Estimation for control – Practical Assignment 5
Discrete-time linear observer design

Logistics

- This practical assignment should be carried out by a group of maximum two students.
- The assignment solution consists of Matlab code and Simulink model. This code will be checked and run by the teacher during the lab class, and your attendance to the lab will only be registered if you have a working, original solution. Validated attendances for all the labs are necessary for eligibility to the exam. Moreover, at most two labs can be recovered at the end of the semester, which means accumulating three or more missing labs at any point during the semester automatically leads to final ineligibility.
- Discussing ideas among students is encouraged; however, directly sharing and borrowing pieces of code is forbidden, and any violation of this rule will lead to disqualification of the solution.

Assignment description

Prerequisite:

- A nonlinear model, with state vector $x = [x_1,\, x_2, ...x_n]^T$, and input vector $u = [u_1,\, u_2, ...u_m]^T$ where $n \geq 3$, is the number of state variables, $m \geq 1$, is the number of inputs, and the dynamic equation has the form:
$$\begin{aligned} \dot{x} =& f(x,u) \\ y =& Cx, \end{aligned} \tag{1}$$
where $f(x) = [f_1(x,u), f_2(x,u), ...f_n(x,u)]^T$ is a vector function with at least one nonlinear term.
- discrete-time nonlinear and linear models – implementation in Matlab or Matlab-Simulink

Recap on discrete-time observer design

The discrete-time observer design follows the same procedure as in the continuous-time. The considered discrete-time linear observer has the form:

$$\begin{aligned} \hat{x}(k+1) =& A\hat{x}(k) + Bu(k) + L(y(k) - \hat{y}(k)) \\ \hat{y}(k) =& C\hat{x}(k), \end{aligned} \tag{2}$$

where $\hat{x}(k)$ is the estimate of $x(k)$ in discrete time step k. The discrete-time error dynamics is the same as it was in the continuous-time (Laboratory 4):

$$e(k+1) = (A - LC)e(k)$$

Just now everything is written in discrete-time, and in order to obtain a stable system, the eigenvalues of the error dynamics described with $(A - LC)$, should be inside of the unit circle, not in the left half plane!

In order to determine the observer gains the `place` function can be used in Matlab, in the following way: L^T `= place(`A^T`,`C^T`,[poles]).`

Consider the following pseudocode as a good starting point for your implementation: The nonlinear

---

declare matrices A, B, C
initial conditions for x, $\hat{x}$
compute L with `place()`
repeat
    $u(k) = -Kx(k)$
    $x(k+1) = Ax(k) + Bu(k)$
    $\hat{x}(k+1) = A\hat{x}(k) + Bu(k) + L(Cx(k) - C\hat{x}(k))$
    $k = k + 1$
until $k_{\max}$ was reached

---

dynamics can be similarly implemented. As it can be see in the algorithm we are using the states of the system and not the estimate. So, next try your algorithm with $u(k) = -K\hat{x}(k)$.

Requirements:

- Change your C matrix: $C \neq I$ (is not the identity matrix), $D = 0$ (D is still zero matrix)
- Check if the discrete-time linear system is fully state observable, by calculating the rank of the observability matrix:

$$P_o = \begin{bmatrix} C \\ CA_d \\ . \\ . \\ . \\ CA_d^{n-1} \end{bmatrix}$$

  , where the A_d refers to the discrete-time matrix.
- Find the observer gain L
- Test observer on both discrete-time linear and nonlinear models, without control
- Use your state-feedback control, $u(k) = -Kx(k)$ with the estimated states: $u(k) = -K\hat{x}(k)$
- Compare the estimated state, $\hat{x}(k)$, with the true state, $x(k)$

Hint: Useful Matlab functions `obsv`, `place`.