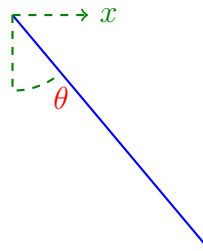


Damping and stabilization of a pendulum



Pendulum dynamics

Consider the pendulum above. If the length of the pendulum is ℓ and a point mass of weight m is hanging at its tip, the Newton force law, in tangential direction of rotation, gives (neglecting friction temporarily)

$$-\ell\ddot{\theta}m = mg \sin \theta + m\ell \cos \theta \cdot \ddot{x}$$

Assume the position x can be controlled by a servo motor with the simplified dynamics

$$\dot{x} = u$$

where u is the input (control variable) to the servo. Let $g_\ell = g/\ell$ and also introduce a friction force that depends of rotational velocity then

$$\ddot{\theta} = -g_\ell \sin \theta - d\dot{\theta} - \cos \theta \cdot \dot{x}$$

This pendulum dynamics has been implemented in the Sysquake file *Pendulum-Damp.sq* where it is animated. Your objective is to implement two controllers. The first controller should damp the pendulum dynamics when it is hanging down (as shown in the figure). The second controller should stabilize the pendulum when it is inverted. Thus, the pendulum can be considered as two different processes, one then it is hanging down and another one then it is inverted.

Linearization

There are two equilibrium states for the pendulum, $\theta = 0$ and $\theta = \pi$. The first one is stable and the other is unstable. The linearized dynamics consider deviation from equilibrium. Let $y = \theta - \theta_0$ and linearize around θ_0 .

$$\begin{aligned}\sin \theta &\approx \sin \theta_0 + \frac{d}{d\theta} \sin \theta_0 (\theta - \theta_0) = \sin \theta_0 + \cos \theta_0 y \\ \cos \theta &\approx \cos \theta_0 + \frac{d}{d\theta} \cos \theta_0 (\theta - \theta_0) = \cos \theta_0 - \sin \theta_0 y\end{aligned}$$

Down

For $\theta_0 = 0$ it follows that $\sin \theta \approx y$ and $\cos \theta_0 \approx 1$ giving

$$\ddot{y} = -g_\ell y - d\dot{y} - \dot{u}$$

and the transfer function ($Y(s) = G(s)U(s)$) is

$$G(s) = \frac{-s}{s^2 + ds + g_\ell}$$

The poles of the continuous-time system are $p_{1,2} = -d/2 \pm i\sqrt{g_\ell - d^2/4}$, which are both in the left half plane (stable) because of the friction term $d > 0$. The less friction the more oscillatory are the poles. Feedback can be used to make the system less oscillatory despite the badly damped poles.

Up

For $\theta_0 = \pi$ it follows that $\sin \theta \approx -y$ and $\cos \theta_0 \approx -1$ giving

$$\ddot{y} = g_\ell y - d\dot{y} + \dot{u}$$

and the transfer function is

$$G(s) = \frac{s}{s^2 + ds - g_\ell}$$

This system has one unstable pole $p_1 = -d/2 + \sqrt{d^2/4 + g_\ell} > 0$ and one stable pole $p_2 = -d/2 - \sqrt{d^2/4 + g_\ell} < 0$. The system is therefore unstable and any small perturbation will make the pendulum fall down. Feedback is needed in order to stabilize this system.

Problem 1 — Sampling

Calculate the sampled systems using zero-order-hold sampling for the two cases above. Thus, calculate the discrete-time system

$$A(q^{-1})y(k) = B(q^{-1})u(k)$$

for the case when pendulum is

- a) down (stable but badly dampened)
- b) up (unstable)

Use the parameter values $g_\ell = 0.981$, $d = 0.1$ and sampling interval $h = 1$ s.

Problem 2 — Estimation

Open the file *PendulumDamp.sq* in Sysquake. Two windows are shown, one where the pendulum is animated graphically and another one showing the input u (blue) and the output y (red). In order to access these data samples in the command window, write there

```
global samples
```

The variable `samples` is now continuously undated and the latest 100 samples are stored there. To make a snap-shot and store these samples you just write

```
s=samples;
```

and exactly those samples that are seen in the Signal-window will be stored in your variable s . It contains three columns of data samples: sampling time, u and y . Start an identification experiment by moving the mouse into the Signal-window. A horizontal line is then shown, following your mouse movements. This also changes the input u every sampling instant. Keeping the mouse in the middle makes $u = 0$, when dragging it up u increases and dragging it down u decreases. By moving the mouse up and down you excite the system and the pendulum is swinging around. For the case when the pendulum is hanging down (stable system) the parameters in the discrete-time description can be estimated experimentally. Do this, as described below.

- a) Perform an identification experiment. Start and stop by keeping $u = 0$ and try to make $mean(u) \approx 0$ for the entire excitation. When you have data samples you think you can use, make a snap-shot and extract these from your variable s . Show how you can use the extracted data samples for least-square estimation and determine the parameters in A and B . Compare your estimated A and B to your calculated in Problem 1a.
- b) Compare the real output y to your modeled response yh .

```
yh=filter(B,A,u);
plot([y yh]','br');
```

Problem 3 — Control design

Calculate feedback control designs as

$$R(q^{-1})u(k) = -S(q^{-1})y$$

for the two system configurations down and up, such that the closed-loop characteristic polynomial becomes

$$AR + BS = A_c$$

You may use the Sysquake file `polp.sq` for the calculation. It allows to move closed-loop poles while displaying, for example, step responses and sensitivity functions. Enter your system from the Setting menu and then start to move poles of A_c . To see the resulting controller polynomials, click on **Dump Data** in the File menu. They will then be shown in the command window.

Investigate the following

Down: For the down-situation you have both a calculated model from Problem 1a and an estimated model from Problem 2a. Both of these models will be used in the design below.

- a) Choose $A_c = 1$ with design based on calculated model (Problem 1a).
- b) Choose $A_c = 1$, with design based on estimated model (Problem 2a).

Up:

- c) Choose $A_c = 1$
- d) Choose a pole-placement that minimizes $\|\mathcal{S}_y\|_\infty$

Problem 4 — Implementation and evaluation

Open `PendulumDamp.sq` in a text editor and write the code for the implementation of the control law. In the end of the file you will find where to put the code for the difference equation of the controller. When the angle $|\theta| < \pi/2$, the output is defined as $y = \theta$ and the controller should damp the pendulum hanging down. If you move the pendulum with the mouse such that $|\theta| > \pi/2$, the output is defined as $y = \theta - \pi$ and another controller should stabilize the pendulum in up-position. You may evaluate the performance by moving the pendulum by the mouse, then leave it and see how the controller damp or stabilize the pendulum by the feedback controller automatically.

Report

Answers to all problems above should be clearly documented in a report with comments and evaluations. In addition, your edited `sysquake`-file with the implementation of your controllers from Problem 3a or 3b for down-control and 3d for up-control should be e-mailed to me for checking.