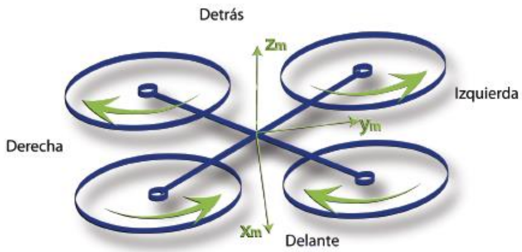
Practical session 3 : Identification of a quadrotor system

<u>Objectives:</u> Perform a black-box linear identification. Black –box identification is data driven, with little additional knowledge from the first principles that drive the system. The objective is to obtain a linear model and its associated accuracy in reproducing the experimental data.

<u>Problem description:</u>

The system to be identified is the ARDrone, an unmanned autonomous vehicle (UAV) designed and built by Parrot as a leisure UAV (https://www.parrot.com/es/drones/parrot-ardrone-20-elite-edition#parrotardrone-20-elite-edition).

The displacement of the vehicle is achieved by modifying the rotation speeds of its rotors. Opposed rotors rotate in the same direction. Manoeuvres are achieved by changing the pitch, roll and yaw angles of the vehicle. For example, in order to move forward the pitch angle needs to be modified by reducing the rotation speed of front motors and increasing the rotation speed of rear motors.



The vehicle has already implemented in it the control loops to follow pitch, roll and yaw references. Those controllers almost completely decouple the velocities in (x_m y_m, z_m) and were designed to facilitate the vehicle guidance.

The experiments begin with the ARDrone hovering (flying but keeping its absolute position still) and involve pitch and roll setpoints as input variables and x, y, v_x, v_y as outputs.

<u>Data inspection and first hypothesis:</u>

The data set: `exp_ini_fase1.txt` is going to be used. It can be loaded calling the provided .m script `loadFlightData`. Matlab variables `mp`, `mr` , `x`, `y`, `vx` and `vy` are created in the workspace and plotted for visual inspection.

Work to be done:

1. **Data preparation.** This session will consider only data related with the displacement of the vehicle in *x* axis, therefore the first task is to identify the input and output signals involved in *x* variations. Divide the data in three sets: identification, test and validation. The first set should cover the first series of input steps, the second set should contain the second series of input steps and the third set should contain the final pseudo-random burst.

   You have to use the identification data set for creating models, the test data set to evaluate those models and the validation data set to compare models performance. Smaller portions of the identification data set can be selected if nonlinear behavior is suspected.

2. **Verification of position as being the velocity integral:** It seems physically obvious that there is a very standard relation between position and velocity variables. The objective of this exercise is to verify that relation in `x` and `vx`. If that is correct, it is possible to consider a simpler relation between input and output (velocity *vs* setpoint instead of position *vs* setpoint).

   Suppose that the integrator relation can be observed from discrete data as:

   $$x(t)\text{-}x(t\text{-}1) = b\cdot\, v_x(t\text{-}1) + v(t)$$

   where $x(t)$ and $v_x(t)$ are measured signals, b is the parameter to be identified and v(t) is the uncertainty due to modelling (for example discretization errors) and measurement.

   Identify parameter b using the command `arx` and the provided data. Obtain also the typical deviation of b, σ_b.

   Are the data consistent with the integrator behavior? Why?

3. **Delay estimation.** One of the most relevant parameters of a model is the input delay of the system. It is, therefore, interesting to dedicate some time estimating the delay. Matlab has several commands that perform a complete delay estimation, look for those commands (in the identification toolbox help) and try them.

   These commands are useful but it is advised in the help page that visual inspection and delay computation is always necessary. Therefore, check the data manually (you can use the piece or pieces you think are more informative) and obtain the input delay estimate of the system.