

---

# UDLAP<sup>®</sup>

**Departamento de Computación, electrónica y mecatrónica**



**Dr. César Martínez Torres**  
[cesar.martinez@udlap.mx](mailto:cesar.martinez@udlap.mx)

**Hacia un laboratorio virtual para Vehículos aéreos no tripulados**

---

## Objetivo

---

Crear un laboratorio virtual para probar algoritmos de control

- ▶ Multiplataforma.
- ▶ Uso educacional.
- ▶ Basado en bloques.
- ▶ Simular ambientes reales.
- ▶ Probar diferentes arquitecturas.
- ▶ Proceso de instalación simple.

- ▶ Open source desde Diciembre 2018. (Soporte de paga)
- ▶ Simulación realista de sensores y actuadores.
- ▶ Simulación de ambientes.
- ▶ Importación de CAD.
- ▶ Importación desde Google maps y open street maps.
- ▶ Simulación 3D
- ▶ Lenguaje de programación (C,C++,java,python y Matlab).
- ▶ Uso de Nodos y APIS.

# Cuadri-rotor



$$\ddot{x} = \frac{1}{m} (\cos\psi \sin\theta \cos\phi + \sin\psi \sin\phi) U_1$$

$$\ddot{y} = \frac{1}{m} (\sin\psi \sin\theta \cos\phi - \cos\psi \sin\phi) U_1$$

$$\ddot{z} = -g + \frac{1}{m} (\cos\theta \cos\phi) U_1$$

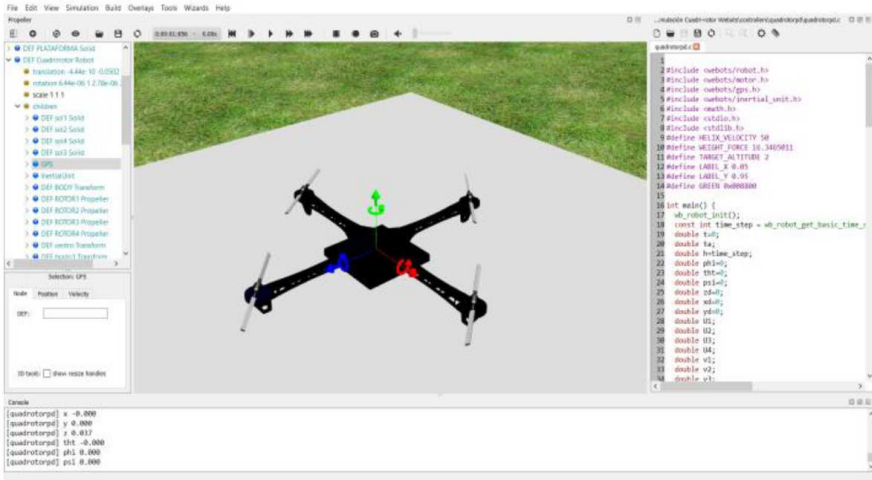
$$\ddot{\phi} = \frac{(I_y - I_z)}{I_x} \dot{\psi} \dot{\theta} - \frac{J_R \Omega}{I_x} \dot{\theta} + \frac{l}{I_x} U_2$$

$$\ddot{\theta} = \frac{(I_z - I_x)}{I_y} \dot{\psi} \dot{\phi} + \frac{J_R \Omega}{I_y} \dot{\phi} + \frac{l}{I_y} U_3$$

$$\ddot{\psi} = \frac{(I_x - I_y)}{I_z} \dot{\phi} \dot{\theta} + \frac{1}{I_z} U_4$$

# Robot

- Creación de un nodo robot
- Conectar el nodo robot a los nodos propeller, IMU, GPS etc..



The screenshot displays a ROS simulation environment. The central 3D view shows a black quadrotor robot with four propellers, positioned on a grey ground plane. The robot's body is marked with colored dots: blue on the front-left, red on the front-right, and green on the back. A green arrow points upwards from the robot's center, indicating its vertical axis. The background is a green field under a clear sky.

On the left side, a tree view shows the robot's structure, including components like 'DEF PLATAFORMA Serial', 'DEF Coardmotor Robot', 'translation', 'rotations', 'SCALE', 'children', 'Vector3int', 'DEF BODY Swacharm', 'DEF ROTOR1 Propeller', 'DEF ROTOR2 Propeller', 'DEF ROTOR3 Propeller', 'DEF ROTOR4 Propeller', 'DEF motor Swacharm', and 'DEF rotor1 Transforn'. Below this, a 'Selecciona GPS' window is visible with a 'Node' field set to 'GPS' and a 'Position' field with a velocity input.

At the bottom left, a console window shows the following output:

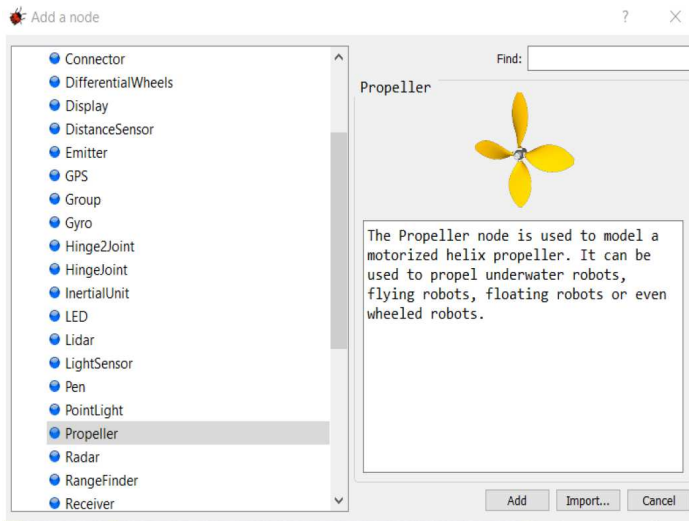
```
[quadrotorpd] x -0.000
[quadrotorpd] y 0.000
[quadrotorpd] z 0.017
[quadrotorpd] tht -0.000
[quadrotorpd] phi 0.000
[quadrotorpd] psi 0.000
```

On the right side, a text editor window shows the C++ code for the robot node:

```
1
2#include <uwbots/robot.h>
3#include <uwbots/motor.h>
4#include <uwbots/gps.h>
5#include <uwbots/inertial_unit.h>
6#include <uwbots.h>
7#include <stdlib.h>
8#include <stdio.h>
9#define HELIX_VELOCITY 50
10#define WEIGHT_FORCE 16.3409011
11#define TARGET_ALTITUDE 2
12#define LABEL_X 0.05
13#define LABEL_Y 0.05
14#define GREEN 0x000000
15
16int main() {
17    wh_robot_init();
18    const int time_step = wh_robot_get_basisc_time_s;
19    double t=0;
20    double ta;
21    double h=time_step;
22    double phi=0;
23    double tht=0;
24    double psi=0;
25    double z=0;
26    double xd=0;
27    double yd=0;
28    double U1;
29    double U2;
30    double U3;
31    double U4;
32    double v1;
33    double v2;
34    double v1;
35}
```

## Robot

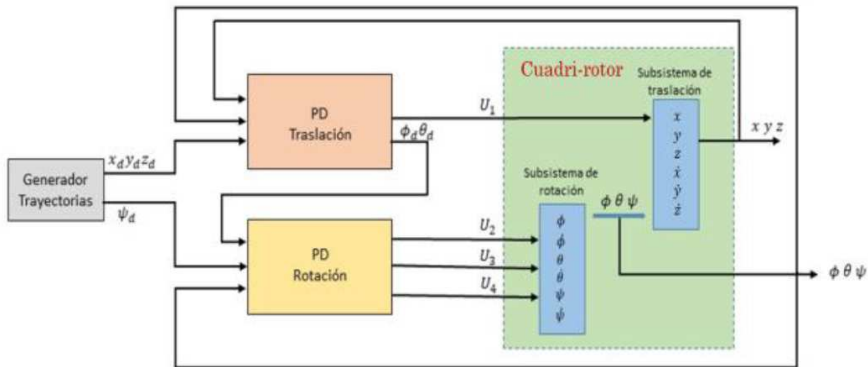
Cada nodo children tiene características específicas, que pueden ser obtenidas por experimentación o por datos de fabricante.



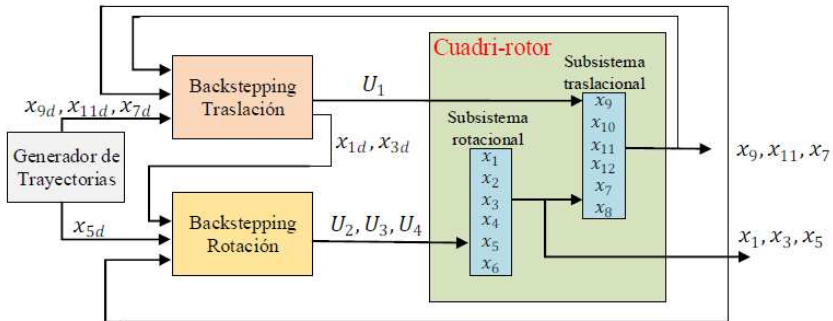
- ▶ Control PD.
  
  
  
  
  
  
  
  
  
  
- ▶ Control Backstepping.

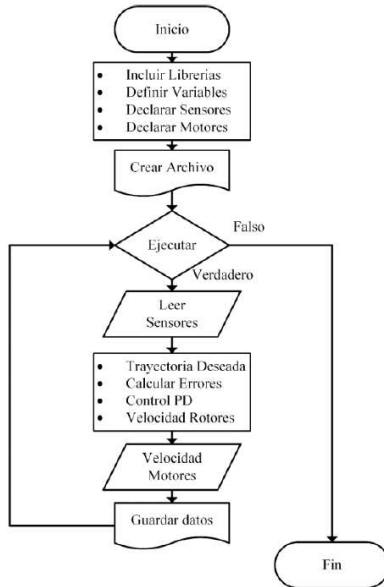


# Controlador PID

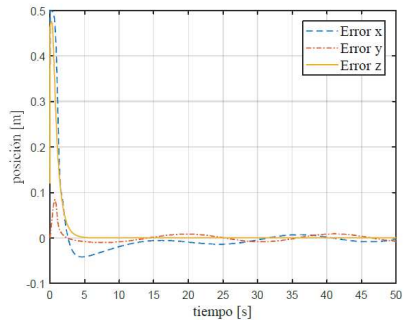
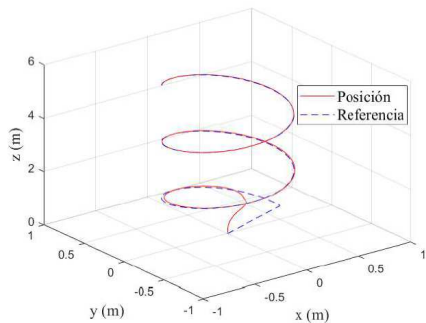


# Backstepping





# Resultados



---

Gracias por su atención