Introduction to Rotorcraft Control Systems

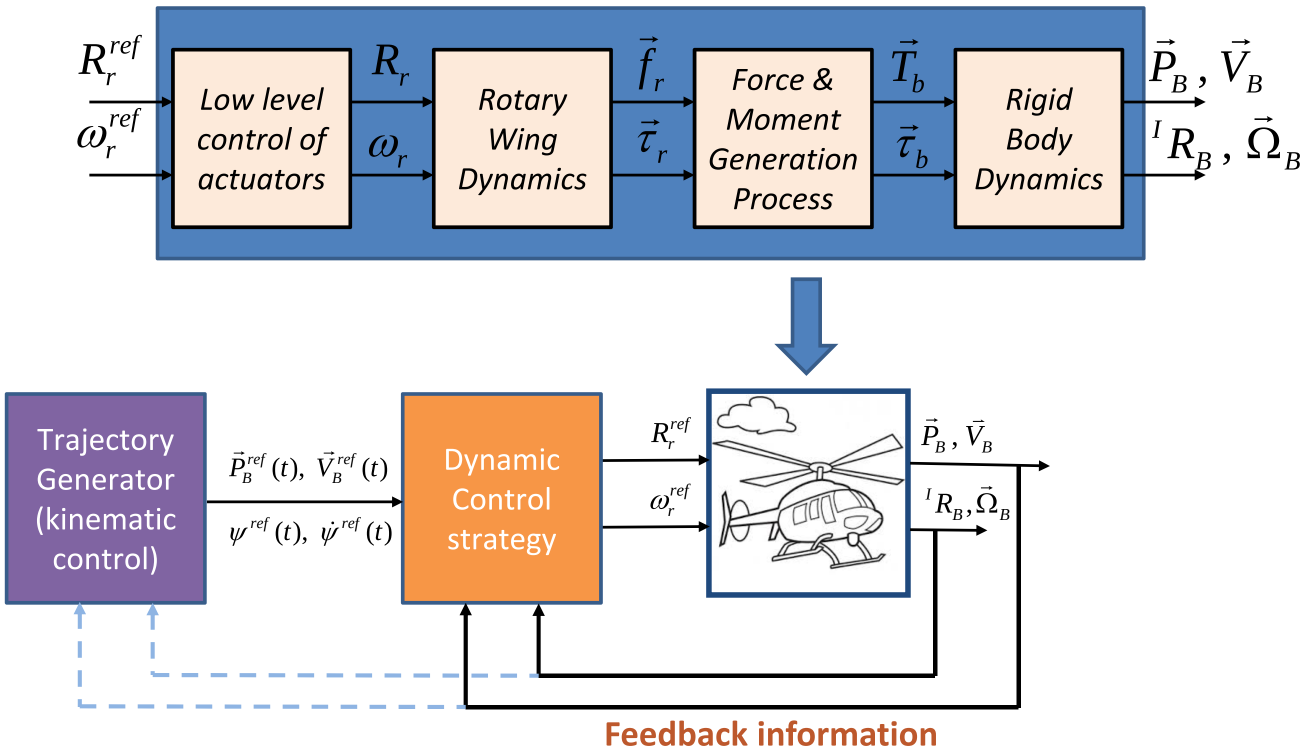
CONTROL Y PROGRAMACIÓN DE ROBOTS

Grado en Electrónica, Robótica y Mecatrónica

Outline

1. Introduction
2. Control Fundamentals applied to RPAS.
3. Stabilization Control.
4. Path Tracking Control.

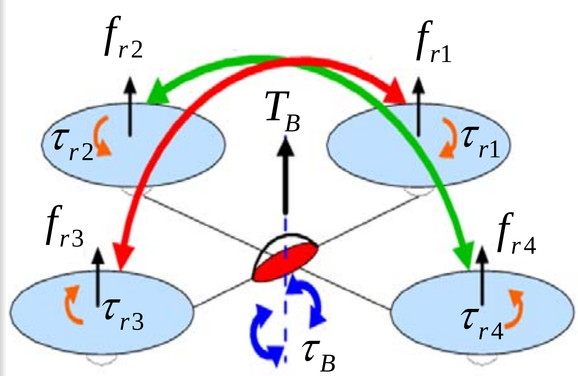
Problem



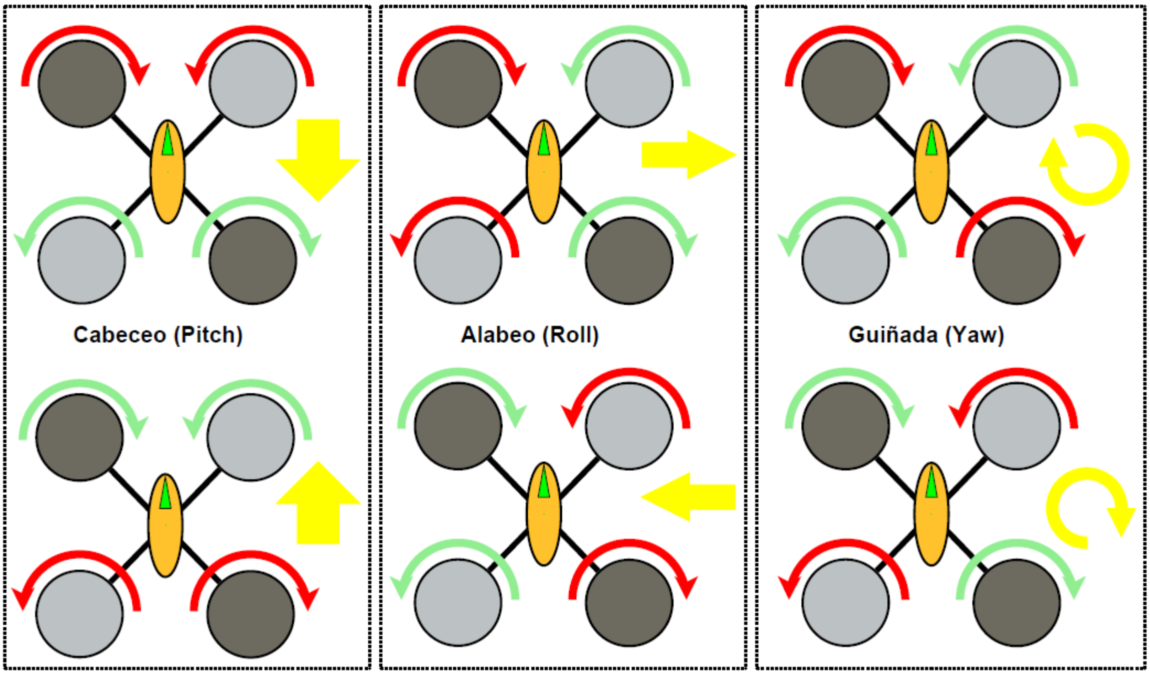
Introduction

QuadRotor helicopter

- Underactuated system
  - 4 actuators
  - 6 DOF
    - rotation $(\phi,\ \theta,\ \psi)$
    - translation $(x,\ y,\ z)$
- Variable rotor speeds
- Decoupled inputs
- Control strategies
  - Cascade structure
  - Augmented state-space



Introduction



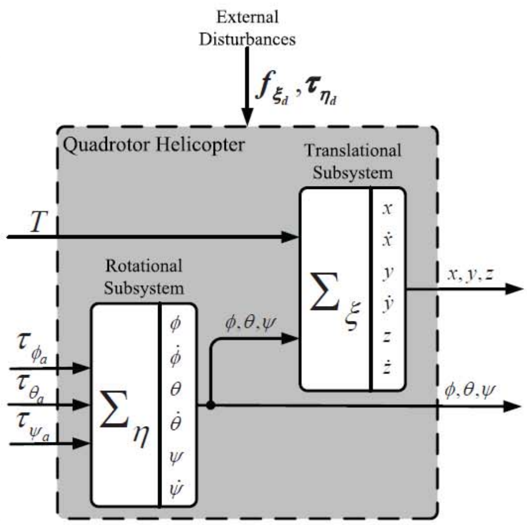
Helicpoter dynamics

- Simplified equations of motion $\rightarrow$ center of mass congruent with the center of rotation
- Separation in 2 interconected subsystems
  1. Rotational subsytem

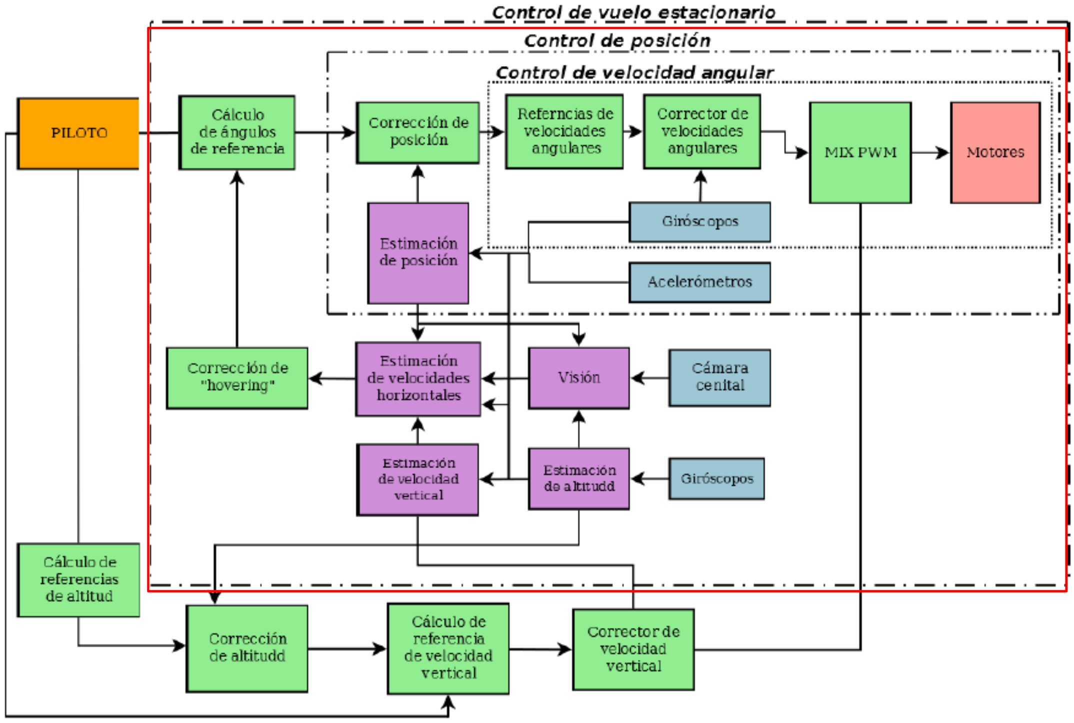
     $$\mathcal{J}(\boldsymbol{\eta})\ddot{\boldsymbol{\eta}} + \mathbf{C}(\boldsymbol{\eta}, \dot{\boldsymbol{\eta}})\dot{\boldsymbol{\eta}} = \boldsymbol{\tau}_{\boldsymbol{\eta}}$$

  2. Translational subsytem

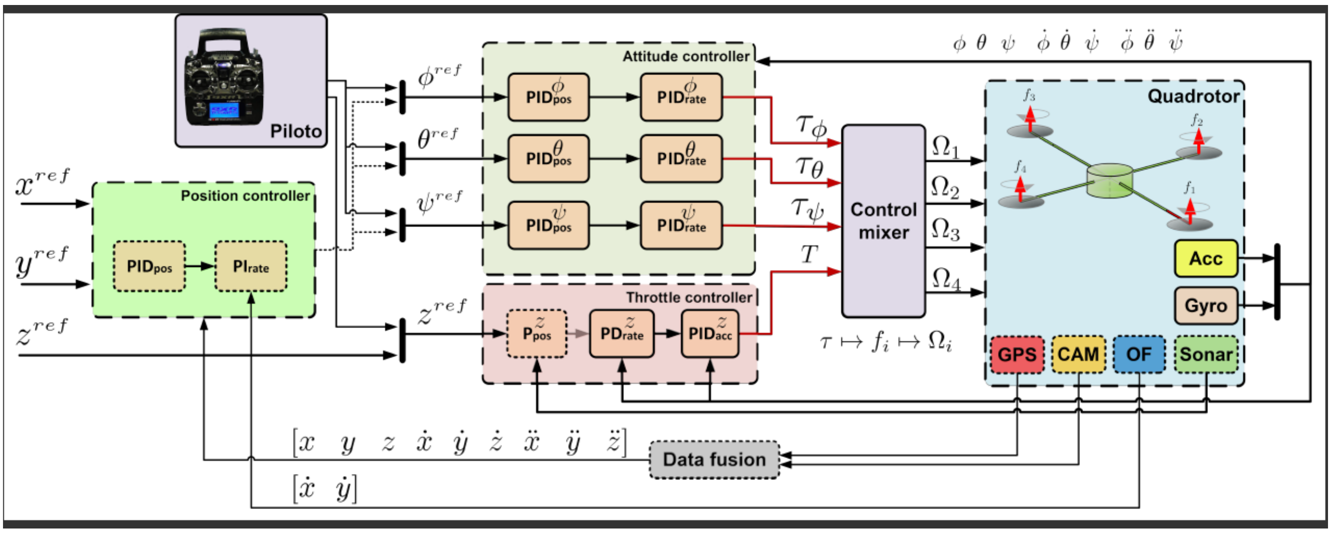
     $$m\ddot{\xi} + mg\mathbf{e}_3 = \mathbf{f}_\xi$$



AR_Drone Control



Quadrotor Control (Arducopter)



Control levels

Low level

- Speed control of motors. Speed sensor.
- Usually a PI or PID controller

Stabilization Control

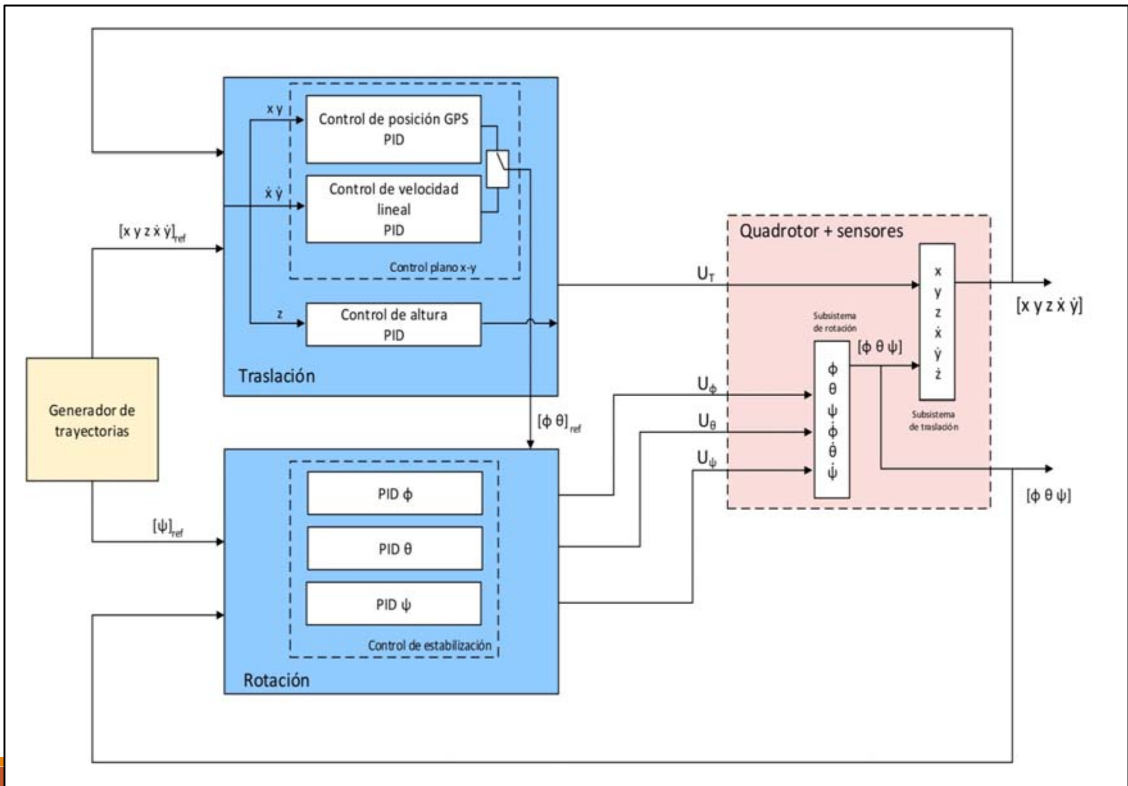
- Navigation angles control. IMU sensor
- Height control of gravity center. Height sensor.

Translation in x-y plane

- Position control. GPS sensor
- Speed control of the angles. From IMU.

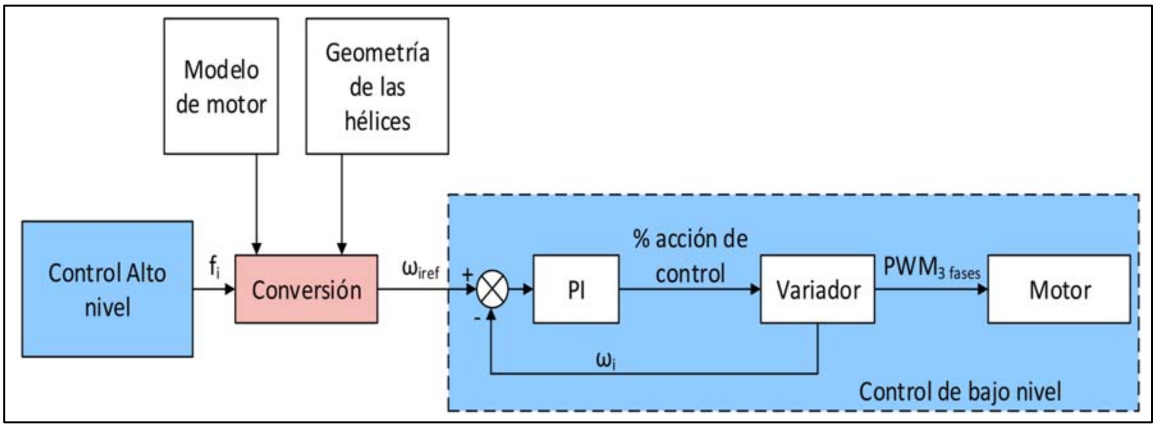
Path tracking control

Quadrotor Control (Arducopter)



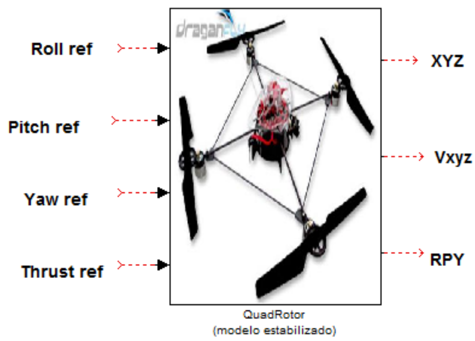
Quadrotor Control

- **Low level control**
  - **Speed control of motors**
  - **Usually a PI or PID controller**



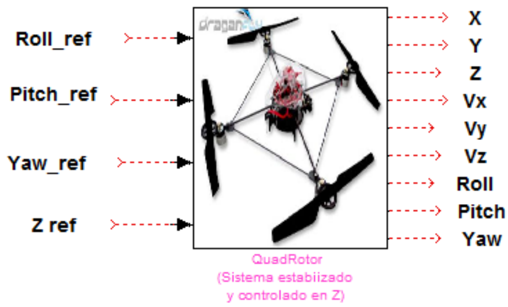
Quadrotor Control

- **Stabilization control**
  - **Angles control (Roll, Picth, Yaw)**
  - **Thrust control (T)**
  - **Usually a PI or PID controller with cascade scheme o with realization taking into account the velocity measurement.**



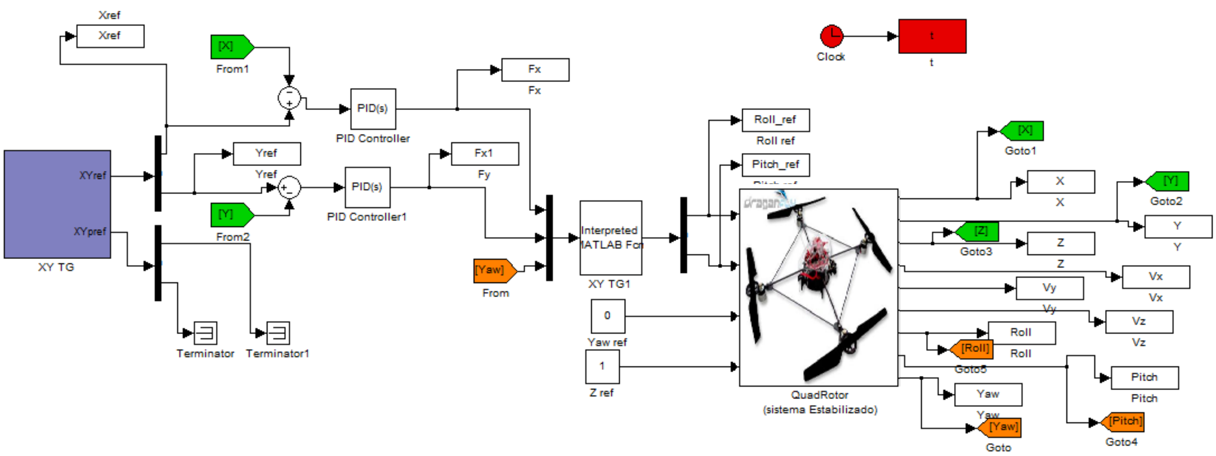
Quadrotor Control

- **Altitude control**
  - Angles control (Roll, Picth, Yaw)
  - Thrust control (T)
  - Z control

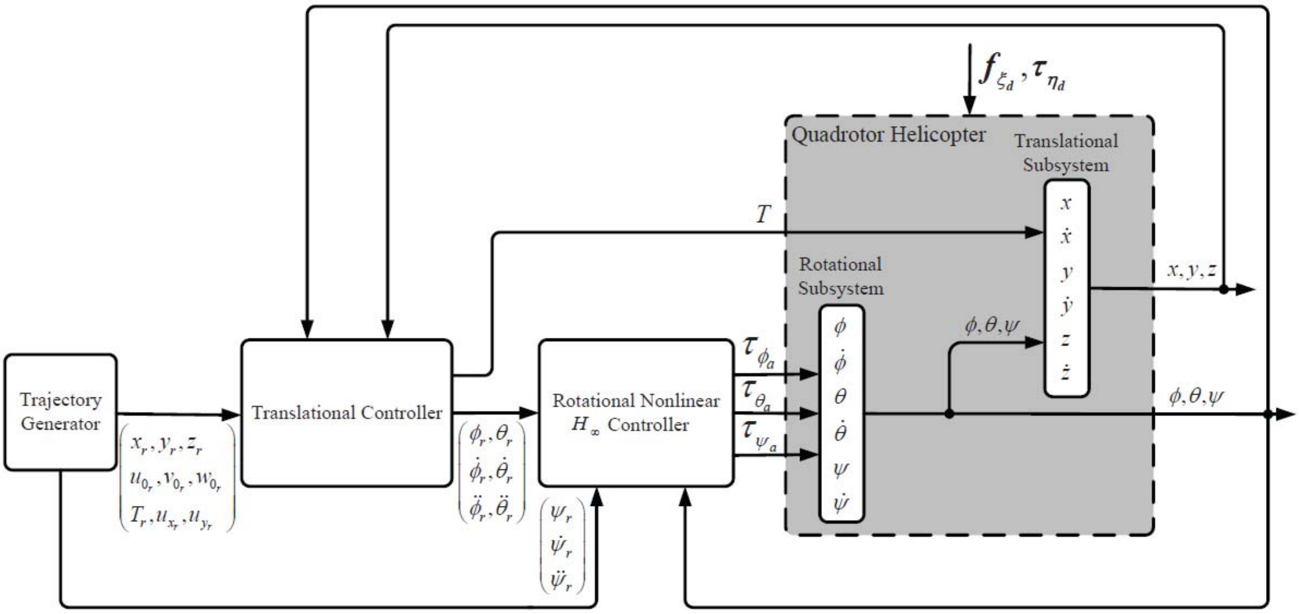


Quadrotor Control

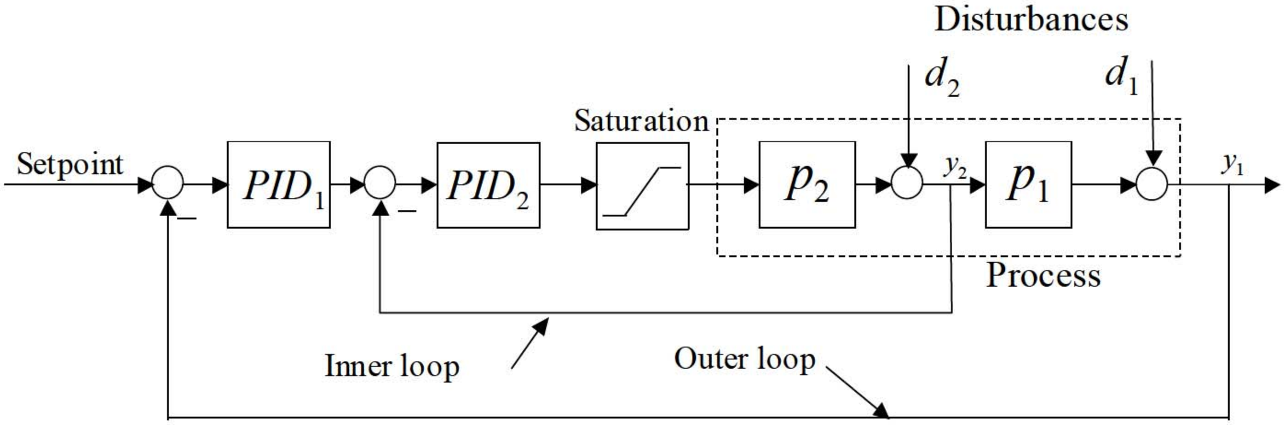
- **Path tracking control**
  - Trajectory control in XYZ



Cascade Control



Cascade Control (single loop)



Traditional cascade block diagram

Conditions for Cascade Control

1. There must be a clear relationship between the measured variables of the primary and secondary loops.
2. The secondary loop must have influence over the primary loop.
3. Response period of the primary loop has to be at least 4 times larger than the response period of the secondary loop.
4. The major disturbance to the system should act in the primary loop.
5. The primary loop should be able to have a large gain, Kc.

Cascade Control

CASCADE CONTROL PROS

Accounts for disturbances in the primary variable more quickly and hence control the primary variable more effectively.

Reduces the effects of dead time and phase lag time in the system

Can be combined with feed-forward control.

Integrated multiple sensor readings.

Improve dynamic performance and provide limits on the secondary variables

CASCADE CONTROL CONS

Cascade control makes the system more complex

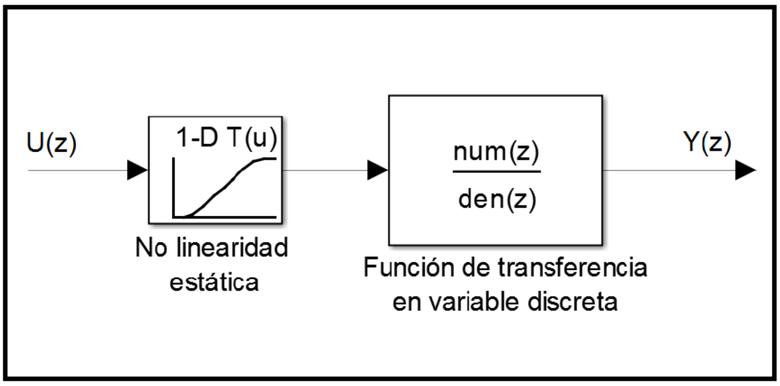
Cascade control requires more equipment and instruments that will drive up the cost of the process

Tuning cascade controllers is more difficult as the set point change + more parameters

Control in X-Y plane

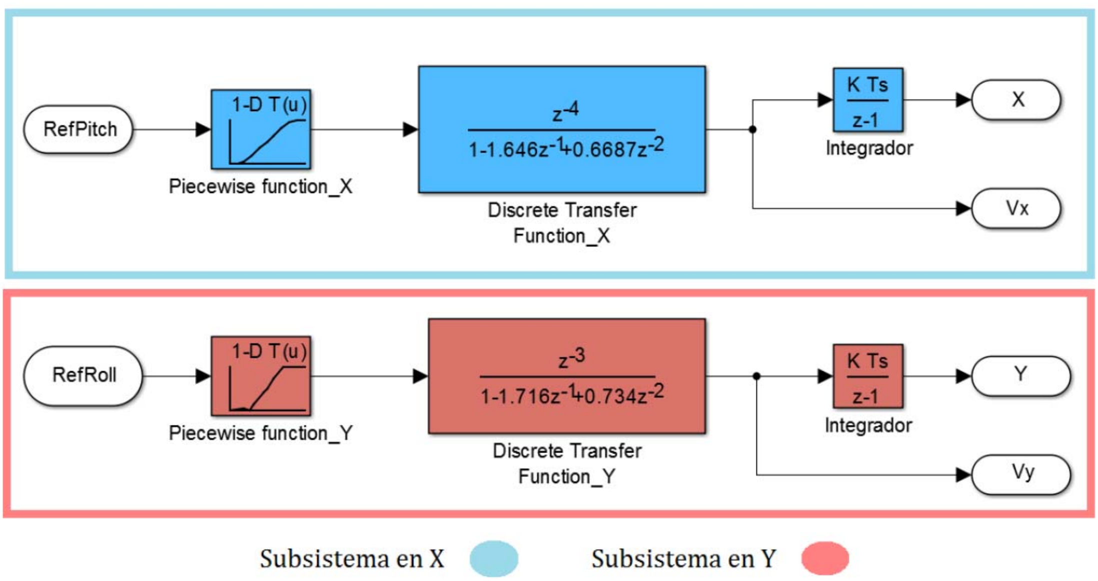
- **Modelo de caja negra**

Modelo de Hammerstein



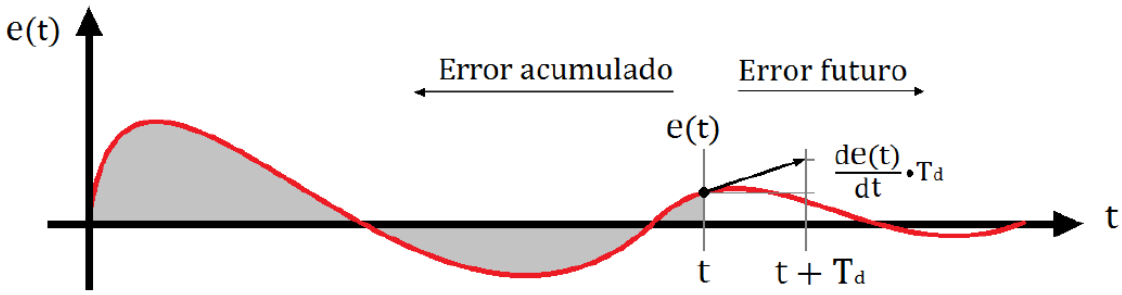
Control in X-Y plane

Modelo identificado



PID controller

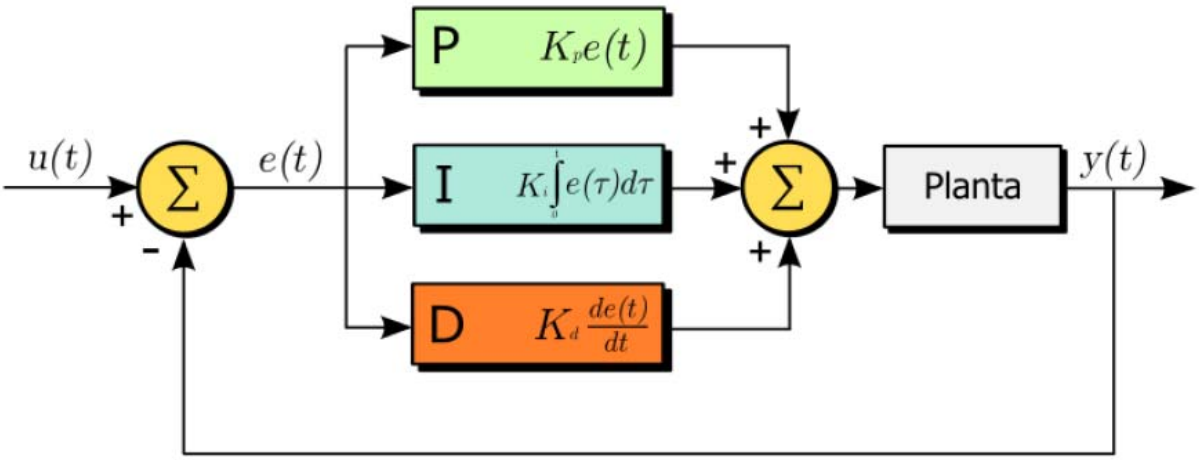
$$u(t) = K\left(e(t) + \frac{1}{T_i}\int_0^t e(t)dt + T_d\frac{de(t)}{dt}\right)$$



PID

- **Parallel implementation**

$$y(t) = Pe(t) + I\int_0^t e(t)dt + D\frac{de(t)}{dt}$$



```
evxk = RefVx - VelX;

Upvxk = Pvx*evxk;
Uivxk = Uivxk_1+Ivx*Ts*evxk_1;
Udvxk = Dvx*Nvx *(evxk- evxk_1)- Udvxk_1*(Nvx*Ts-1);

Uxk = Upvxk + Uivxk + Udvxk;

% % ANTI WIND-UP
if (Uxk>0.35)
  Uxk= 0.35;
  if (Uxk*evxk>0)
    Uivxk = Uivxk_1;
  end
end
if (Uxk<-0.35)
  Uxk= -0.35;
  if(Uxk*evxk>0)
    Uivxk = Uivxk_1;
  end
end

refPitch=Uxk;

% % ACTUALIZACIÓN DE VARIABLES
evxk_1=evxk
Uivxk_1=Uivxk;
Udvxk_1=Udvxk;
Uxk_1=Uxk
```

Matlab code for a discrete PID

$$Y = P + IT_s \frac{1}{z-1} + D \frac{N}{1 + NT_s \frac{1}{z-1}}$$

Tuning PIDs

- **By heuristic rules. Matlab Tool.**
- **By location of poles and zeros. Root locus.**
- **Method of Chien, Hrones and Reswick**
- **By ITAE criterion**

$$ITAE = \int_0^\infty t|e(t)|dt$$

$$K_P = \frac{0.965}{K} \cdot \left(\frac{\tau}{d}\right)^{0.855}, \quad T_I = \frac{\tau}{0.796 - 0.147 \cdot \left(d/\tau\right)}, \quad T_D = 0.308 \cdot \tau \left(d/\tau\right)^{0.929}$$