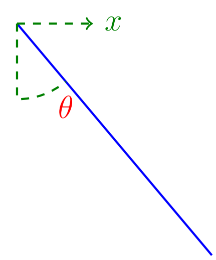
Pendulum position control



Hierarchical control of position and pendulum

The previously studied pendulum designed for damping and stabilization by freedback from the angle is now considered again. This time, in addition to stabilization, a position reference should be introduced and the designs changed such that the pendulum follows the reference. A hierarchical control structure is chosen where first an inner loop for the position control by the servo is designed and then an outer loop for pendulum stabilization is designed. In the outer loop the position is considered to be the input to the system.

Servo position control

The servo model is an integrator with unit gain for simplicity, i.e.

$$\dot{x} = u$$

where x is the position and u the input (voltage) to the servo. After zero-order-hold sampling with sampling period $h = 1$, the discrete-time model becomes

$$x(k) = \frac{\mathrm{q}^{-1}}{1 - \mathrm{q}^{-1}} u(k)$$

Since this system is unstable, it must be stabilized by feedback. Choosing the P-controller

$$u = r - x$$

Then the closed-loop becomes (dead-bead $A_c = 1$)

$$x(k) = \mathrm{q}^{-1} r(k) = r(k-1)$$

and the position will follow the reference directly after one sampling period. This completes the design of the inner loop. You may try how it performs so far by opening the Sysquake-file `PendulumMove.sq`. In the animated pendulum window you can click on the reference r (a black ring) and move it horizontally. The pendulum will then follow after one sampling period, causing the large oscillations. The task is to avoid these oscillations while moving the pendulum. This is the objective of the outer loop design and your challenge.

Stabilization while moving the pendulum

Since the reference to the servo is going to act as input to the outer loop we change notation and call it v. The external variable r should then be used as reference for v. The inner loop is now

$$x = \mathrm{q}^{-1} v$$
$$u = (1 - \mathrm{q}^{-1}) v$$

Let the pendulum dynamics (for either down or up situation) be

$$Ay = Bu$$

Substituting u gives the system of interest

$$Ay = B(1 - \mathrm{q}^{-1}) v$$

The controller to find for the outer loop is

$$Rv = -Sy + Tr$$

where R and S are solved from the polynomial equation

$$AR + B(1 - \mathrm{q}^{-1}) S = A_c$$

and A_c is chosen for stabilization or damping of the pendulum dynamics. Since

$$v = \frac{AT}{A_c} r$$

it follows that the simplest choice for T is to adjust the steady-state gain as

$$T = \frac{A_c}{A}(1) = R(1)$$

which makes $v(k) \to r(k)$, when $k \to \infty$.

Problem 1 — Control design

Use the zero-order-hold sampled models derived in previous exercise and make controller design as above for the pendulum down and up. Base the design on the sampled models.

Down:

a) Find R and S that gives $\|\mathcal{S}_y\|_\infty < 2$ and choose $T = R(1)$.

Up:

b) Find R and S that gives $\|\mathcal{S}_y\|_\infty < 4$. Then in order to speed up the reference response, cancel the slowest pole in A_c by including it in T. Thus, if λ_s is the dominating (slowest) pole choose $T = R(1)(1 - \lambda_s \mathrm{q}^{-1})/(1 - \lambda_s)$.

Problem 2 — Implementation and verification

Modify the code in the Sysquake-file `PendulumMove.sq` at the indicated places in the end of the file. Both the controller for pendulum down and up should be written. The inner loop is already implemented.

Report

Document in a report your answers to Problem 1 and 2. Include plots showing performance. The best (in your opinion) controller implementations should be included in one edited Sysquake file and sent to me by e-mail for verification.