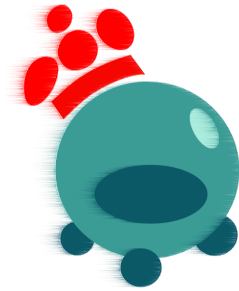


Programación de drones



José María Cañas
josemaria.plaza@urjc.es

26 de octubre de 2017

Robótica



- *robot = hardware + software*
- *hardware = sensores + actuadores + computadoras*



Hardware de robots aéreos

- IMU, giróscopos
- GPS
- Cámaras
- LIDAR...
- Motores de hélices
- Computadores a bordo
- Placa estabilizadora
- Comunicaciones: radio, Wifi...





Software para robots

- La **inteligencia** de los robots radica en su **software**.
- $sw = middleware + aplicaciones$

Middleware

- $middleware = drivers + arquitecturaSW + herramientas + bibliotecas$
- arquitectura SW distribuida
- ROS, ICE, DDS...
- OpenCV, PCL, OMPL, nodos ya existentes...
- Simuladores

Aplicaciones

- *Aplicacion = percepcion + decisiones*
- Percepción: elaborar información, sensores
- Decisiones: control, planificación, actuación
- **Algoritmos**
- Lenguajes texto: C++, Python... y visuales

```
84 int segmentationType; //0 ninguna, 1 NITE
85 int mainFPS;
86
87
88 void* updateThread(void*)
89 {
90     //glfwInit();
91
92     libfreenect2::Freenect2 freenect2;
93     libfreenect2::Freenect2Device *dev = freenect2.openDefaultDevice();
94
95     if(dev == 0)
96     {
97         std::cout << "no device connected or failure opening the default one
98         return (NULL);
99     }
100     libfreenect2::SyncMultiFrameListener listener(libfreenect2::Frame::Color | 1:
101     libfreenect2::FrameMap frames;
102
103     dev->setColorFrameListener(&listener);
104     dev->setIrAndDepthFrameListener(&listener);
105     dev->start();
106
107     std::cout << "Device serial: " << dev->getSerialNumber() << std::endl;
108     std::cout << "Device firmware: " << dev->getFirmwareVersion() << std::endl;
109
110     float cycle=(float)(1/(float)mainFPS)*1000000;
```

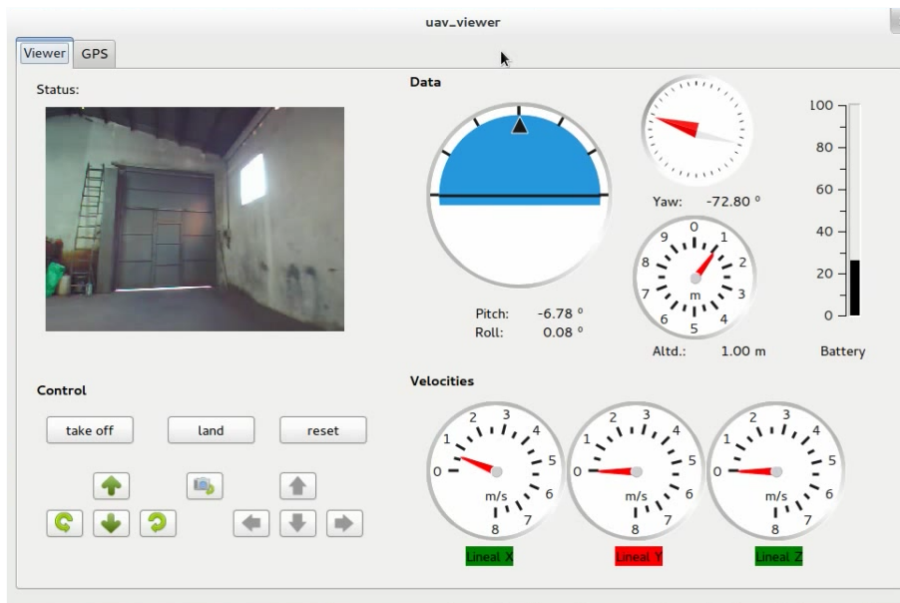


Software de robots aéreos

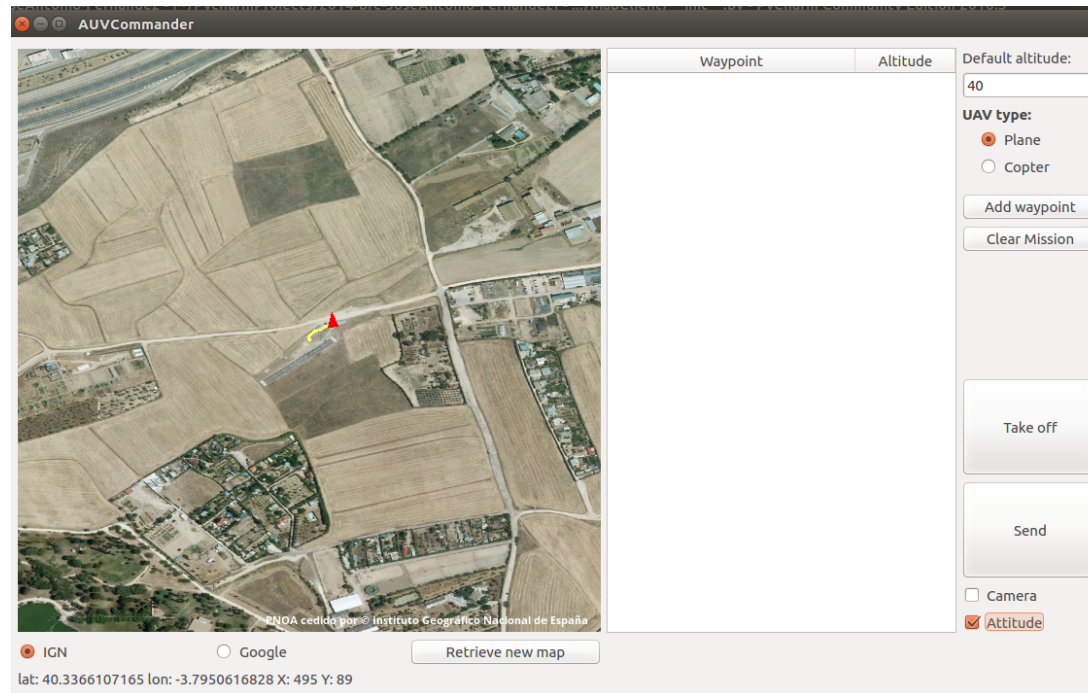
- Permite ir más allá que simples teleoperadores
- Control total del drone
- ArduPilot, software libre
- Controladora Pixhawk
- MAVLink, protocolo para comunicación con tierra
- Middleware robótico ROS
- Middleware robótico JdeRobot (<http://jderobot.org>)

Middleware para programar drones

Herramienta UAVViewer



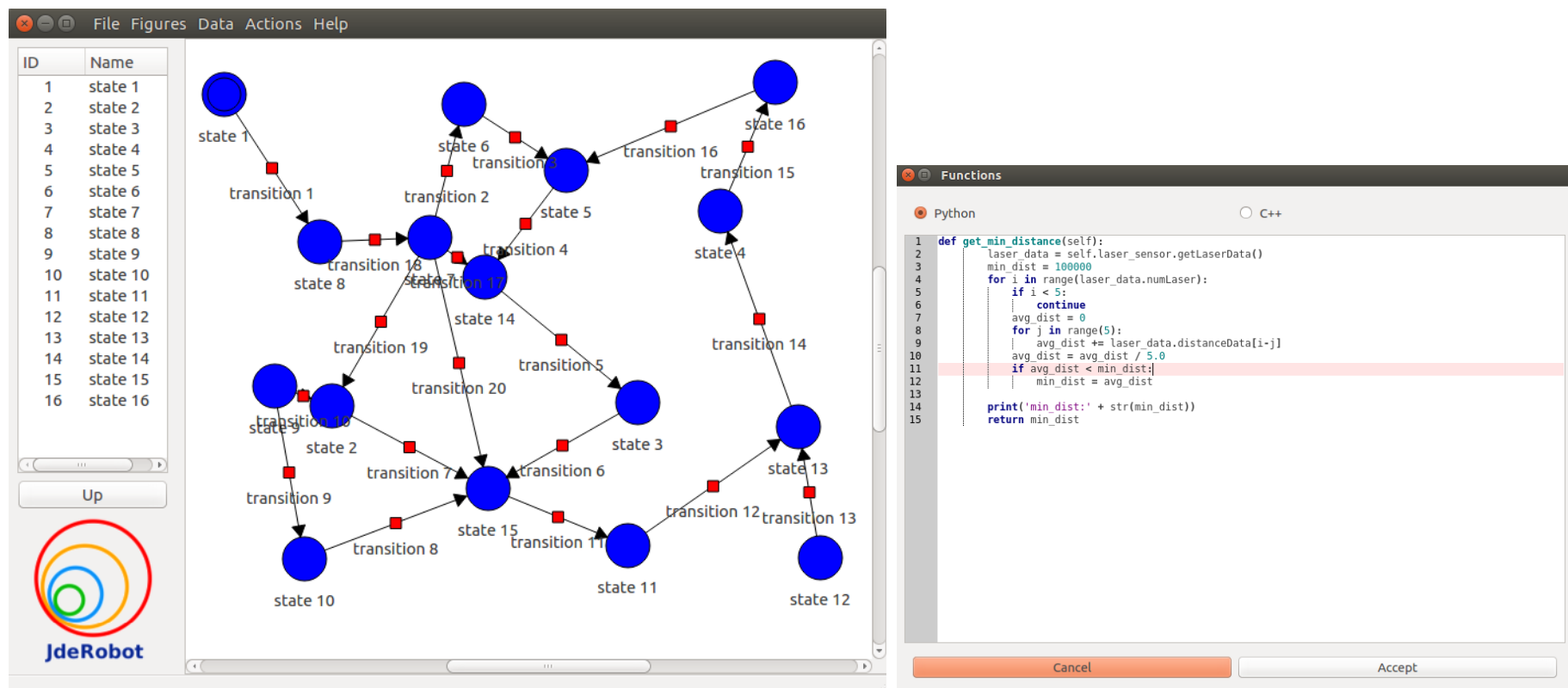
Herramienta UAVCommander



- Posición actual en el mapa
- Secuencias de puntos de paso

Herramienta VisualStates

- Edición visual: estados y transiciones
- Ventanas de texto: código de estados, de transiciones y auxiliar



The screenshot displays the VisualStates tool interface. On the left, a table lists 16 states:

ID	Name
1	state 1
2	state 2
3	state 3
4	state 4
5	state 5
6	state 6
7	state 7
8	state 8
9	state 9
10	state 10
11	state 11
12	state 12
13	state 13
14	state 14
15	state 15
16	state 16

The main area shows a state transition diagram with 16 blue circular nodes (states) and 20 red square nodes (transitions) connected by arrows. The transitions are labeled from transition 1 to transition 20. The diagram shows a complex network of states and transitions, with some states having multiple outgoing and incoming transitions.

On the right, a code editor window titled "Functions" shows Python code for a method named `get_min_distance`:

```

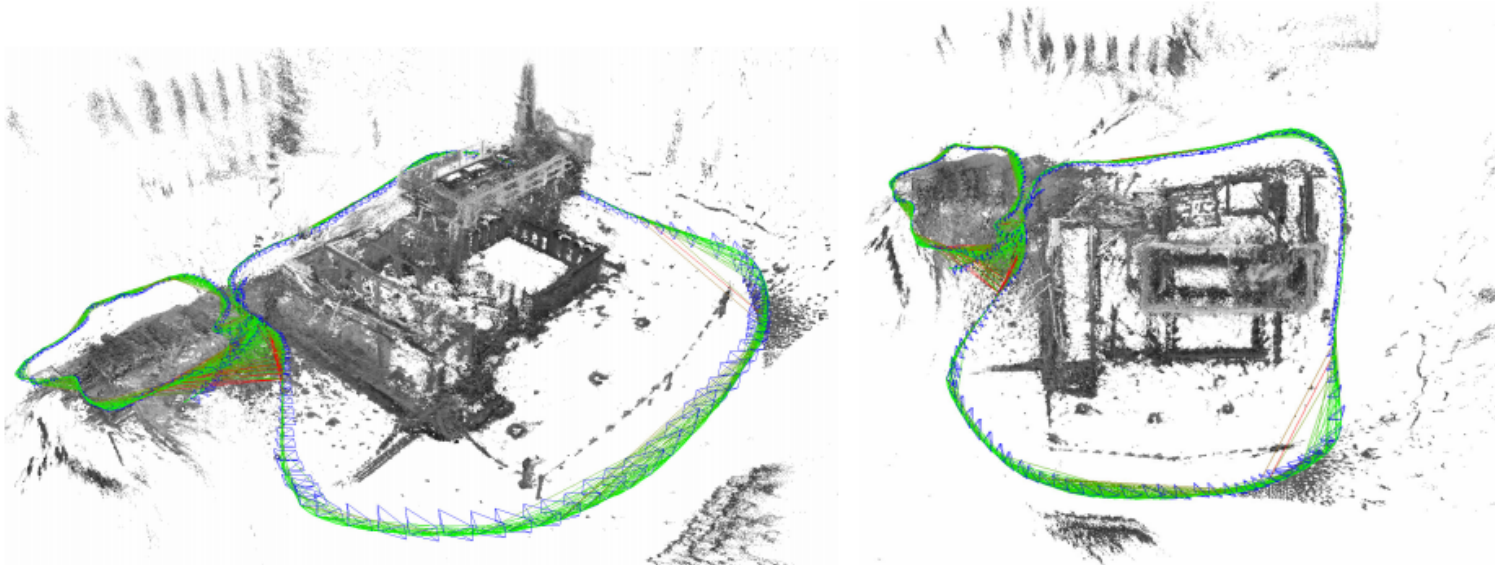
1 def get_min_distance(self):
2     laser_data = self.laser_sensor.getLaserData()
3     min_dist = 100000
4     for i in range(laser_data.numLaser):
5         if i < 5:
6             continue
7         avg_dist = 0
8         for j in range(5):
9             avg_dist += laser_data.distanceData[i-j]
10            avg_dist = avg_dist / 5.0
11            if avg_dist < min_dist:
12                min_dist = avg_dist
13
14 print('min_dist:' + str(min_dist))
15 return min_dist

```

At the bottom of the code editor, there are "Cancel" and "Accept" buttons.



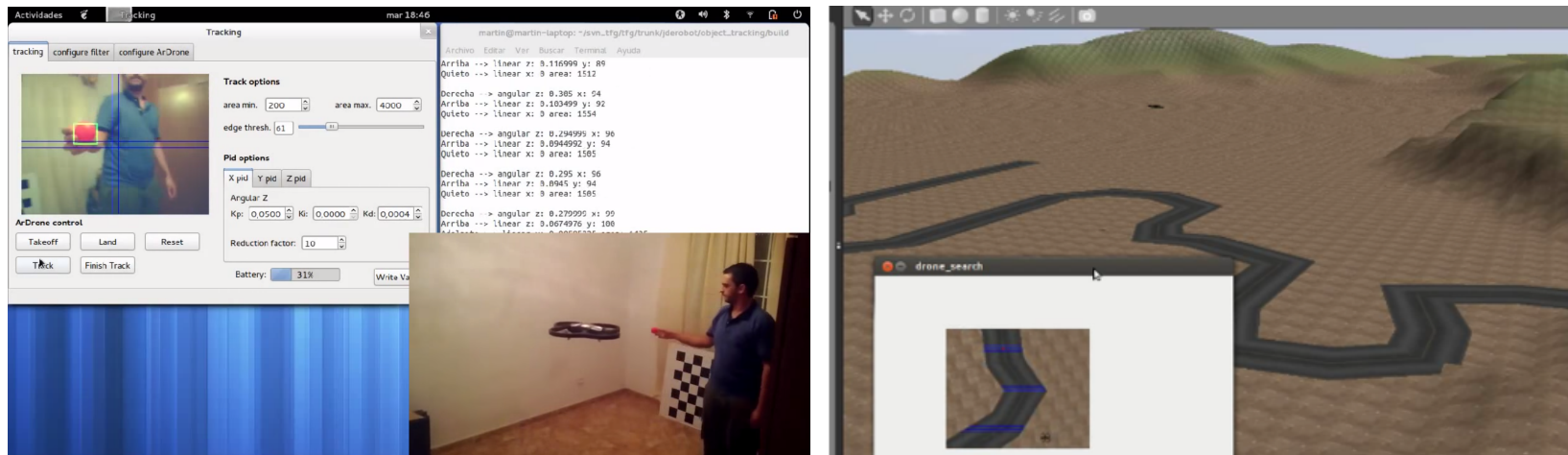
Autolocalización visual



- LSDSLAM, SVO...

Aplicaciones con drones

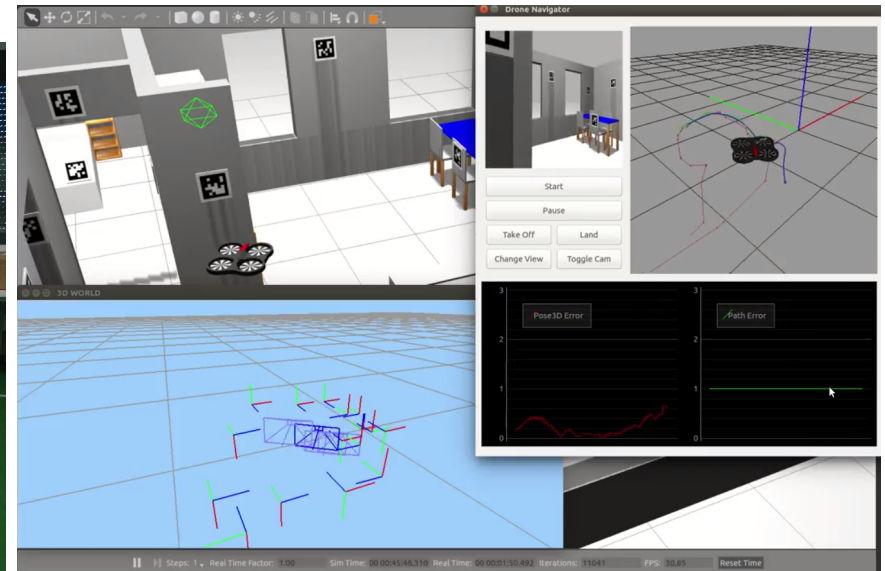
Seguimiento: de objetos, de carretera...



- Detección del objeto
- Control visual reactivo

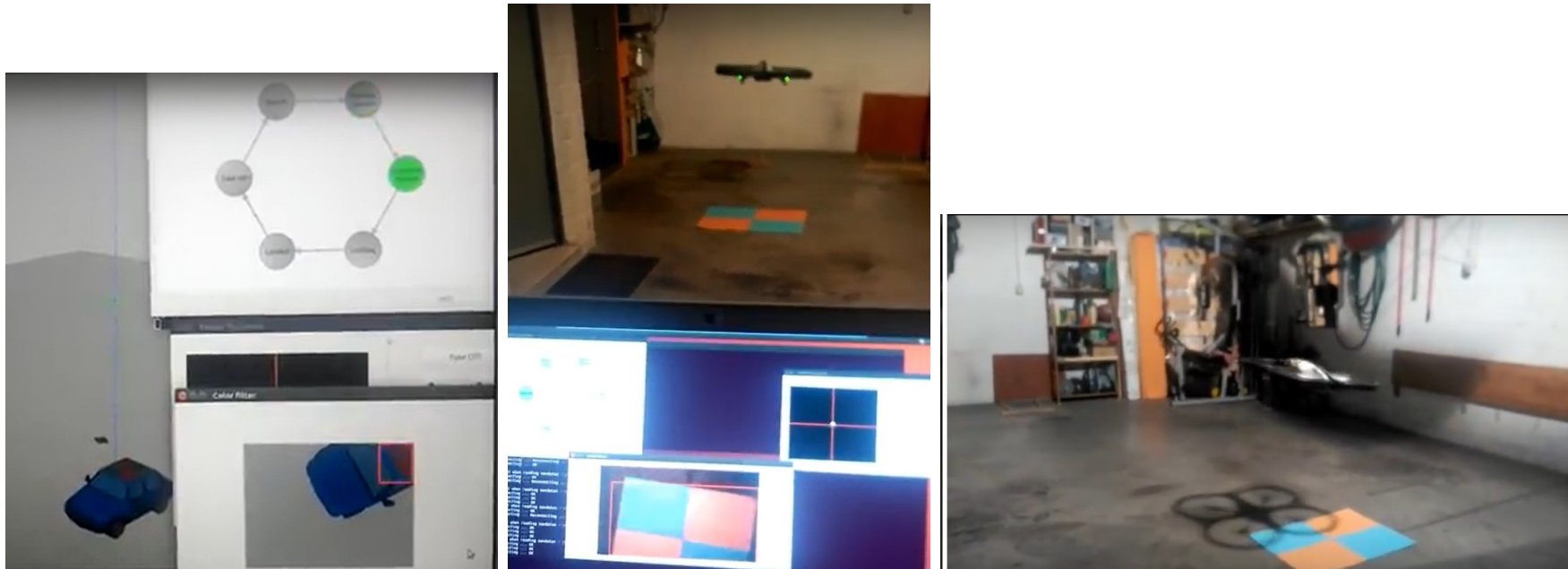


Recorrido de ruta 3D en interiores



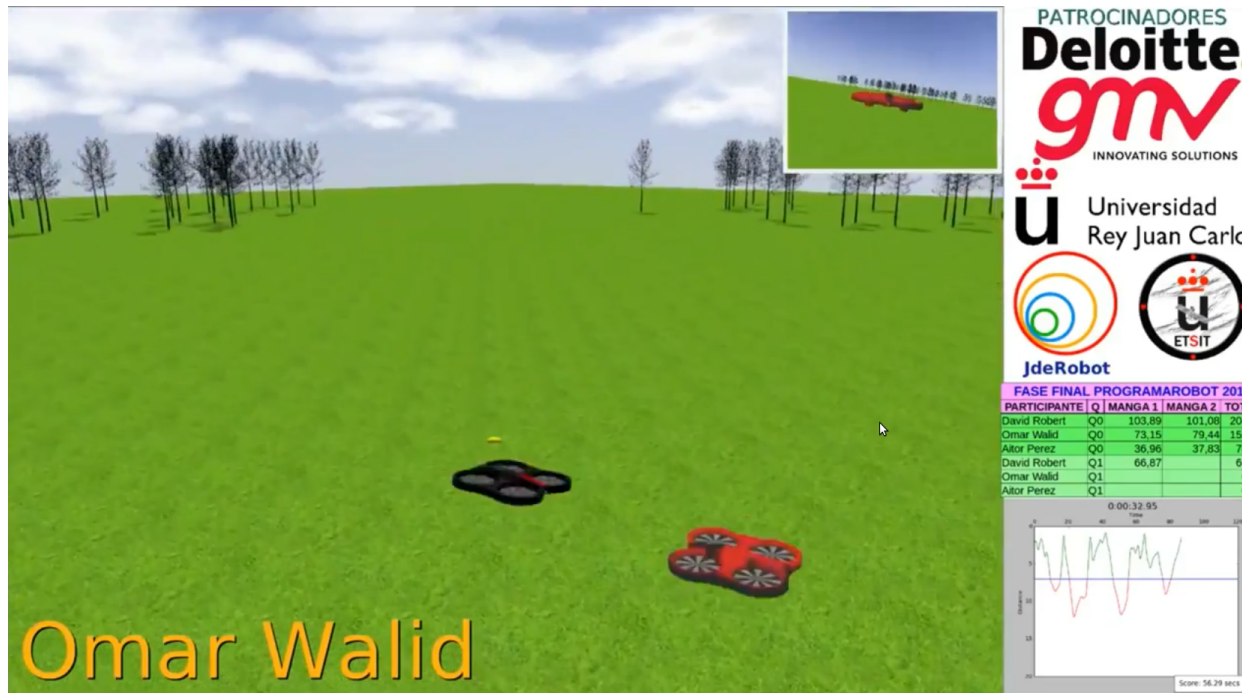
- Autocalización visual basada en baliza
- Piloto controla desviaciones de ruta

Aterrizaje preciso



- Detección visual de la baliza
- Autómata de estados: despegue, búsqueda y aterrizaje

Campeonato Program-A-Robot



Omar Walid

PATROCINADORES
Deloitte
gmv
 INNOVATING SOLUTIONS

Universidad Rey Juan Carlos

JdeRobot

FASE FINAL PROGRAMAROBOT 2017				
PARTICIPANTE	Q	MANGA 1	MANGA 2	TOTAL
David Robert	Q0	103.89	101.08	204.97
Omar Walid	Q0	73.15	79.44	152.59
Altor Perez	Q0	36.96	37.83	74.79
David Robert	Q1	66.87		66.87
Omar Walid	Q1			0,00
Altor Perez	Q1			0,00

0:00:32.95

Score: 56.79 secs

- Análisis de imagen, filtro de color
- Control reactivo
- <http://jderobot.org/Campeonato-programacion-de-robots>