Estimation for control – Practical Assignment 2
Linear model, state-feedback control, 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.

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 dynamics having the form:
$$\begin{aligned} \dot{x} =& f(x,u) \\ y =& Cx. \end{aligned}$$

Assignment description

In this assignment first you need to find an equilibrium point of your model. Based on [1] an equilibrium point for a nonlinear dynamic system of the form $\dot{x} = f(x,u)$ can be found in the point where the dynamic system of equations is 0, i.e. $f(x_0, u_0) = 0$, where $x_0 = [x_{10},\, x_{20}, ..., x_{n0}]^T$ and $u_0 = [u_{10},\, u_{20}, ..., u_{m0}]^T$. For example, consider the inverted pendulum model:

$$\begin{aligned} \dot{x}_1 &= x_2 \\ \dot{x}_2 &= 2\sin(x_1) - x_2 + u, \end{aligned}$$

we can denote $x = \begin{bmatrix} x_1 & x_2 \end{bmatrix}^T$, and $\dot{x} = \begin{bmatrix} \dot{x}_1 & \dot{x}_2 \end{bmatrix}^T$, so we have $\dot{x} = f(x,u)$, where

$$f(x,u) = \begin{bmatrix} x_2 \\ 2\sin(x_1) - x_2 + u \end{bmatrix}.$$

An equilibrium point can be found at $f(x_0, u_0) = 0$, which for this example is:

$$\begin{aligned} x_{20} &= 0 \\ 2\sin(x_{10}) - x_{20} + u_0 &= 0 \end{aligned}$$

From here we obtain that $x_{20} = 0$, $u_0 = 0$. For x_{10} we have many options since $\sin(x_{10}) = 0$, can be obtained for $x_{10} = n\pi$, where n can be any natural number. This makes sense, since if we think about the physical meaning of the states every even number n means the point up position and every odd means the pointing down position of the pendulum.

Once we have the equilibrium point we can use the Taylor series approximation:

$$
\begin{aligned}
g(x_1, ..., x_n,\ u_1, ..., u_m) =& f(x_{10}, x_{20}, ..., x_{n0}) + \frac{\partial f}{\partial x_1}|_{x_0,u_0}(x_1 - x_{10}) + ... + \frac{\partial f}{\partial x_n}|_{x_0,u_0}(x_n - x_{n0}) \\
&+ \frac{\partial f}{\partial u_1}|_{x_0,u_0}(u_1 - u_{10}) + ... + \frac{\partial f}{\partial u_m}|_{x_0,u_0}(u_m - u_{m0}).
\end{aligned}
\tag{1}
$$

Requirements:

- Find the equilibrium points of the model
- linearize the model around an equilibrium point
- Define your model in the classical linear state-space representation:

$$
\begin{aligned}
\dot{x} =& Ax + Bu \\
y =& Cx
\end{aligned}
$$

- Add the linear model to the Matlab-Simulink design
- Compute the controllability matrix P_c. If your model is fully state controllable, then design a state-feedback control law: $u = -Kx$. Use the pole-placement to obtain the control gain K.
- Add the control law to your Simulink model, and simulate it with non-zero initial conditions. What do you expect? Is it working properly?
- Assume that only some of the states are available. For example at the pendulum example we only measure the angle, x_1, but we do not measure the angular velocity, x_2. In this case the output matrix is $C = \begin{bmatrix} 1 & 0 \end{bmatrix}$. Choose some of the states that are measured in your model, and leave the others unmeasured by not considering C as the identity matrix.
- Compute the rank of the observability matrix P_o. If it is full rank, then compute a linear observer for your model, using again the pole-placement technique.
- Add the observer to your linear model. Give a different initial condition for the observer, and simulate it. Is it working as it was expected? Is the error converging to 0?

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

References

[1] H. K. Khalil, *Nonlinear Systems, Third edition*, P. Education, Ed. Prentice Hall, Upper Saddle River, NJ 07458, 2000.