UNIVERSIDAD
REY JUAN CARLOS

TITULACIóN EN MAYúSCULAS

Curso Académico 2020/2021

Trabajo Fin de Carrera/Grado/Máster

TíTULO DEL TRABAJO EN MAYúSCULAS

Autor : Nombre del Alumno

Tutor : Dr. Gregorio Robles

Proyecto Fin de Carrera

FIXME: Título

Autor : FIXME

Tutor : Dr. Gregorio Robles Martínez

La defensa del presente Proyecto Fin de Carrera se realizó el día de de 20XX, siendo calificada por el siguiente tribunal:

Presidente:

Secretario:

Vocal:

y habiendo obtenido la siguiente calificación:

Calificación:

Fuenlabrada, a de de 20XX

Dedicado a

mi familia / mi abuelo / mi abuela

II

Agradecimientos

Aquí vienen los agradecimientos... Aunque está bien acordarse de la pareja, no hay que olvidarse de dar las gracias a tu madre, que aunque a veces no lo parezca disfrutará tanto de tus logros como tú... Además, la pareja quizás no sea para siempre, pero tu madre sí.

Resumen

Aquí viene un resumen del proyecto. Ha de constar de tres o cuatro párrafos, donde se presente de manera clara y concisa de qué va el proyecto. Han de quedar respondidas las siguientes preguntas:

- ¿De qué va este proyecto? ¿Cuál es su objetivo principal?
- ¿Cómo se ha realizado? ¿Qué tecnologías están involucradas?
- ¿En qué contexto se ha realizado el proyecto? ¿Es un proyecto dentro de un marco general?

Lo mejor es escribir el resumen al final.

Summary

Here comes a translation of the “Resumen” into English. Please, double check it for correct grammar and spelling. As it is the translation of the “Resumen”, which is supposed to be written at the end, this as well should be filled out just before submitting.

Índice general

Índice de figuras

Capítulo 1

Introducción

Los UAV o Drones se han popularizado en los últimos años hasta es punto de formar parte de nuestro día a día con aplicaciones en muchos ambitos de nuestra vida.

Si bien se están utilizando ya de forma habitual en sectores como el cine o la ingeniería civil, aún se están explorando muchas de las posibles utilidades que estos robots pueden llegar a ofrecer.

El objetivo de este trabajo final es poner en valor y asentar el uso de un tipo de UAV que no está hoy muy representado en el ámbito civil y que aventaja en varios aspectos al mas popularizado quadracóptero, se trata del avión.

1.1. Tipos de aeronaves

Las aeronaves son la base sobre las que se asienta la inteligencia que permite que nuestro robot vuele de ahí que convenga dedicar unas líneas a entender la base de las mismas y en particular las que son objeto de estudio y desarrollo en este PFC los llamados aerodinos.

Existen principalmente 2 tipos de aeronaves si atendemos al modo en que generan su sustentación con sus alas, de ala fija y las de ala rotatoria.

Dentro de la tipificación de ala fija tenemos aquellas aeronaves que tienes sus alas fijas al fuselaje. Según la definición de la OACI, es un «Aerodino propulsado por motor, que debe su sustentación en vuelo principalmente a reacciones aerodinámicas ejercidas sobre superficies que permanecen fijas en determinadas condiciones de vuelo» Algunos ejemplos de aeronaves de ala fija son los aeroplanos, planeadores/veleros, aladeltas, parapentes, paramotores y ultraligeros.

Este tipo de aerodinos los conocemos mas comúnmente como avienes y tienen como principal ventaja de que la carga de aire que necesitan en sus alas puede ser producida de muchas formas distinta (los veleros no tienen ningún tipo de propulsión). Esta carga es variable en función de la superficie alar del mismo y permite por tanto cargas mas grandes que si instalásemos el mismo propulsor en un ala rotatoria. Pongamos como ejemplo el A380 de Airbus, es el avión de pasajeros mas grande del mundo y cuenta con 4 motores que producen un empuje de entre 70.000 y 80.000lbs, unas 32-36 toneladas de empuje cada uno generando por tanto entre los 4 a máximo rendimiento y optimas condiciones alrededor de 144 toneladas de empuje. Este avión tiene un peso máximo al despegue[1] de entre 560 y 590 toneladas. Tenemos por tanto que necesitamos en este caso ¼ del peso total en empuje para despegar este avión. Si hiciésemos este mismo ejercicio con un aerodino de ala rotatoria como el Boing AH-64 o Apache con un peso máximo al despegue de 9,5 toneladas necesitaríamos que la combinación que realizan empuje y palas superase esos 9,5 toneladas para siguiera levantar del suelo. Este tipo de aerodinos son por tanto mas eficientes, rápidos, con mayor carga de pago, mayor alcance debido a su menor consumo y mas estables.

Dentro de la tipificación de ala rotatoria tenemos aquellas aeronaves que producen su sustentación con el movimiento (rotación) de sus alas. En este tipo de aeronaves las alas, también llamadas "palas.[en] este tipo de aerodinos, giran en torno a un eje produciendo con este giro la sustentación necesaria para despegar del suelo. Algunos ejemplos de este tipo de aeronaves son los helicópteros, autogiros, convertibles o los ampliamente conocidos en robotica aerea los quadracópteros. Este tipo de aerodino tiene como principal ventaja frente a los ala fija en su versatilidad a la hora de realizar las maniobras de despegue y aterrizaje que pueden realizarse de forma vertical (VTOL[2]) además de la capacidad de realizar vuelo estacionario[3] que le hacen imprescindible en escenarios poco accesibles o donde nos es posible aterrizar como el rescate marítimo.

[1] Peso máximo que es capaz de soportar un avión en su maniobra de despegue

[2] Vertical take off and landing

[3] Mantenerse estáticamente en un punto elevado

1.2. Origenes

Los orígenes de la robótica aerea tienen origen militar y su avance ha estado intrínsecamente ligado a este ámbito durante todo el siglo XX.

Se consideran el origen de los aviones no tripulados los experimentos llevados a cabo a principios del siglo XX durante la 1ª guerra mundial como el .ªerial Target"desarrollado por el capitán A. H. Lowpara para su uso como blanco aéreo. Si bien eran vehículos no tripulados (Unmaned Aereal Vehicles) no eran autónomos y eran manejados desde tierra a través de una radio. No es hasta el final del siglo XX cuando bajo el escenario de la guerra de Vietnam y ante la creciente perdida de vidas de los pilotos estos vehículos vuelvan de nuevo a ser objeto de desarrollo y se conviertan en vehículos autónomos.

Desde ese momento y hasta nuestros días se utilizan de forma habitual en el ámbito militar en misiones de reconocimiento, bombardeos o apoyo sin arriesgar vidas humanas.

A los largo de los primeros años de este siglo debido al abaratamiento de los componentes electrónicos y a su minituarización y potencia, la robótica aérea se ha "desmilitarizadoz esta experimentado un enorme crecimiento en el ámbito de las aplicaciones civiles.

Hoy en día es común encontrar en cualquier juguetería quadracópetros radio-pilotados por poco menos de 30 euros y en tiendas especializadas podemos encontrarlos ya con el hardware y software integrados que les permiten seguir una serie de puntos de control y comportarse de forma autónoma por poco mas de 200?.

Por ello se ha popularizado su uso en aplicaciones civiles como: Fotografía aérea. Para por ejemplo usos topográficos o educativos e incluso recaudatorios[4] Cine y televisión. Hoy en día es raro encontrar una producción que no haya hecho uso de ellos y es que permite la captura de tomas que de otra forma serían extremadamente complicadas o imposibles o bien factibles pero económicamente inviables. Vigilancia y protección. Las principales fuerzas y cuerpos de seguridad de una gran cantidad de países los utilizan.

Se esta experimentando su uso en otros ámbitos como el logístico[5], sanitario o salvamento.

[4] El ministerio de hacienda español tiene varios drones para cotejar los datos catastrales con la vivienda física

[5] Amazon plantea utilizarlo en algunas entregas

1.3. Estado del arte

En la actualidad el uso de AUV o drones se ha popularizado tanto que es una de las industrias en las que mas ha crecido su inversión, y es que según la empresa analista especializada en dones Droneii con sede en Hamburgo en un estudio sobre la inversión en el sector[6] en europa se invirtió en en proyectos domésticos en 2016 cerca de 65 millones de dólares incrementándose esta cifra hasta los 314 millones si atendemos al mercado norteamericano. Estas datos se asientan en un mercado cada vez mas extendido y con una gran proyección de crecimiento,la publicación BI Intelligence[7] espera que las ventas de drones alcancen los 12.000 millones en 2021.

Con su uso ya ampliamente extendido en sectores como el cine, la televisión, fotografía, agropecuario, educativo, forestal, ingeniería civil y presencia en sectores como el sanitario, salvamento o seguridad y protección el nicho de mercado de los UAV esta lejos de su cima y se investigan dia a dia nuevos usos en sectores como la logística.

Dentro del laboratorio de robótica de la Universidad Rey Juan Carlos cabe destacar los trabajos realizados para profundizar investigar y experimentar con ellos. Trabajos como el de Alberto Martínez Florido quien desarrollo un driver para poder utilizar desde jderobot el ar-drone comercial, desarrollado por la empresa francesa parrot, y el software de control UAV Viewer. Y mas adelante Daniel Yague que realizó un driver para utilizar el ar-drone de forma simulada en gazebo. O los trabajos de Jorge Cano quien construyo propio su drone utilizando como base un quadracóptero y dando soporte a jderobot de muchas de las intruciones de MAVLink. Son destacables también los trabajos que se están llevando a cabo por Diego Jimenez adaptando el interfaz SoloDrone de la empresa 3DR[8] o Jorge Vela quien se encuentra desarrollando como realizar la maniobra de aterrizaje de forma automática.

1.4. Objetivos

Los objetivos de este PFC son arrojar luz sobre el uso de drones de ala fija poniendo en valor éstos en situaciones donde la carga de pago, la autonomía, velocidad o consumo desaconsejan

[6] http://www.droneii.com/drone-investment-trends-2016

[7] http://www.businessinsider.com/the-drones-report-research-use-cases-regulations-and-

[8] Empresa norteamericana con sede en California especializada en robótica aérea. Se sitúa en 2017 como la 3ª empresa del sector

el uso de los mas extendidos quadracópteros.

Para ésto hemos adaptado un avión de radio-control comercial añadiéndole toda la aviónica necesaria para que puede operar como drone. Hemos desarrollado un driver para jderobot del interfaz MAVLink que nos permite recoger actitud, velocidades lineales y angulares, altura, posición y otros parámetros como la batería restante. Nos permite también la actuación con el seguimiento de misiones que no sólo incluyen el paso por ciertos puntos de control o waypoints sino que también soporta las maniobras de despegue y aterrizaje en drones de ala fija. Hemos desarrollado también un software de control que da soporte a este tipo de actuación de mas alto nivel, las misiones.

1.5. Estructura de la memoria

En esta sección se debería introducir la esctura de la memoria. Así:

- En el primer capítulo se hace una intro al proyecto.
- En el capítulo **??** se muestran los objetivos del proyecto.
- A continuación se presenta el estado del arte.
- ...

Capítulo 2

Infraestructura utilizada

2.1. Hardware

Este PFC se apoya principalmente en 3 piezas hardware:

- El avión a radio-control. Para este desarrollo hemos elegido el avión Bix3 distribuido por la empresa china `www.hobbyking.com`, hemos elegido este modelo por ser un avión muy estable debido principalmente a sus cualidades como velero y su alta superficie alar. Para poder cargar con el equipo necesario nos ha sido necesario cambiar el motor, el variador y las baterías de serie por otras de mayor rendimiento que nos permiten volar con tanta carga de pago.
- El estabilizador/piloto automático, para este desarrollo hemos optado por un Ardupilot que trae incoorporados en su placa los giróscopos y acerelómetros para su estabilización y que trae de serie un receptor GPS con brújula.
- Una raspberry PI3 que es nuestro ordenador de abordo y en el que instalaremos la infraestructura necesaria para que podamos dotarle de inteligencia. Hemos elegido éste dispositivo debido al compromiso peso/potencia que nos otorga así como porque se trata de un hardware muy asequible y extendido.

2.2. Software

labelsec:software

En este apartada trataremos las capas de software sobre las que nuestro desarrollo se apoya para lograr nuestros objetivos, vamos a agrupar las mismas en los dispositivos hardware sobre las que se apoyan o el las que estarán instaladas.

1. Placa estabilizadora/piloto automático Ardupilot. Esta placa accede directamente a los actuadores del avión (los servos y el motor) a través de señales PWM. Esta y otras placas similares como Pikhawnk ofrecen un interfaz que se apoya en comandos llamado MAVLink. A través de comandos MAVLink se puede acceder a los sensores, a los que incoorpora la placa o a los que le añadamos a la misma como el GPS o sensores de velocidad del aire. A través de estos comandos se le puede también enviar ordenes al piloto automático quien las ejecutará. Un ejemplo de comando MAVLink sería:

```
type GpsStatus struct {
    SatellitesVisible  uint8      Número de satélites visibles
    SatellitePrn       [20]uint8  Id Global de cada satélite
    SatelliteUsed      [20]uint8  Lista con el uso de cada satélite 0: no se
    SatelliteElevation [20]uint8  Elevación, nos da el ángulo sobre el horiz
    SatelliteAzimuth   [20]uint8  Dirección del satélite, 0: 0 grados, 255:
    SatelliteSnr       [20]uint8  Señal/ruido de cada uno de los satélites
}
```

Este mensaje trae la información del enlace actual con el GPS y se envía periódicamente en ciclos que decidimos en parámetros de conexión con el dispositivo. Otro parámetro, esta vez vinculado a la actuación sería:

```
type MissionItem struct {
    Param1          float32    parámetro variable en función del comando.
    Param2          float32    parámetro variable en función del comando.
    Param3          float32    parámetro variable en función del comando.
    Param4          float32    parámetro variable en función del comando.
    X               float32    latitud
    Y               float32    longitud
    Z               float32    altitud
    Seq             uint16     Número del item en la misión
    Command         uint16     Tipo de comando de navegación por ejemplo 16
```

```
    TargetSystem    uint8      ID del sistema
    TargetComponent uint8
    Frame           uint8      Sistema de coordenadas que se utiliza
    Current         uint8      Misión actual no:0, si:1
    Autocontinue    uint8      Autocontinuar al siguiente objeto de misión t
}
```

2. Raspberry PI. Dentro del dispositivo instalaremos la distribución linux Ubuntu en su versión 14.04 sobre la que instalaremos el framework Jderobot. Jderobot es un framework desarollado por el departamento de robótica de la Universidad rey Juan Carlos que nos expone un interfaz sobre el que recoger/servir datos recogidos por sensores así como de actuar sobre los actuadores del robot. Se apoya en el interfaz de ZeroC Ice[1] para la distribución es estos interfaces aislando por tanto el lenguaje de programación utilizado pudiendo utilizar el lenguaje mas apropiado para cada tarea. Expone varios interfaces, pero en este capítulo explicaremos los que durante nuestro desarrollo hemos implementado:

   - Pose3D. Utilizado para recoger los datos de actitud y la posición de la aeronave.

```
Pose3DData
  {
float x;  /* x coord */
float y;  /* y coord */
float z;  /* z coord */
   float h;  /* */
float q0; /* qw */
float q1; /* qx */
float q2; /* qy */
float q3; /* qz */
  };
```

   - Camera. Utilizado para servir imágenes.

```
  class CameraDescription
```

[1] algo

```
 {
   string name;
   string shortDescription;
   string streamingUri;
   float fdistx;
   float fdisty;
   float u0;
   float v0;
   float skew;
   float posx;
   float posy;
   float posz;
   float foax;
   float foay;
   float foaz;
   float roll;
 };
```

- NavData. Utilizado para servir datos secundarios de actuación como velocidades lineales o angulares o el estado de la batería.

```
class NavdataData
{
 int vehicle; //0-> ArDrone1, 1-> ArDrone2
 int state; // landed, flying,...
 float batteryPercent; //The remaing charge of baterry %

 //Magnetometer ArDrone 2.0
 int magX;
 int magY;
 int magZ;
```

```
int pressure; //Barometer ArDrone 2.0
int temp;     //Temperature sensor ArDrone 2.0
float windSpeed; //Estimated wind speed ArDrone 2.0

float windAngle;
float windCompAngle;

float rotX; //rotation about the X axis
float rotY; //rotation about the Y axis
float rotZ; //rotation about the Z axis

int altd; //Estimated altitude (mm)

//linear velocities (mm/sec)
float vx;
float vy;
float vz;

//linear accelerations (unit: g) ¿ArDrone 2.0?
float ax;
float ay;
float az;

//Tags in Vision Detectoion
//Should be unsigned
int tagsCount;
arrayInt tagsType;
arrayInt tagsXc;
arrayInt tagsYc;
arrayInt tagsWidth;
```

```
 arrayInt tagsHeight;
 arrayFloat tagsOrientation;
 arrayFloat tagsDistance;

 float tm; //time stamp
};
```

- Extra. Utilizado principalmente para las órdenes de despegue y aterrizaje.

```
    void land() - land drone.
    void takeoff() - takeoff drone.
    void reset()
    void recordOnUsb(bool record)
    void ledAnimation(int type,float duration, float req)
    void flightAnimation(int type, float duration)
    void flatTrim()
    void toggleCam() - switch camera.
```

- Misión. Hemos desarrollado el interfaz misión para poder dar cabida a este tipo de actución que Jderobot no soportaba hasta el momento

```
class Pose3DData  //we consumes Pose3DData
  {
float x;
float y;
float z;
   float h;
float q0;
float q1;
float q2;
float q3;
  };

  ["python:seq:list"] sequence<Pose3DData> PoseSequence;
```

```
/**
* Mission data information
*/
class MissionData
{
  PoseSequence mission;
};

/**
 * Interface to the Mission.
 */
interface Mission
{
  idempotent MissionData getMissionData();
  int setMissionData(MissionData data);
};
```

2.3. Planificación temporal

labelsec:planificacion-temporal

Capítulo 3

Estado del arte

Descripción de las tecnologías que utilizas en tu trabajo. Con dos o tres párrafos por cada tecnología, vale.

Puedes citar libros, como el de Bonabeau et al. sobre procesos estigmérgicos [1].

También existe la posibilidad de poner notas al pie de página, por ejemplo, una para indicarte que visite la página de LibreSoft[1].

3.1. Sección 1

[1] `http://www.libresoft.es`

Capítulo 4

Diseño e implementación

4.1. Arquitectura general

figura 4.1.

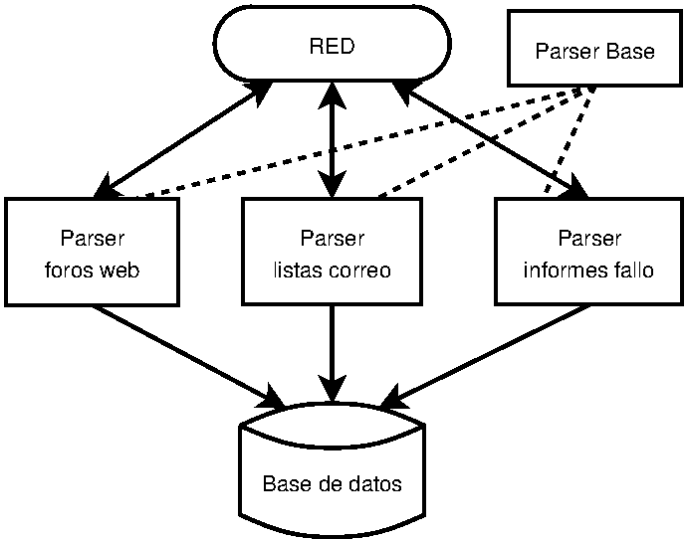

Figura 4.1: Estructura del parser básico

Capítulo 5

Resultados

Capítulo 6

Conclusiones

6.1. Consecución de objetivos

Esta sección es la sección espejo de las dos primeras del capítulo de objetivos, donde se planteaba el objetivo general y se elaboraban los específicos.

Es aquí donde hay que debatir qué se ha conseguido y qué no. Cuando algo no se ha conseguido, se ha de justificar, en términos de qué problemas se han encontrado y qué medidas se han tomado para mitigar esos problemas.

6.2. Aplicación de lo aprendido

Aquí viene lo que has aprendido durante el Grado/Máster y que has aplicado en el TFG/TFM. Una buena idea es poner las asignaturas más relacionadas y comentar en un párrafo los conocimientos y habilidades puestos en práctica.

1. a
2. b

6.3. Lecciones aprendidas

Aquí viene lo que has aprendido en el Trabajo Fin de Grado/Máster.

1. a

2. b

6.4. Trabajos futuros

Ningún software se termina, así que aquí vienen ideas y funcionalidades que estaría bien tener implementadas en el futuro.

Es un apartado que sirve para dar ideas de cara a futuros TFGs/TFMs.

6.5. Valoración personal

Finalmente (y de manera opcional), hay gente que se anima a dar su punto de vista sobre el proyecto, lo que ha aprendido, lo que le gustaría haber aprendido, las tecnologías utilizadas y demás.

Apéndice A

Manual de usuario

Bibliografía

[1] E. Bonabeau, M. Dorigo, and G. Theraulaz. *Swarm Intelligence: From Natural to Articial Systems*. Oxford University Press, Inc., 1999.