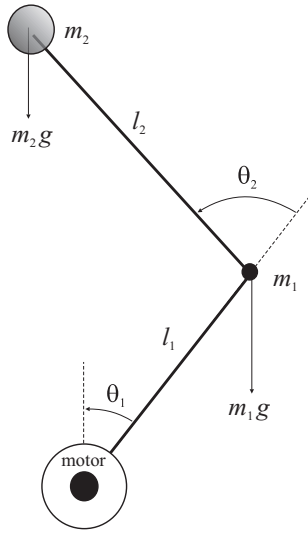

Figure 1: Schematic drawing of the rotational pendulum.

This system has one control input u, which is the torque that accelerates the lower link (delivered by the motor). This input is commanded from the computer and is scaled between -1 (corresponding to the maximal torque moving the arm clockwise) and +1 (corresponding to the maximal torque moving the cart counter-clockwise). There are two measured outputs: θ_1 – the angle of the lower joint, and θ_2 – the angle between the lower and upper joint. These measured values are given in radians. The physical parameters of the system are listed in Table 1.

Control Objective

Design a controller to stabilize the second link in the upright position. It is easier to start with the stable equilibrium (both links down) and just make a controller to damp the swing (like an overhead crane). The controlled system should have zero steady state error in θ_1 (small oscillations around the reference are permitted, as these are caused by Coulomb friction and stiction and are difficult to avoid when using linear control only) and adequate disturbance rejection properties, i.e., it should be able to recover from a small tick against the pendulum.

Table 1: Physical parameters and their values.

Symbol	Parameter	Value
g	acceleration due to gravity	$9.81 \, \mathrm{ms^{-2}}$
l_1	length of first link	$0.1 \, \mathrm{m}$
l_2	length of second link	$0.1 \, \mathrm{m}$
m_1	mass of first link	$0.125 \, \mathrm{kg}$
m_2	mass of second link	$0.05 \, \mathrm{kg}$
c_1	center of mass of first link	$-0.04 \, \mathrm{m}$
c_2	center of mass of second link	$0.06 \, \mathrm{m}$
I_1	inertia of first link	$0.074 \, \mathrm{kgm^2}$
I_2	inertia of second link	$0.00012 \, \mathrm{kgm^2}$
b_1	damping of first joint	$4.8 \, \mathrm{kgs^{-1}}$
b_2	damping of second joint	$0.0002 \, \mathrm{kgs^{-1}}$
k_m	motor gain	$50 \, \mathrm{Nm}$
τ_e	motor electrical time constant	$0.03 \, \mathrm{s}$

Physical Modeling

The nonlinear model equations are given below. They can be derived by using the Euler–Lagrange equations, neglecting the Coulomb friction and stiction:

$$M(\theta)\ddot{\theta} + C(\theta, \dot{\theta})\dot{\theta} + G(\theta) = \left[\begin{array}{c} T \\ 0 \end{array} \right]$$

with

$$T + \tau_e \dot{T} = k_m u \quad \text{and} \quad \theta = \left[\begin{array}{c} \theta_1 \\ \theta_2 \end{array} \right]$$

Here, θ are the joint angles, M is the (positive-definite) inertia matrix, C contains the Coriolis and centrifugal forces, G contains the effects of gravity, and T is the torque applied to the first link. The matrices are:

$$M(\theta) = \left[\begin{array}{cc} P_1 + P_2 + 2P_3 \cos\theta_2 & P_2 + P_3 \cos\theta_2 \\ P_2 + P_3 \cos\theta_2 & P_2 \end{array} \right]$$

$$C(\theta, \dot{\theta}) = \left[\begin{array}{cc} b_1 - P_3\dot{\theta}_2 \sin\theta_2 & -P_3(\dot{\theta}_1 + \dot{\theta}_2)\sin\theta_2 \\ P_3\dot{\theta}_1 \sin\theta_2 & b_2 \end{array} \right]$$

$$G(\theta) = \left[\begin{array}{c} -g_1 \sin\theta_1 - g_2 \sin(\theta_1 + \theta_2) \\ -g2 \sin(\theta_1 + \theta_2) \end{array} \right]$$

with

$$\begin{aligned} P_1 &= m_1 c_1^2 + m_2 l_1^2 + I_1 \\ P_2 &= m_2 c_2^2 + I_2, & P_3 &= m_2 l_1 c_2 \\ g_1 &= (m_1 c_1 + m_2 l_1)g, & g_2 &= m_2 c_2 g \end{aligned}$$

Hint: To implement the M, C and G matrices in Simulink use the 'MATLAB Function' block in the Functions & Tables library. The inverse of M can be computed by using the 'Product' block in the Math library.

Simulink Template

A Simulink template `rpendtemplate.mdl` contains the necessary real-time interface blocks and some scopes. Make your own copy of this file and use it as a starting point for your experiments. Before starting the first simulation, define the sampling period h as a variable in the workspace of MATLAB.