Estimation for control – Practical Assignment 8
Unknown input 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 linear 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:
$$\begin{aligned}\dot{x} =& Ax + Bu \\ y =& Cx,\end{aligned} \tag{1}$$
where A and B are the system matrices and C is the output matrix.
- discrete-time linear model:
$$\begin{aligned}x(k+1) =& A_d\, x(k) + B_d\, u(k) \\ y(k) =& C\, x(k)\end{aligned} \tag{2}$$
where A_d and B_d are the discrete-time system matrices.

Unknown input observer design

In some cases it can happen that the system is affected with some disturbance inputs what we cannot directly measure, for instance, imagine the case when we have a pendulum system, which we want to stabilize around an equilibrium point. The control works well, but the experiment is in a windy place and the model is disturbed with a constant wind. If we know some information of the effect of the disturbances, like what states are affected, with what amplitude etc., we can include it in the model. The following continuous-time linear model can be considered in this case, including the disturbance:

$$\begin{aligned}\dot{x} =& A\, x + B\, u + E\, d \\ y =& C\, x,\end{aligned} \tag{3}$$

where $d \in \mathbb{R}^{n_d}$ is the disturbance and E is the disturbance matrix. It is assumed that the disturbance is constant or piece-wise constant, so the derivative is 0: $\dot{d} = 0$.

Note that, the notation 0 defines a zero matrix of appropriate dimension.

We denote the extended state vector with $x_e := \begin{bmatrix} x \\ d \end{bmatrix}$, and the dynamics of x_e is:

$$\begin{bmatrix} \dot{x} \\ \dot{d} \end{bmatrix} = \begin{bmatrix} A & E \\ 0 & 0 \end{bmatrix} \begin{bmatrix} x \\ d \end{bmatrix} + \begin{bmatrix} B \\ 0 \end{bmatrix} u. \tag{4}$$

The objective here is to estimate these disturbances, so later we can use it in the controller design to compensate their effects. We denote

$$A_e := \begin{bmatrix} A & E \\ 0 & 0 \end{bmatrix}, \quad B_e := \begin{bmatrix} B \\ 0 \end{bmatrix}, \quad C_e := \begin{bmatrix} C & 0 \end{bmatrix} \tag{5}$$

so our new model has the form:

$$\begin{aligned} \dot{x}_e =& A_e x_e + B_e u \\ y =& C_e x_e. \end{aligned} \tag{6}$$

Now we managed to obtain an extended model dynamics for which we can design a linear observer. Note that, in order to design the observer first we need to verify if the $(A_e,\, C_e)$ pair is observable. The rest of the observer design is the same as in the standard Luenberger observer design case, but for equation (6).

Example 1

Consider the following linear model:

$$\begin{aligned} \dot{x}_1 =& x_2 + 5d \\ \dot{x}_2 =& -2x_1 - 3x_2 + 4u, \\ \\ y =& x_1 \end{aligned} \tag{7}$$

this can be written in the form of (3):

$$\begin{aligned} \begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \end{bmatrix} &= \underbrace{\begin{bmatrix} 0 & 1 \\ -2 & -3 \end{bmatrix}}_{\text{A}} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} + \underbrace{\begin{bmatrix} 0 \\ 4 \end{bmatrix}}_{\text{B}} u + \underbrace{\begin{bmatrix} 5 \\ 0 \end{bmatrix}}_{\text{E}} d \\ y &= \underbrace{\begin{bmatrix} 1 & 0 \end{bmatrix}}_{\text{C}}. \end{aligned} \tag{8}$$

Since it is assumed that the disturbance is piece-wise constant ($\dot{d} = 0$), the extended state vector, and the extended matrices are the following:

$$x_e \begin{bmatrix} x_1 \\ x_2 \\ d \end{bmatrix}, \quad A_e = \begin{bmatrix} A & E \\ 0 & 0 \end{bmatrix} = \begin{bmatrix} 0 & 1 & 5 \\ -2 & -3 & 0 \\ 0 & 0 & 0 \end{bmatrix}, \quad B_e \begin{bmatrix} 0 \\ 4 \\ 0 \end{bmatrix}, \quad C_e = \begin{bmatrix} 0 & 1 & 0 \end{bmatrix}. \tag{9}$$

The observer has the form:

$$\begin{aligned} \dot{\hat{x}}_e =& A_e \hat{x}_e + B_e\, u + L(y - \hat{y}) \\ y =& C_e \hat{x} \end{aligned} \tag{10}$$

The rank of the observability matrix of the extended model is 3, which means the extended model is observable. We want a stable observer, so the desired poles (eigenvalues) are negative: $\text{eig}_{des} = \begin{bmatrix}-1 & -2 & -3\end{bmatrix}$. The discrete-time observer design follows the same procedure as in the continuous-time. The considered discrete-time linear observer has the form:

$$\begin{aligned}\hat{x}(k+1) =& A\hat{x}(k) + Bu(k) + L(y(k) - \hat{y}(k))\\ \hat{y}(k) =& C\hat{x}(k),\end{aligned} \tag{11}$$

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`.