# **Estimation for control** – Practical Assignment 4
# Discrete-time linear model, control and estimation

## 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 dynamic equation has the form:
$$\dot{x} = f(x, u).$$
- A linear model in classical state space form, linearized around an equilibrium point
- A state feedback control law, which brings the system to 0 from a non-zero initial condition
- An observer, designed for the linear model, implemented in Simulink

## Assignment description

In many real application a digital controller is used, which requires discrete-time analysis. For this reason, today we are going to discretetize the model obtained for continuous-time.

There are many ways to obtain a discrete-time model from which we highlight the Euler discretization. The basic idea is coming from the approximation of the derivation in the following way:

$$\dot{x} \simeq \frac{x(k+1) - x(k)}{T_s}. \tag{1}$$

It means that the derivative can be approximated with the difference between the state values at point $k+1$ and $k$, divided by the sampling time, $T_s$. If the sampling time is small enough, then we have a good approximation. Lets see what happens with the linear model, if we use this approximation:

$$\dot{x}(t) = Ax(t) + Bu(t) \quad \Rightarrow \quad \frac{x(k+1) - x(k)}{T_s} = Ax(k) + Bu(k). \tag{2}$$

Basically the continuous-time is converted into a discrete-time model. In addition to this we can further simplify the model and express just $x(k+1)$:

$$x(k+1) - x(k) = T_s\, Ax(k) + T_s\, Bu(k) \quad \Rightarrow \quad x(k+1) = (I - T_s\, A)x(k) + T_s\, Bu(k). \tag{3}$$

We can define the new discrete-time state matrices as $A_d = I + T_s A$, and $B_d = T_s B$; we have the following model:

$$x(k+1) = A_d x(k) + B_d u(k) \tag{4}$$

Keep in mind that, in order to obtain stability for the discrete-time model, the eigen-values of the closed loop system have to be inside of the unit circle (not negative, as it was in the continuous-time case).

**Requirements:**

- Find the discrete-time model of the linear and nonlinear model
- Implement both the linear and the nonlinear model in Matlab, not in Simulink!
- Compute a state-feedback control gain to stabilize the system, add the control to your linear model
- Compute and observer gain, so that the error dynamics is converging to 0; add this also to your linear model
- apply the previously computed control and observer gains to your nonlinear model