



UNIVERSITAT
POLITÀCNICA
DE VALÈNCIA



Escuela Técnica Superior de Ingeniería del Diseño



DISEÑO DE ALGORITMOS DE GUIADO PARA EL ATERRIZAJE AUTÓNOMO DE VEHÍCULOS AÉREOS NO TRIPULADOS BASADO EN VISIÓN ARTIFICIAL.

Trabajo Final de Grado

Grado en Ingeniería Aeroespacial

Autor: Adrián Ruiz García

Tutor: Sergio García-Nieto Rodríguez

El siguiente documento incluye las siguientes partes en orden:

MEMORIA

PRESUPUESTO

PLIEGO DE CONDICIONES



UNIVERSITAT
POLITÀCNICA
DE VALÈNCIA


Escuela Técnica Superior de Ingeniería del Diseño



DISEÑO DE ALGORITMOS DE GUIADO PARA EL ATERRIJAZE AUTÓNOMO DE VEHÍCULOS AÉREOS NO TRIPULADOS BASADO EN VISIÓN ARTIFICIAL.

Memoria

Trabajo Final de Grado
Grado en Ingeniería Aeroespacial
Autor: Adrián Ruiz García
Tutor: Sergio García-Nieto Rodríguez

ÍNDICE

1. INTRODUCCIÓN	5
2. ALCANCE DEL PROYECTO Y OBJETIVOS.....	7
3. ANTECEDENTES	8
3.1. Introducción a los UAVs.....	8
3.2. Descripción de la plataforma Kadett 2400	14
3.3. Despegue y aterrizaje autónomos en UAVs	21
3.4. Plataforma de simulación.	23
4. SOLUCIÓN ADOPTADA	28
4.1. Visión artificial.....	28
4.2. Tratamiento de imagen.	31
4.3. Diseño de algoritmos mediante Matlab para aterrizaje autónomo.	43
5. TEST Y RESULTADOS.	47
6. CONCLUSIONES	55
7. BIBLIOGRAFÍA	56

ÍNDICE DE FIGURAS

Figura 1. Plataforma de vuelo Kadett 2400.....	5
Figura 2. De Havilland DH82B 'Queen Bee' Foto: Aerospace Museum San Diego	9
Figura 3. Los primeros tres OQ-1. Foto: Righter Family Archives	10
Figura 4. DC-130 de la U.S. Navy preparado para el lanzamiento de drones FireBee ...	11
Figura 5. MQ-9 Reaper.	11
Figura 6. Clasificación por tipo de aeronave. [4].....	12
Figura 7. Características según el tipo de aeronave. [4]	13
Figura 8. Aeronave Kadett 2400.	14
Figura 9. Esquema de conexión del receptor y los actuadores.....	15
Figura 10. Plataforma de vuelo.	16
Figura 11. Servo Switch Controller.	17
Figura 12. Sensor IG-500N. SBG Systems.	18
Figura 13. Relación plataforma de simulación.	23
Figura 14. Esquema de control básico.....	25
Figura 15. Esquema de control.....	25
Figura 16. Representación en Simulink de la aerodinámica del Kadett 2400.....	26
Figura 17. Transmisión Simulink-FlightGear.....	26
Figura 18. Imagen del Kadett 2400 en FlightGear.....	27
Figura 19. Cockpit View del Kadett 2400 en FlightGear.....	27
Figura 20. Captura de imagen del simulador FlightGear.....	29
Figura 21. "Cockpit View" del Kadett 2400.	30
Figura 22. Proceso de tratamiento de imagen.....	31

Figura 23. Comparación imagen en blanco y negro con imagen en color.	33
Figura 24. Recorte hecho mediante Matlab de la figura 19.	33
Figura 25. Transformación a escala de grises de la figura 19.	34
Figura 26. Borde ideal.	34
Figura 27. Borde rampa.	35
Figura 28. Detección de bordes mediante el algoritmo de Canny.	36
Figura 29. Detección de bordes usando el algoritmo de Prewitt.	36
Figura 30. Detección de bordes usando el operador de Sobel.	36
Figura 31. Histograma de la figura 19 obtenida mediante Matlab.	37
Figura 32. Binarizado de la figura 19.	38
Figura 33. Recta en el espacio x-y con los parámetros de Hough.	39
Figura 34. Ejemplo simple de uso de la transformada de Hough.	40
Figura 35. Líneas detectadas tras aplicar la transformada de Hough.	41
Figura 36. Imagen inicial.	41
Figura 37. Imagen recortada.	41
Figura 38. Imagen transformada a escala de grises.	42
Figura 39. Imagen binarizada.	42
Figura 40. Resultado de la aplicación de la transformada de Hough.	42
Figura 41. Puntos y líneas en imagen binarizada.	43
Figura 42. Imagen procesada de la pista a 200 pies de altitud.	44
Figura 43. Imagen procesada de la pista a 500 pies de altitud.	45
Figura 44. Imagen procesada de la pista a 1000 pies de altitud.	45
Figura 45. Ajuste potencial de la altitud de vuelo.	46
Figura 46. Ajuste logarítmico de la altitud de vuelo.	46
Figura 47. Diagrama de un PID.	47
Figura 48. Diagrama de un controlador basado en lógica difusa.	48
Figura 49. Ruta seguida en la simulación.	49
Figura 50. Esquema de Simulink con los bloques mencionados añadidos.	50
Figura 51. Variación temporal del Roll.	51
Figura 52. Variación temporal de las Distancias.	51
Figura 53. Validación de la función de transferencia usando 'ident'	52
Figura 54. Lugar de las raíces del sistema.	53
Figura 55. Respuesta del sistema a una entrada tipo escalón.	53
Figura 56. Variación del ángulo a controlar.	54

1. INTRODUCCIÓN

En el siguiente proyecto se ha diseñado e implementado un sistema de aterrizaje autónomo para vehículos no tripulados, basándose en un sistema de visión artificial. Además, se ha programado la adquisición de la imagen, así como su posterior tratamiento, de manera que, en función de las líneas de una pista de aterrizaje, la aeronave pueda saber su posición relativa a la pista, así como la altitud de vuelo.

Para realizar esta tarea, se parte de un modelo de UAV desarrollado por el Grupo de Control Predictivo y Optimización Heurística de la Universidad Politécnica de Valencia.



Figura 1. Plataforma de vuelo Kadett 2400.

El trabajo se realiza virtualmente, partiendo del modelo de la aeronave, realizado mediante Matlab y Simulink, el cual es transferido a la plataforma de simulación de vuelo FlightGear. De esta plataforma se obtiene la información relativa a la geografía y las imágenes que van a ser posteriormente tratadas.

En las imágenes obtenidas se deben visualizar las características que pueden proporcionar información relativa a la posición de la aeronave con respecto a la pista. Se establece que dichas características serán las líneas de la pista, a partir de las cuales se calculará el centro de la pista de aterrizaje para posteriormente referenciar su distancia al centro de la aeronave.

El tratamiento de la imagen se realiza mediante Matlab, usando su herramienta Image Procesing Toolbox. Con la imagen tratada las líneas de la pista se identifican usando el algoritmo de Hough.

Además de la distancia relativa al centro de la pista, se puede obtener en función de la separación de estas líneas la altitud de vuelo de la aeronave, de manera que el descenso se programe en función de dicha altitud.

Para que el centrado de la aeronave en la pista se realice de forma autónoma se ha configurado un controlador PID a partir de la función de transferencia que relaciona las distancias de la pista con el roll o ángulo de alabeo de la aeronave. El controlador será el encargado de reducir esta distancia y preparar la aeronave para su aterrizaje.

La función de transferencia se ha obtenido realizando una tabla de valores en la que se relaciona el roll, las distancias y el tiempo. Dichas variables se calculan realizando una simulación en FlightGear.

El UAV para el que se ha diseñado el proceso descrito es el Kadett 2400, plataforma de vuelo para la que fueron desarrolladas todo el comportamiento dinámico que ofrecen las ecuaciones de Bryan. Esta aeronave de ala fija presenta unas excelentes características para que se realice un control basado en técnicas de control predictivo gracias a la moderna instrumentación a bordo.

2. ALCANCE DEL PROYECTO Y OBJETIVOS

Los objetivos a alcanzar durante la realización del proyecto son los siguientes:

- Generar un sistema de adquisición de imagen *on-board* que transmita imágenes al control en tierra mediante radio.
- Tratar las imágenes de manera que se obtenga información consistente que permita el posterior aterrizaje de la aeronave.
- Diseñar un algoritmo que permita el aterrizaje autónomo de vehículos aéreos no tripulados (UAV), basándose en dichas imágenes.
- Implementar el código en el *Kadett-2400*.

Para poder alcanzar dichos objetivos se deben realizar tareas específicas para cada una de ellas.

Para comenzar, se debe entender la importancia que están cobrando los vehículos aéreos no tripulados.

En base al sistema de adquisición, hay que empezar por comprender la plataforma de simulación con la que se va a trabajar, así como el modelo implementado en Simulink que ha sido anteriormente desarrollado.

La familiarización con el entorno de tratamiento de imagen y más concretamente el que proporciona la herramienta Matlab es un paso fundamental a la hora de poder realizar un procesado adecuado.

Las tareas a realizar en este apartado son:

- Lectura de la imagen.
- Recorte y transformación de la imagen.
- Binarizado y tratamiento de las características de la imagen.
- Tratamiento adecuado de la información obtenida tras el procesado.

Tras este proceso, las tareas relativas al diseño de algoritmos son las siguientes:

- Generación de un patrón que permita obtener el centro de la pista utilizando Matlab.
- Cálculo de una regla que proporcione la altitud de vuelo en función de las características obtenidas.
- Simulación para obtener datos que permitan el diseño de un controlador que alinee la aeronave con la pista.
- Cálculo de la altitud de vuelo para comenzar el descenso.

Finalmente se deberá conseguir relacionar estos algoritmos diseñados con el modelo presente en Simulink para poder implementar el aterrizaje autónomo dentro de la plataforma Kadett 2400.

En último lugar se analizarán los resultados obtenidos, y se redactarán las conclusiones sacadas durante la realización del proyecto.

3. ANTECEDENTES

Antes de comenzar con la descripción del trabajo en sí, se debe introducir el campo de aplicación del trabajo, es decir, los vehículos aéreos no tripulados.

Primero se ofrece la definición de las aeronaves no tripuladas (UAVs), así como una introducción histórica. Tras ello, se hará un breve resumen de las distintas clases de UAVs que se pueden encontrar en la actualidad.

A continuación, se realizará la descripción de la plataforma de vuelo utilizada en este proyecto, el Kadett 2400.

Por último, se hablará sobre los antecedentes y aplicaciones del despegue y aterrizaje autónomo de este tipo de vehículos.

3.1. Introducción a los UAVs

Un UAV, del inglés Unmanned Aerial Vehicle, se define como “una aeronave reutilizable que vuela sin tripulación, siendo capaz de mantener un vuelo nivelado y sostenido, y que es propulsado por un motor de explosión o de reacción”. [4]

Sin embargo, se debe tener en cuenta que la condición de no alojar una tripulación a bordo no implica que exista un piloto, controlador u operador, que realice esta función desde tierra.

De hecho, la principal diferencia que se ha presentado durante el desarrollo histórico de estos vehículos es si podían ser pilotados a distancia, mediante el envío de señales de control, o si el pilotaje se hace con ayuda de ordenadores a bordo de la aeronave.

El concepto de UAV se ha extendido hasta los UAS (Sistema Aeronáutico No Tripulado), precisando en que además de la aeronave, debe existir una estación en tierra que completa la instrumentación y capacidades embarcadas.

Para terminar con la definición, se puede presentar el caso de una aeronave autónoma (AA: Autonomous Aircraft) o Sistema Aéreo Autónomo (AAS: Autonomus Aerial System), siendo este tipo de vehículo o sistema, capaz de desarrollar la misión de forma autónoma sin la necesidad de intervención humana. [1]

Se repasará a continuación algunos de los momentos más importantes de la evolución de los UAVs, en vistas a conocer de forma más amplia el estado actual en el que se encuentra la industria de las aeronaves no tripuladas.

Uno de los primeros caso con cierto interés del uso de un vehículo aéreo controlado y sin tripulación se dio poco antes de la Segunda Guerra Mundial, en el año 1935, cuando

el ejército británico desarrolló los modelos 'Queen' y 'Queen Bee', basándose en las aeronaves Fairey III F y Havilland Tiger Moth, respectivamente.

Dichas aeronaves tenían uso claramente militar, y estaban pilotadas mediante radiocontrol, desde tierra, siendo capaces de despegar tanto desde un aeródromo como desde el agua. El uso pretendido de este tipo de aeronaves era la experimentación con nuevos conceptos de desarrollo que hicieran aumentar la capacidad militar nacional, así como el entrenamiento de pilotos y operarios de cañones antiaéreos. Estos usos siguen estando presentes en la actualidad.

En la siguiente figura se puede observar una imagen del UAV Queen Bee.



Figura 2. De Havilland DH82B 'Queen Bee' Foto: Aerospace Museum San Diego

Paralelamente al desarrollo llevado a cabo en Reino Unido, el ejército de los Estados Unidos diseñó otro tipo de aeronave no tripulada con el mismo objetivo mencionado anteriormente, entrenar tanto a pilotos de aeronaves de caza como a los operarios de los cañones antiaéreos, encargados de la defensa de las bases estadounidenses ante ataques enemigos con aviones de combate. El guiado se realizaba también mediante radiofrecuencia desde tierra. A partir de este desarrollo, comenzó la comercialización de la tecnología aeronáutica para el vuelo recreativo y el aeromodelismo, siendo el OQ-1 (también conocido como RP-4) la primera aeronave no tripulada fabricada a gran escala.



Figura 3. Los primeros tres OQ-1. Foto: Righter Family Archives

Tras la Segunda Guerra Mundial, el desarrollo de los UAVs siguió su curso, y las potencias militares comenzaron a fabricar aeronaves capaces de reconocer y tomar imágenes del territorio enemigo, además de las posiciones de sus tropas. La gran ventaja de la ausencia de tripulación es que las dimensiones de la aeronave pueden ser sustancialmente inferiores, haciendo más compleja su detección. Además de evitar las pérdidas humanas.

Sin embargo, en la actualidad aparecen una gran cantidad de aplicaciones civiles que se irán desarrollando en los próximos años, así como toda la legislación pertinente para controlar el uso de dichas aeronaves en espacios públicos sin que exista peligro para las personas que las rodean.

Este es un rasgo que se mantiene en los UAVs actuales, habiendo aumentado notablemente la maniobrabilidad y la calidad de la visión de las cámaras presentes en las aeronaves.

El desarrollo de la vigilancia de territorio enemigo fue impulsado por el Ejército de Estados Unidos durante la guerra de Vietnam, pese a que ya se había hecho uso de este sistema durante la guerra de Corea. Concretamente se diseñó el Ryan Firebee, un drone que podía ser lanzado desde el aire. Dicha aeronave contaba con motores adicionales utilizados durante el despegue, de los cuales se deshacía tras la maniobra. Cabe destacar que fueron los primeros drones con propulsores turbojet.



Figura 4. DC-130 de la U.S. Navy preparado para el lanzamiento de drones FireBee

En la actualidad, la gran evolución de los sistemas automáticos y los sistemas de cómputo ha permitido una mejora en las prestaciones de los vehículos no tripulados. Además, también permiten mejoras técnicas que no se pueden incorporar si tenemos personas dentro de la aeronave, como pueden ser los límites en las aceleraciones de la cabina. A lo anterior se debe sumar la posibilidad de diseñar con una mayor eficiencia aerodinámica, ya que no es necesario contar con una cabina que permita una buena visibilidad al piloto. Esta característica es apreciable, por ejemplo, en el MQ-9 Reaper.



Figura 5. MQ-9 Reaper.

Durante estos últimos años, el mundo de los vehículos no tripulados se ha extendido al uso civil, tanto para investigación científica como para la aviación recreativa. Esto ha sido posible gracias a la aparición de nuevos conceptos de UAVs, los cuatrirrotores, que

permiten un vuelo más sencillo y que son fácilmente controlables desde el ordenador o incluso algunos smartphones.

En resumen, se puede decir que las aplicaciones de los UAVs son extensas y muy variadas, ejemplos de ello son: inspección de infraestructuras, vigilancia, supervisión de tráfico, filmografía, climatología, agricultura, intervención en desastres naturales, transporte de paquetería, búsqueda y rescate, etc.

Tras conocer más a fondo la evolución de los vehículos aéreos no tripulados se puede pasar a clasificarlos según algunas de sus características más importantes. La clasificación principal, y más evidente, es atendiendo al tipo de aeronave en sí misma.

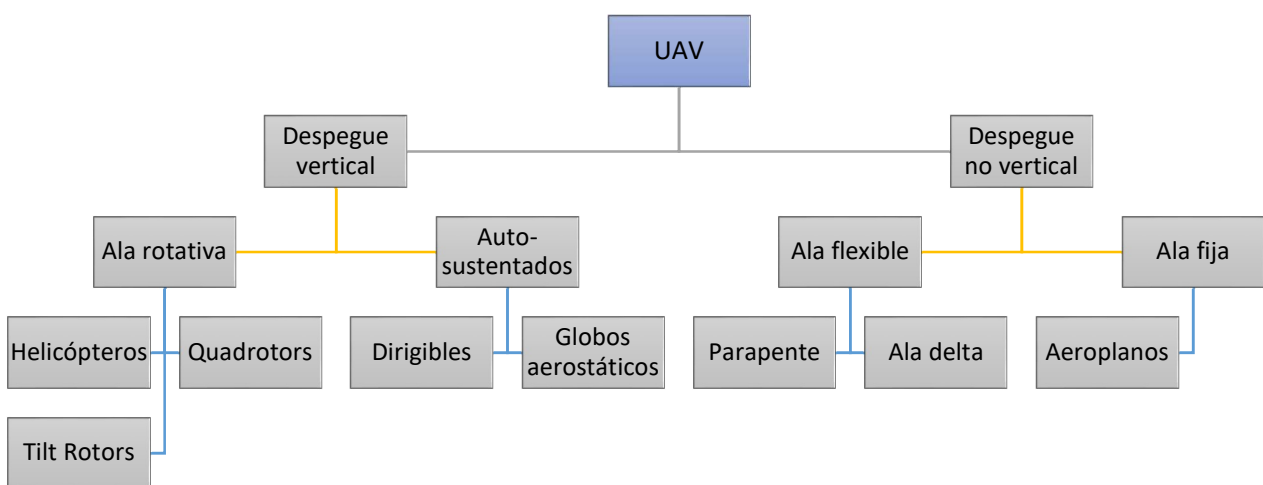


Figura 6. Clasificación por tipo de aeronave. [4]

Los vehículos de ala rotativa, los dirigibles y los aeroplanos son los que abarcan el mayor tipo de aeronaves no tripuladas, quedando el resto destinados a aplicaciones más específicas.

A continuación, se presenta una tabla que muestra las características más importantes de los UAVs anteriormente mencionados.

Característica	Helicópteros	Aeroplanos	Quadrotors	Dirigibles
Vuelo estacionario	***		***	****
Velocidad de desplazamiento	***	****	**	*
Autonomía	**	***	*	****
Resistencia a perturbaciones externas	**	****	**	*
Autoestabilidad	*	***	**	****
Vuelo vertical	****	*	****	**
Capacidad de carga	***	****	**	*
Vuelo en interiores	**	*	****	***
Techo de vuelo	**	****	*	***

Figura 7. Características según el tipo de aeronave. [4]

De esta forma, dependiendo de la misión, se seleccionará un tipo de UAV u otro, priorizando unas características sobre otras.

3.2. Descripción de la plataforma Kadett 2400

Se pasa ahora a la descripción de la plataforma de vuelo utilizada para la realización del proyecto.

Se trata de una plataforma de vuelo de ala fija completamente instrumentada, realizada por el Grupo de Control Predictivo y Optimización Heurística de la Universidad Politécnica de Valencia. Dicha plataforma consta de una aeronave radiocontrol de gran tamaño a la que se le han incorporado sensores inerciales, sensores de viento y el hardware de telemetría necesario para la comunicación en tiempo real con tierra.

El componente principal de la plataforma es la aeronave radiocontrol Kadett 2400, fabricado por la empresa Graupner. Observamos dicha aeronave en la figura siguiente.



Figura 8. Aeronave Kadett 2400.

El modelo Kadett 2400 presenta las siguientes características:

- Envergadura: 2.4 m.
- Longitud del fuselaje: 1.75 m.
- Peso en vacío: 4.9 kg.
- Superficie alar total: 1 m².
- Coeficiente peso/superficie: 49 g/m².
- Cuerda media: 0.4 m.

Su bajo índice peso/superficie, además de su reducido peso y dimensiones le confieren una ventaja respecto de otras soluciones en términos de estabilidad y capacidad de carga. Estos rasgos son imprescindibles a la hora de transformar un avión común en un vehículo autónomo, ya que será necesaria la incorporación de equipos con cierto

volumen y peso extra. Se calcula que, tras la inclusión de estos equipos, la masa e inercias del avión son:

$$m = 6.3 \text{ kg}; \quad I = \begin{bmatrix} 0.5789 & 0 & -0.0564 \\ 0 & 1.48 & 0 \\ -0.0564 & 0 & 1.9437 \end{bmatrix} \text{ kg} \cdot \text{m}^2; \quad [1]$$

La elección de una aeronave de ala fija se debe a que son las aeronaves que presentan una mejor relación de autonomía, resistencia a perturbaciones, capacidad de carga y estabilidad.

En el control manual de una aeronave de este tipo, intervienen dos factores principales. En primer lugar, tendremos un sistema de comunicación o control a distancia, que transmita a los actuadores las órdenes del piloto, que se encontrará en tierra. El segundo lo conforman el conjunto de actuadores que transforman esas órdenes en cambios en alerones, elevadores, revoluciones del motor, etc. El esquema sería el que se muestra a continuación.

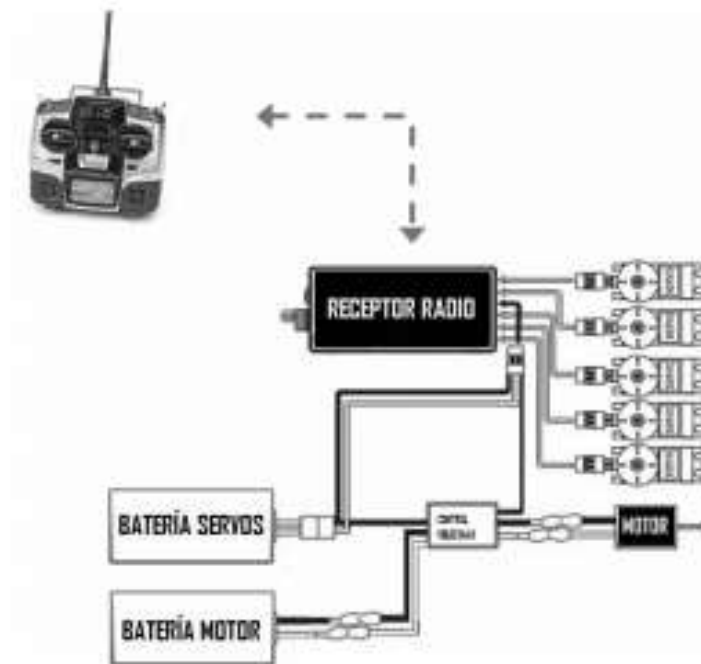


Figura 9. Esquema de conexión del receptor y los actuadores.

Por lo tanto, se ha incorporado al avión los dispositivos necesarios para su control en modo manual. Esto es, siete servomotores que controlan las cuatro superficies de control de las que dispone: timón de cola, elevadores, alerones y flaps.

La aeronave incorpora un motor OS OMA-5020-480, con dos baterías de tres celdas con una capacidad de 5000 mAh cada una.

El motor presenta así un empuje máximo de 32 N y las superficies de control admmiten deflexiones de hasta 0.4 rad. A pesar de estos límites, las restricciones serán menores con la intención de volar siempre con deflexiones pequeñas.

Las dos baterías a plena carga proporcionan una autonomía de vuelo de unos 30 minutos. La aeronave cuenta con un interruptor que permite el cambio de control manual a control automático implementado de forma independiente al sistema de control de vuelo, esto permite que ante cualquier fallo o problema en el programa se pueda recuperar el control manual de la aeronave.

Tanto los servomotores como el variador del motor son controlados a través de señales por modulación de ancho de pulso o señales PWM, gestionadas por el receptor.

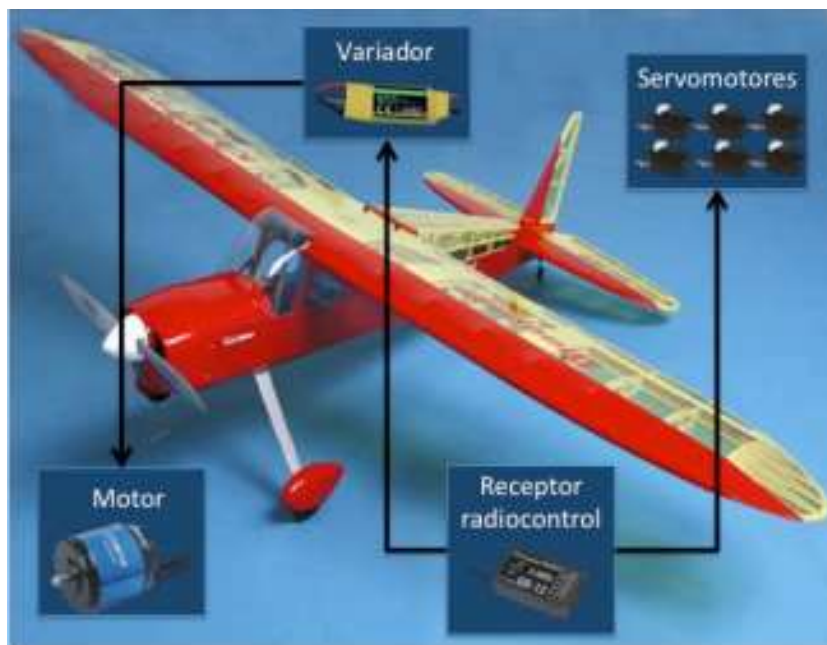


Figura 10. Plataforma de vuelo.

Se verá a continuación los sistemas que incorpora la aeronave:

- Servo Switch Controller: es el dispositivo puente entre la actuación manual y la autónoma. Este es capaz de realizar la conmutación entre las diferentes fuentes de entrada, además de encargarse de transformar paquetes binarios provenientes de un puerto serie a señales PWM y viceversa.

Por un lado, conecta con el receptor de radiocontrol. Hay 6 señales PWM disponibles, de las cuales se reserva una para comandar la orden de conmutación entre manual y automático. De esta forma, se consigue un mayor grado de seguridad, independizando la conmutación del ordenador de a bordo.

Con ello se consigue que, si el sistema se encuentra en modo manual, el resto de los canales del receptor son pasados por la SSC a los servomotores y al variador, dependiendo de lo que corresponda en cada momento. En caso contrario, el procesador de esta unidad transforma los paquetes de datos provenientes por puerto serie a las señales PWM que ordena el Flight control System (FCS).

Por último, esta unidad se puede configurar para que proporcione las medidas de las acciones de control introducidas por el piloto a través de su conexión con el ordenador, cuando se encuentre controlando la aeronave. Esta capacidad es muy importante puesto que permite identificar el modelo a partir de datos de vuelo real.

Las medidas obtenidas por la SSC presentan una frecuencia de muestreo máxima de 50 Hz.



Figura 11. Servo Switch Controller.

- Unidad IG-500N: uno de los puntos fuertes de la plataforma de vuelo es la sensorización que se ha montado a bordo, que no sólo permite una telemetría continua entre la estación de tierra y el UAV, sino que incorpora un sensor IG-500N que proporciona:
 - Ángulos de Euler.
 - Velocidad y posición en 3D.
 - Datos GPS.

En este caso, los datos GPS se suman a las lecturas de sensores inerciales de aceleraciones lineales y angulares con la ayuda de un filtro de Kalman, que actúa como un observador óptimo del estado en todo momento. La transmisión de telemetría se realiza mediante un módulo de radio con un alcance de hasta 20 km.

El paquete de medidas puede configurarse para ser transmitido a la frecuencia que se desee, hasta un máximo de 40 Hz.



Figura 12. Sensor IG-500N. SBG Systems.

A continuación, se especifican las características proporcionadas por el fabricante.

POSICIÓN		
Precisión de posición	SPS	2,5 CEP
Frecuencia de actualización máxima	Procesado onboard	100 Hz
	Procesado externo	500 Hz

Tabla 1. Características de posición del IG-500N de SBG Systems.

ACTITUD Y RUMBO		
Precisión estática (roll/pitch)		<0.5°
Precisión estática (yaw)		<1°
Precisión dinámica		1° RMS
Resolución angular		0.05°
Rangos dinámicos	Pitch	±90°
	Roll/Yaw	±180°

Tabla 2. Características de actitud y rumbo del IG-500N de SBG Systems

INTERFAZ	
Interfaz digital	SERIE, CAN y USB
Voltaje operativo	3.3-30 V
Consumo de potencia	800 mW

Tabla 3. Características de interfaz del IG-500N de SBG Systems.

SALIDAS
Ángulos de Euler
Quaternion
Matrices de rotación
Velocidad 3D
Posición 3D
Datos de sensores calibrados
Datos de sensores en crudo
Datos del GPS

Tabla 4. Posibles salidas del sensor IG-500N de SBG Systems.

- Flight Control System. Ordenador de a bordo: está constituido por un PC-1044 con QNX como Sistema Operativo de Tiempo Real. Esta unidad alberga los algoritmos de control, por lo que es la encargada de todas las tareas necesarias en cada fase de vuelo. La tarea principal consiste en recoger los datos de cada uno de los sistemas de medición para almacenarlos en la memoria. Cada nuevo conjunto de medidas se almacena cada 20 ms.

El FCS está constituido por los siguientes módulos físicos:

- Módulo CMX27686HK. Módulo CPU con procesador AMD Geode GX1 MMX basado en la arquitectura x86 de Intel.

Características de I/O:

- Controlador SVGA.
- Interfaz de teclado.
- 2 puertos serie RS32/422/485.
- Un reloj de tiempo real.
- 16 bits de I/O digital.
- Un puerto para ratón PS/2.
- 2 puertos USB.
- Par 10/100 con conexión Base T y TX basado en Intel 82559ER.

- HPWR104HR-75W. Suministrador de potencia que provee de una solución compacta para sistemas PC104 de alto rendimiento que requieren un suministro de potencia fiable.
 - Capacidad de alta potencia.
 - Potencia de salida máxima 75 W.
 - Eficiencia 88%.
 - Alto rango de voltajes de salida.
 - 8-30 Vdc de entrada. 36 V máx.
 - Protección de polaridad inversa de hasta 40V en la entrada.
 - Protección ante voltajes de entrada transitorios.
 - Salidas
 - +5 Vdc a 15 A.
 - +12 Vdc a 2 A.
 - -12 Vdc a 500 mA.
 - -5 Vdc a 100 mA en Bus PC104.

- Batería LIPO de 3 celdas.
 - 5 Ah.
 - 11.1 Vdc.

En términos de software se encuentra el sistema operativo QNX, mediante el cual se ha creado una aplicación de tiempo real para la adquisición de datos y para el control de estabilidad y navegación de la aeronave. A continuación, se describen algunas de las características más importantes del sistema operativo.

- QNX:
 - Sistema operativo basado en micro-Kernel. Permite prescindir de las funcionalidades que no van a ser necesarias.
 - Sistema operativo distribuido. Estas características son conocidas como Procesamiento Distribuido Transparente (TDP). Esto facilita el desarrollo de software, así como la depuración de errores.
 - Incluye en la capa externa al microkernel el resto de los servicios propios de un sistema Operativo.
 - Ofrece licencias académicas.
 - Disponible para arquitecturas x86.

3.3. Despegue y aterrizaje autónomos en UAVs

Como se ha mencionado anteriormente, este proyecto consiste en el diseño de algoritmos para el aterrizaje autónomo del Kadett 2400. Por lo que se terminará esta sección introductoria haciendo referencia a este tipo de maniobra y la evolución que ha sufrido durante el desarrollo de los UAVs.

En el mundo de los vehículos aéreos no tripulados, uno de los aspectos más críticos se produce en el momento del aterrizaje.

Se debe tener en cuenta que existe una gran diferencia entre los vehículos aéreos con despegue y aterrizaje vertical y los que realizan estas maniobras de manera horizontal.

Durante el desarrollo de estos vehículos, han sido diseñados e implementados numerosas técnicas de aterrizaje, las cuales no han resultado siempre exitosas. De hecho, una tercera parte de los accidentes de vehículos no tripulados ocurre durante las fases de despegue y aterrizaje, por lo que es de vital importancia dar con un sistema automático que reduzca considerablemente los riesgos que asumen estos vehículos durante dichas maniobras.

Si hablamos de vehículos aéreos VTOL (Vertical Take-Off and Landing), el aterrizaje se puede realizar sobre una plataforma, que puede estar en movimiento o no.

En el caso de los vehículos aéreos no VTOL, el aterrizaje se debe hacer sobre una pista, lo que complica considerablemente la maniobra.

En general, un UAV requiere un sistema que localice una zona de aterrizaje y se desplace hacia ella, una vez situado deberá localizar el punto concreto de aterrizaje y, por último, realizar un descenso controlado hasta finalizar la maniobra.

A la hora de la detección de la zona aterrizaje existen tres tipos de estrategias:

- Detección basada en GPS: es el sistema que conlleva el menor de los costes ya que simplemente utiliza el posicionamiento global del UAV y del lugar de aterrizaje para generar los comandos de movimiento hasta llegar al punto deseado. Sin embargo, presenta los inconvenientes de que es un sistema de seguridad reducida y la precisión no es siempre la deseada.
- Detección basada en SLAM: el sistema se basa en la teoría de la Localización y Modelado Simultáneos (SLAM). Mediante este sistema se obtienen estimaciones de la posición del UAV, así como un mapa conformado por puntos 3D que representan el ambiente que se observa a través de la cámara a bordo del vehículo. Este sistema es totalmente independiente de la señal GPS.
- Detección basada en microondas: este sistema consiste en un conjunto de antenas que se incorporan en las pistas de aterrizaje, más otro conjunto que será incluido en el UAV. Mediante el uso de ondas se posiciona el UAV en relación a las antenas situadas en la pista.

Tras este proceso, se debe orientar la aeronave para conseguir la alineación de esta con la zona de aterrizaje. Existen tres soluciones que destacan entre el resto:

- Sistemas de visión a bordo: requiere situar una cámara en la parte inferior del UAV. Esta se encarga de obtener imágenes que permiten obtener la posición y orientación de la pista mediante visión artificial.
- Sistemas de visión en pista: este sistema consiste en obtener la posición mediante un sistema que está instalado en tierra, y envía a la aeronave la información relativa a su posición.
- Radiobalizas: es el sistema más básico. Este consiste en cuatro radiobalizas situadas en cada una de las esquinas de la pista, las cuales emiten una señal captada por el vehículo aéreo.

Finalmente, el descenso se realiza cuando el UAV ya se encuentra orientado sobre la zona de aterrizaje, durante el cual se debe conocer el error de posición en cada instante. Utilizando esta información se dispondrá de un controlador que realizará las correcciones pertinentes, generando los comandos de movimiento oportunos.

3.4. Plataforma de simulación.

Para realizar las simulaciones de vuelo se trabaja con Matlab, Simulink y FlightGear.

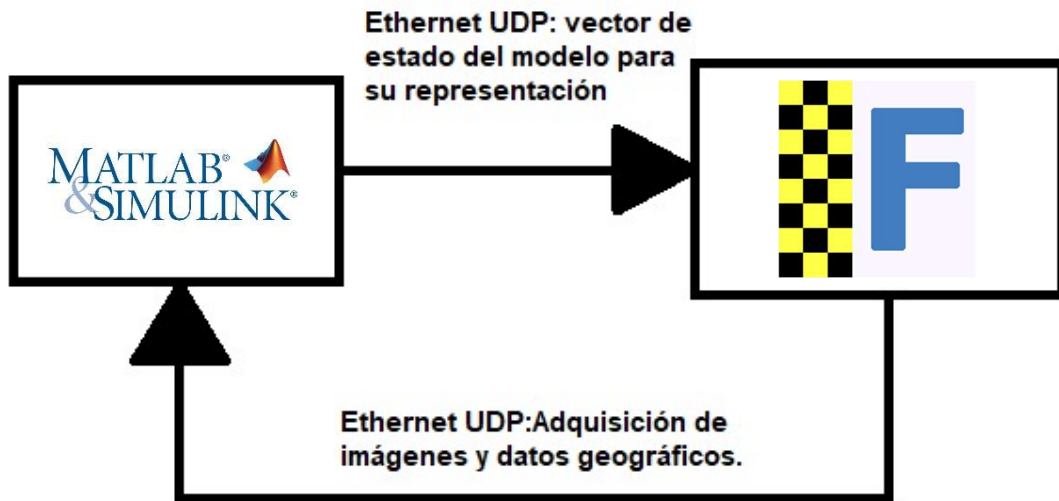


Figura 13. Relación plataforma de simulación.

De esta forma, podemos simular en FlightGear el UAV sobre el que trata el proyecto y utilizar FlightGear para incorporar en Matlab el tratamiento de imagen necesario para realizar la tarea sobre la que versa este proyecto.

El modelo ha sido desarrollado por el Grupo de Control Predictivo y Optimización Heurística de la Universidad Politécnica de Valencia. A continuación, se explicará cómo funciona y de qué partes consta.

Cuando se habla de control predictivo, se hace referencia a un método de control de sistemas que trata sencillamente los problemas multivariables, así como las restricciones pertinentes y las limitaciones de las propias variables del sistema. Esta forma de trabajo permite operar en unas condiciones muy cercanas a las óptimas, dotándolo de una funcionalidad idónea para aeronaves no tripuladas, ya que ofrecen unas condiciones de estabilidad muy buenas con un rendimiento alto.

La idea de este modelo se basa en un esquema de control predictivo en el que se han añadido restricciones en las salidas, las entradas, así como la variación de las entradas aplicadas. Esta tarea se resuelve utilizando una optimización QP en cada iteración del controlador. Este tipo de optimización minimiza una función cuadrática de variables sujetas a unas restricciones lineales.

Gracias a la optimización QP se impedirá que la aeronave funcione en zonas inestables del dominio de vuelo de la aeronave, por ejemplo, zonas cercanas a la entrada en pérdida.

Para entender el funcionamiento de la plataforma de simulación, lo mejor es partir del esquema básico del control predictivo no lineal, el cual se presenta a continuación.

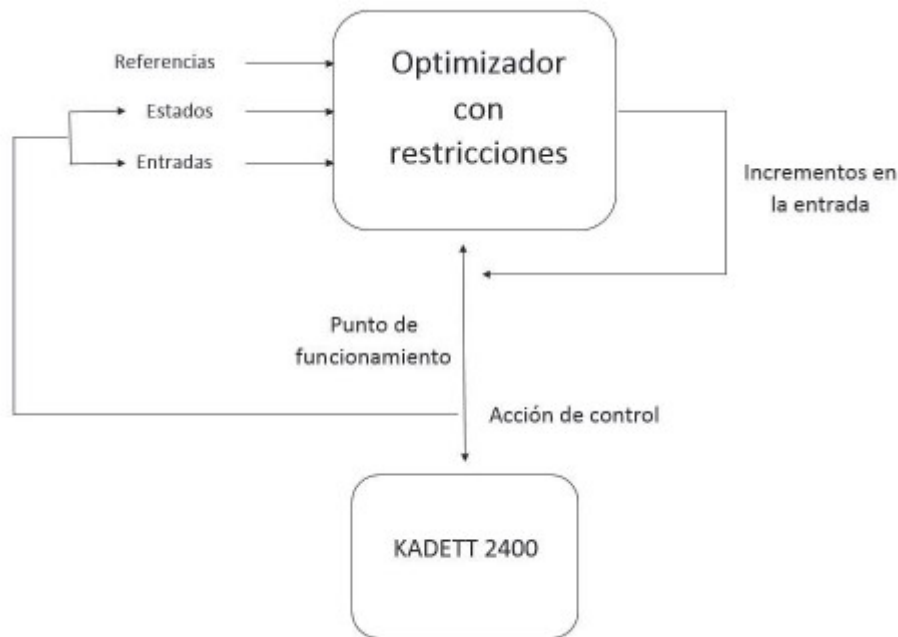


Tabla 5. Esquema básico de funcionamiento del control predictivo no lineal.

El elemento principal, el optimizador, cuenta con distintas entradas, las cuales son las referencias a lo largo del horizonte de predicción, además de los estados y entradas en tiempo real. Se debe tener en cuenta que el optimizador lleva incluidas las restricciones pertinentes en las salidas, las entradas y el incremento de las entradas.

A partir de los datos que entran en el optimizador se calculan las variaciones en las entradas de los actuadores de la plataforma de vuelo, de manera que sumadas a las entradas anteriores proporcionen el efecto integral que se traducirá en errores de posición nulos.

Encontramos tres matrices de restricciones:

- Matriz de restricciones encargada de limitar la variación máxima de la entrada a la plataforma de vuelo.
- Matriz de restricciones correspondiente a la entrada máxima del sistema.
- Matriz de restricciones correspondiente a las salidas máximas permitidas en el sistema.

Se pasa ahora a ver la implementación de esta estrategia de control en Simulink, que es la plataforma sobre la que se va a trabajar. Lo primero es presentar el esquema de control básico, el cual se encuentra a continuación.

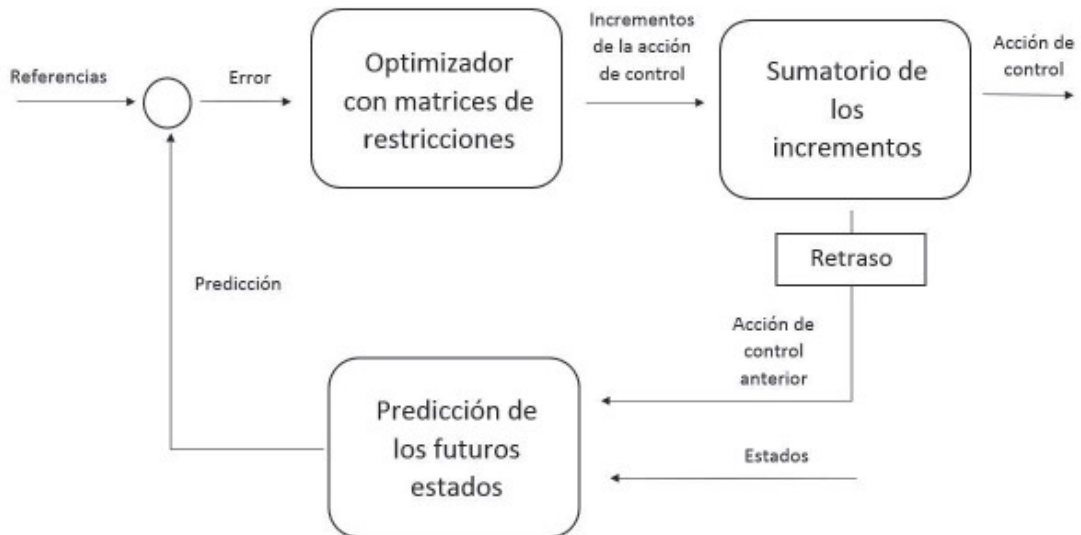


Figura 14. Esquema de control básico.

La implementación en Simulink se realizará tras definir este esquema y el tratamiento de las restricciones. En Simulink, el optimizador queda representado como un bloque capaz de ejecutar funciones en línea, mediante el cual se calculan las matrices definidas anteriormente, además de proporcionar los resultados de los incrementos de entrada que se deben aplicar.

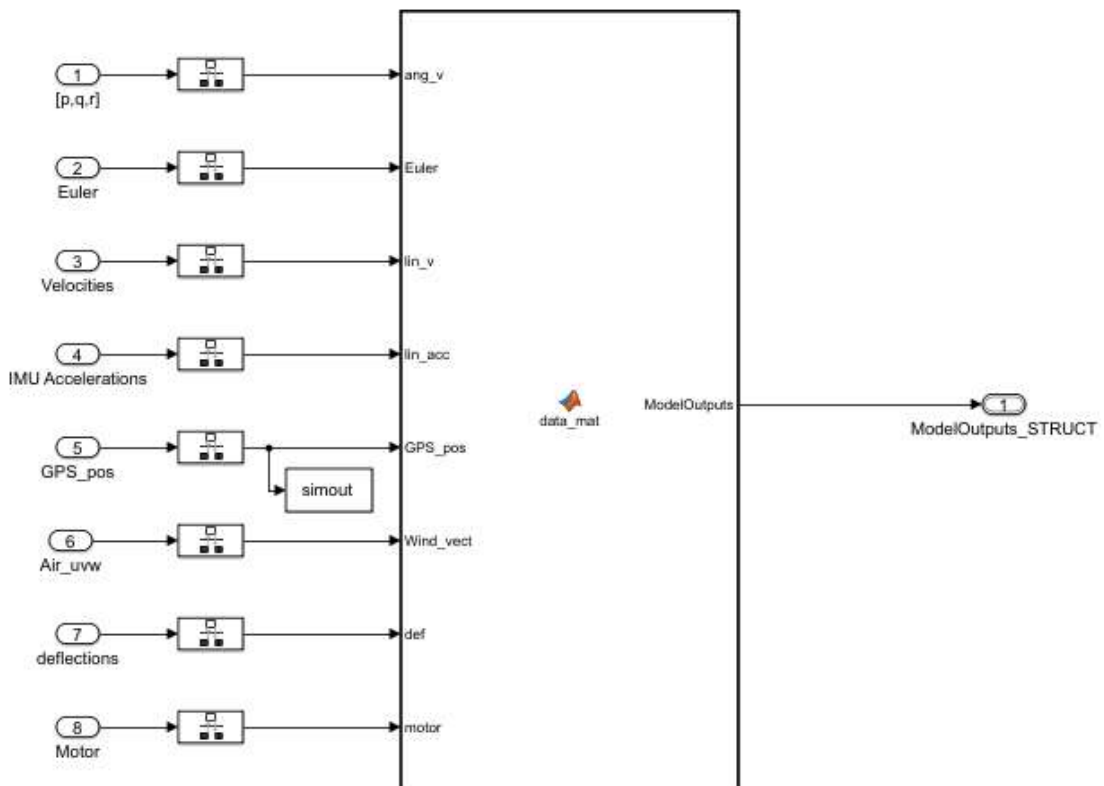


Figura 15. Esquema de control.

Mediante la implementación en Matlab y Simulink podemos incluir todos los aspectos aerodinámicos del UAV, así como representar la actitud de la aeronave en condiciones de vuelo reales.

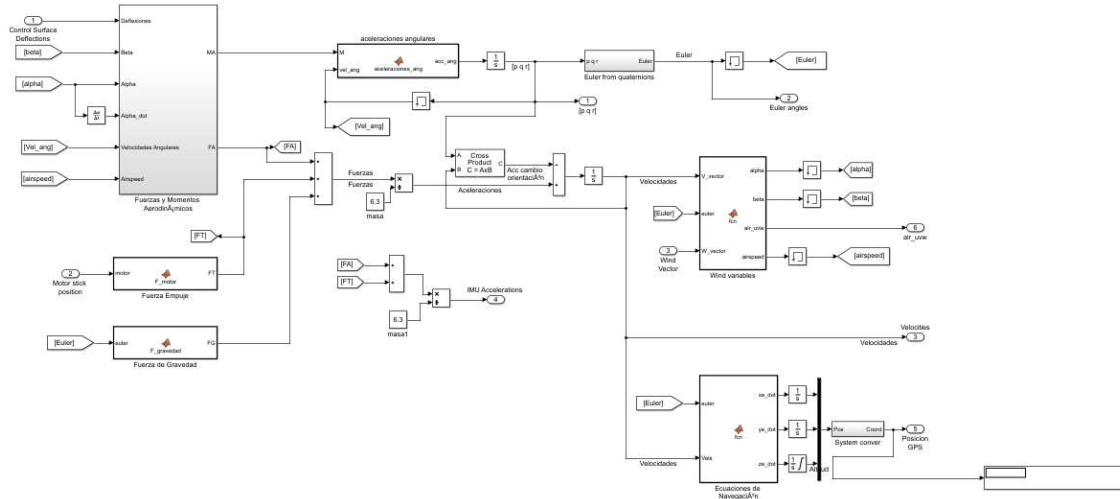


Figura 16. Representación en Simulink de la aerodinámica del Kadett 2400.

Esta representación de la aeronave se transmite mediante ethernet UDP al simulador de vuelo de software libre, FlightGear. De este modo, se puede observar en tiempo real el comportamiento de la aeronave. La conexión se realiza desde Simulink mediante el esquema que se muestra a continuación.

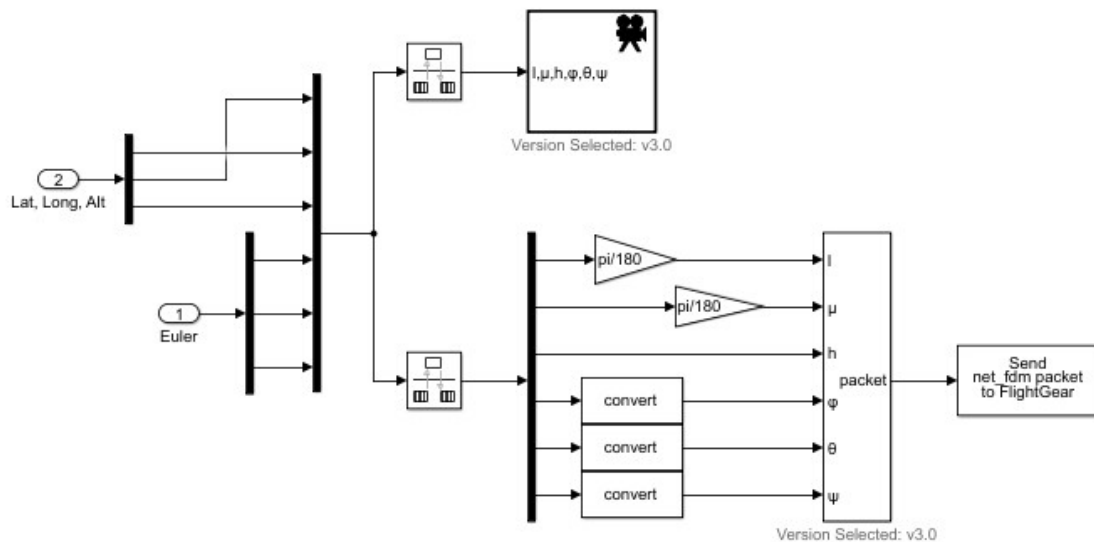


Figura 17. Transmisión Simulink-FlightGear.

De este modo, se pueden realizar simulaciones de vuelo, a través de las cuales es posible obtener información a la que sólo se podría acceder mediante pruebas de vuelo reales, lo que conllevaría un riesgo muy alto tanto a nivel económico como a nivel de equipos e instrumental.



Figura 18. Imagen del Kadett 2400 en FlightGear.

En el caso de este proyecto, la información de la que queremos disponer es la de la captura de pantalla del propio FlightGear, ya que es posible programarlo para situar la visión desde la parte inferior de la aeronave, en la que iría instalada la cámara mediante la cual se llevaría a cabo el aterrizaje de forma autónoma.

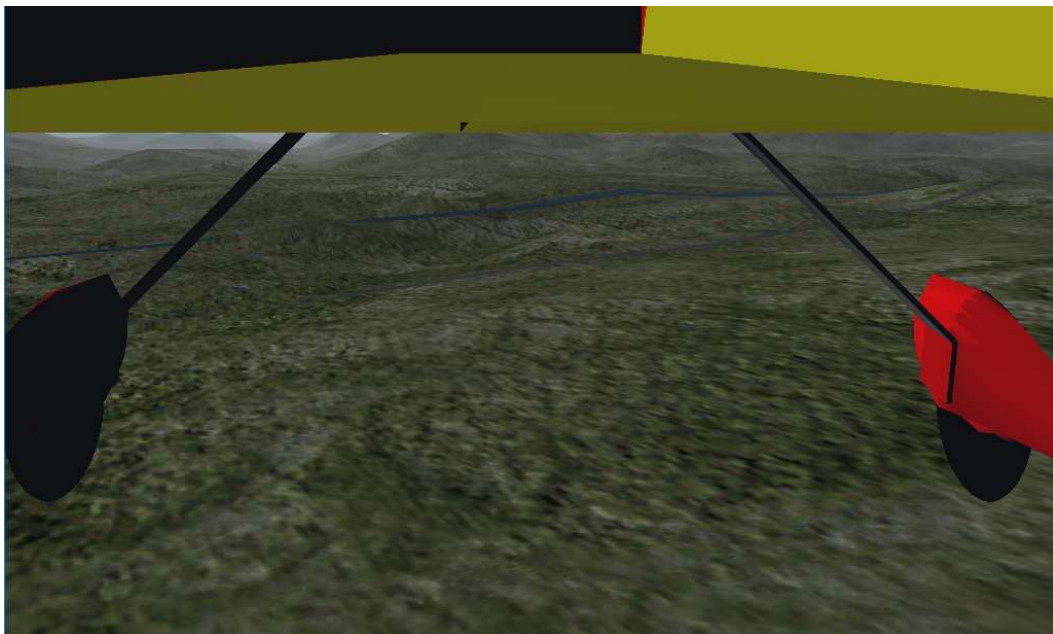


Figura 19. Cockpit View del Kadett 2400 en FlightGear.

4. SOLUCIÓN ADOPTADA

Tras analizar las opciones presentadas en el punto anterior y en base a la plataforma de vuelo con la que se iba a trabajar, se decidió que la mejor opción sería desarrollar un sistema para el aterrizaje autónomo de la aeronave basado en las imágenes obtenidas mediante una cámara situada en la parte inferior de dicho UAV, es decir, utilizando la visión artificial mencionada anteriormente.

Existen numerosos sistemas de aterrizaje que utilizan sistemas de navegación inercial (INS), posición GPS, balizas ILS y demás sistemas. Pero estas opciones no son siempre viables ya que requieren mayor complejidad dentro de la aeronave, además de un mayor coste computacional.

Sin embargo, la visión artificial es sencilla debido a que solo es necesaria una cámara de cuyas imágenes podremos obtener, con un coste computacional bajo, información sustancial de la altitud y orientación de nuestra aeronave con respecto a la pista.

Por estos motivos se ha seleccionado este método para resolver el problema planteado, dicha solución se abordará con más detenimiento en los puntos siguientes.

4.1. Visión artificial.

La visión artificial es un campo de la llamada “Inteligencia Artificial” que permite la obtención, procesado y análisis de cualquier tipo de información obtenida a través de una imagen.

Dicho de otro modo, el propósito de la visión artificial es programar un ordenador para que interprete una determinada imagen.

Mediante la visión artificial se pueden realizar tareas como:

- Detección, segmentación, localización y reconocimiento de ciertos objetos en una imagen.
- Seguimiento de un objeto en una secuencia de imágenes o video.
- Realizar inspecciones de objetos sin contacto físico.
- Discriminar un mismo objeto en varias imágenes.

La gran versatilidad de la tecnología de visión artificial hace que tenga aplicaciones en campos muy distintos: industria alimentaria, automoción, electrónica, packaging...

Esta tecnología es reciente, ya que fue impulsada a partir de los años 80, donde aplicada en los sistemas basados en computadores, se ve beneficiada por los grandes avances informáticos y de comunicaciones.

Hablando en términos más técnicos, la visión artificial consiste en captar imágenes en línea, es decir en tiempo real, mediante cámaras CCD y su posterior tratamiento

El proceso seguido por un sistema de visión artificial es el siguiente:

- La cámara capta una imagen de un objeto real.
- Dicha imagen es convertida a un formato digital.
- Esta imagen es procesada mediante un ordenador.
- Se obtienen unos resultados tras el proceso.
- El sistema actúa convenientemente en base a estos resultados.

Por lo tanto, se debería incorporar a la plataforma de vuelo una cámara incorporada a la aeronave, que permita obtener estas imágenes y transmitir las al computador correspondiente, situado en tierra.

Sin embargo, en este proyecto todo el sistema se ha desarrollado mediante un simulador.

Este simulador es el FlightGear, en su versión 3.4.0. Este simulador permite incorporar toda la dinámica conocida sobre la aeronave, obteniendo unas condiciones de vuelo bastante fieles a la realidad, evitando de esta forma tener que salir a realizar vuelos reales para obtener información y datos para su posterior tratamiento.



Figura 20. Captura de imagen del simulador FlightGear.

Para simular dicho vuelo, se ha utilizado la plataforma Matlab, representando dicha dinámica mediante bloques Simulink, desde el cual se puede controlar el avión, cambiando los ángulos de cabeceo y alabeo, que son los que se variarían para realizar la maniobra de aterrizaje.

Por lo tanto, arrancando el simulador desde la plataforma Matlab, es posible incorporar nuestro avión en dicho simulador, con las condiciones de vuelo que se le indiquen. Pudiendo seleccionar también el lugar de vuelo y múltiples opciones que no son fruto de estudio de este proyecto.

Se debe tener en cuenta, que el simulador incorpora varias vistas desde las que controlar el vuelo de nuestra aeronave y que permiten un análisis más exhaustivo del vuelo del UAV.

Sin embargo, nuestro interés es tratar el simulador como si de una imagen tomada por una cámara se tratara. Esto se realiza implementando una opción de vista, referida como "Cockpit View", que se traduce como vista desde la cabina. Como la aeronave no cuenta con una cabina al uso, puesto que es una aeronave no tripulada, se ha modificado esta vista para situarla como si fuera la visión que proporcionaría una cámara fija situada en la parte inferior de la aeronave.

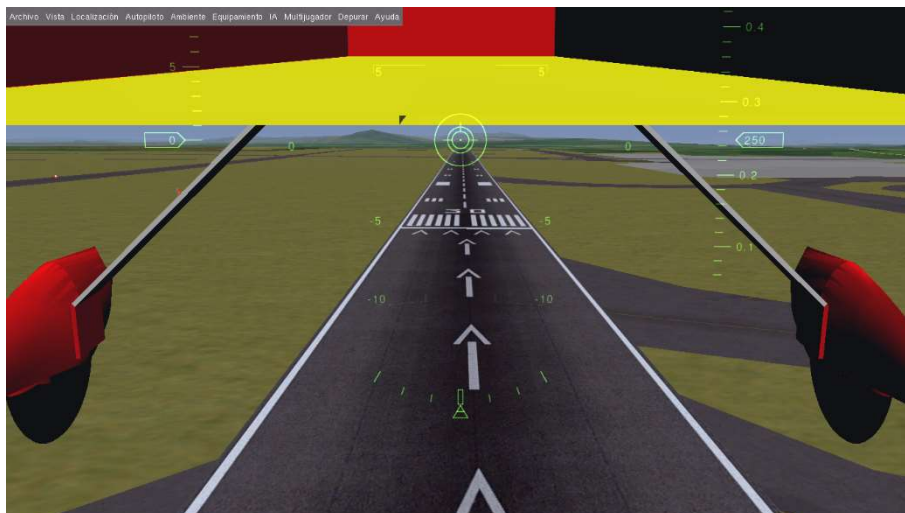


Figura 21. "Cockpit View" del Kadett 2400.

De esta forma es posible trabajar con una imagen que será muy parecida a la que se obtendría si se colocara una cámara en una ubicación similar de la aeronave.

Dicha imagen es con la que se trabajará y la que se tratará para obtener la información pertinente.

Para poder utilizar la imagen presentada en el simulador, se utiliza un script que inicializa todos los sistemas de este y que utiliza una de las salidas del sistema para almacenar capturas del simulador, almacenándolas en el computador.

Tras ello, y mediante el script principal del tratamiento de imagen, se procede a la lectura desde el directorio en el que se almacena la captura de pantalla en tiempo real. La imagen leída se actualiza cada 0.1 segundos, ya que no tendría sentido actualizar la

imagen con menor tiempo puesto que la respuesta no surtiría efecto en un tiempo menor.

Este tiempo se ha estimado mediante ensayo y error, teniendo en cuenta además las velocidades máximas de obtención de imagen de las cámaras que se podrían encontrar en el mercado.

Una vez leída la imagen se puede proceder a su tratamiento. Sin embargo, antes de tratar la imagen, se debe tener una idea inicial de qué se puede utilizar de dicha imagen para realizar la tarea que se desea, en este caso el aterrizaje autónomo del UAV.

Viendo la imagen de la figura 10, observamos que el elemento principal o más destacado de la imagen es la misma pista, por lo que el tratamiento se realizará entorno a las líneas de la pista.

4.2. Tratamiento de imagen.

En este apartado se trata de explicar el proceso que se sigue con las imágenes obtenidas mediante la simulación.

Dicho proceso consiste en recortar la instantánea de manera que sólo tengamos la porción de la imagen que nos interesa, tras lo cual se convierte a escala de grises, ya que de este modo se reduce sustancialmente el peso de los archivos, además esto nos permitirá hacer un binarizado de la imagen para destacar en ellas las líneas de la pista, las cuales serán el factor que determinará la posición de la aeronave relativa a la pista de aterrizaje, ya que a partir de ellas se podrá saber la altitud de vuelo y la distancia horizontal de la aeronave con respecto de la pista.

Dichas líneas se identificarán utilizando el algoritmo de Hough, herramienta presente en las funciones de Matlab, el cual se explica más adelante



Figura 22. Proceso de tratamiento de imagen.

Antes de realizar el proceso se debe explicar convenientemente la forma de tratar las imágenes en el entorno computacional y más concretamente en el entorno de Matlab.

El tratamiento de imágenes tiene por objeto modificar características de la imagen capturada para:

- Solucionar los problemas que suelen producirse al capturarlas, tales como falta de luz, colores apagados, tamaño excesivo...
- Restaurar imágenes con defectos, ojos rojos...
- Recortar partes de la imagen para manipularla o transformarla mediante filtros.

Para entender el proceso seguido, se comenzará comentando aspectos más técnicos referidos al tratamiento de imágenes. Las siguientes definiciones están dadas por el profesor Gonzalo Galiano Casas, del departamento de Matemáticas de la Universidad de Oviedo.

“Una imagen es una función bidimensional $f(x,y)$, donde x e y representan las coordenadas espaciales y el valor de f en cualquier par de coordenadas (x,y) representa la intensidad de la imagen en dicho punto”.

“Una imagen digital $f[x,y]$ descrita en un espacio 2D discreto se deriva de una imagen analógica $f(x,y)$ en un espacio 2D continuo a través de un proceso llamado digitalización”.

Matlab almacena las imágenes como vectores bidimensionales (matrices), en el que cada elemento de la matriz corresponde a un sólo píxel. Es decir, trabajar con imágenes en Matlab es equivalente a trabajar con matrices.

Al digitalizar una imagen, esta es dividida en M filas y N columnas, donde cada intersección de una fila y una columna es un píxel. Según el tipo de dato de cada píxel encontramos distintos tipos de datos que puede contener una imagen:

- double: doble precisión. 8 bytes por elemento, números que varían de -10308 a 10308.
- uint8: números enteros de 8 bits en el rango de [0, 255]. 1 byte por elemento.
- uint16: números enteros de 16 bits en el rango de [0, 65535]. 2 bytes por elemento.
- uint32: números enteros de 32 bits en el rango de [0, 4294967295]. 4 byte por elemento.
- int8: números enteros de 8 bits en el rango de [-128, 127]. 1 byte por elemento.
- int16: números enteros de 16 bits en el rango de [-32768, 32767]. 2 byte por elemento.
- Int32: números enteros de 32 bits en el rango de [-2147483648, 2147483647]. 4 byte por elemento.
- logical: los valores son 0 ó 1. 1 bit por elemento.

De esta forma, existen tres tipos principales de imágenes:

- Imagen de intensidad: es una matriz de datos cuyos valores han sido escalados para que representen intensidades de una escala de grises. En función del tipo

de dato el rango representado será mayor o menor. Si la imagen es del tipo double los valores de toman en el rango de 0 a 1 por convención.

- Imagen binaria: es una imagen en blanco y negro. Cada pixel tiene asignado un valor lógico de 0 ó 1 donde 0 representa el color negro y 1 el blanco.
- Imagen en color: cuenta con tres canales formados por tres imágenes de intensidad. Es decir, a cada píxel le corresponden tres valores (RGB) en lugar de uno solo.

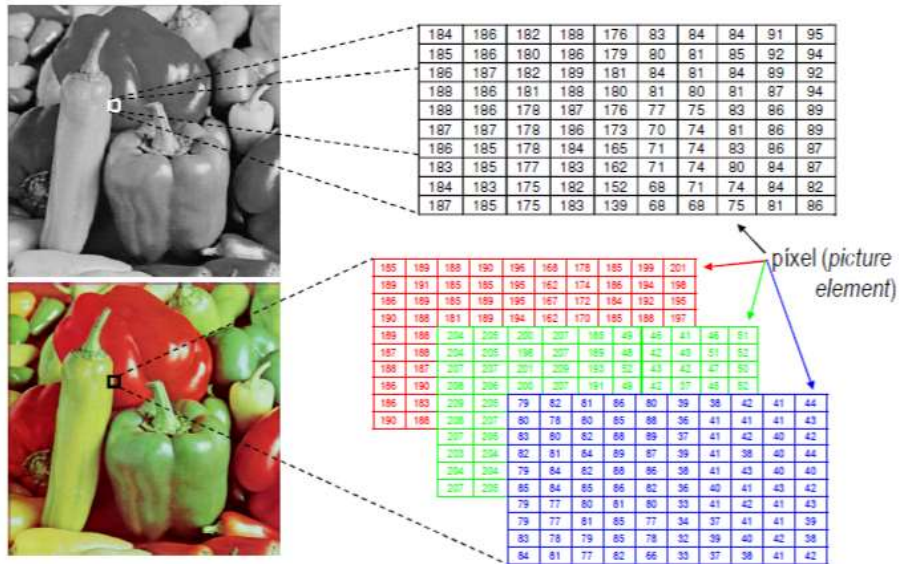


Figura 23. Comparación imagen en blanco y negro con imagen en color.

En el caso de este proyecto, la imagen obtenida mediante la cámara será en color, que como hemos visto significa tener una variable de 3 dimensiones. Por lo que la tarea inicial será convertir la imagen en color a una imagen en escala de grises.

Matlab incorpora el paquete Image Processing Toolbox, mediante el cual podemos realizar la tarea descrita mediante un sencillo comando: `rgb2gray(Imagen)`.

Antes de convertir la imagen a escala de grises, se debe recortar de manera que las zonas que no tienen interés no formen parte del procesado, esta tarea se realiza mediante el comando `"imcrop"`, indicando la región seleccionada para el recorte.



Figura 24. Recorte hecho mediante Matlab de la figura 19.

Se parte de la idea de utilizar un detector de bordes que nos permita obtener los puntos de la imagen que pertenecen a las rectas de la pista. Esto se hace mediante la segmentación de la imagen. La segmentación de imágenes divide la imagen en sus partes constituyentes hasta un nivel de subdivisión en el que se aíslan las regiones u objetos de interés. Los algoritmos de segmentación se basan dos propiedades básicas de los valores del nivel de gris:

- Discontinuidad: divide la imagen basándose en cambios bruscos de nivel de gris.
- Similitud: divide la imagen basándose en la búsqueda de zonas con valores similares, conforme a unos criterios seleccionados previamente.



Figura 25. Transformación a escala de grises de la figura 19.

En general, podemos encontrar dos tipos de bordes en una imagen:

- Borde ideal: forman un camino de un pixel de ancho, en los que se produce un cambio perpendicular en el nivel de gris.

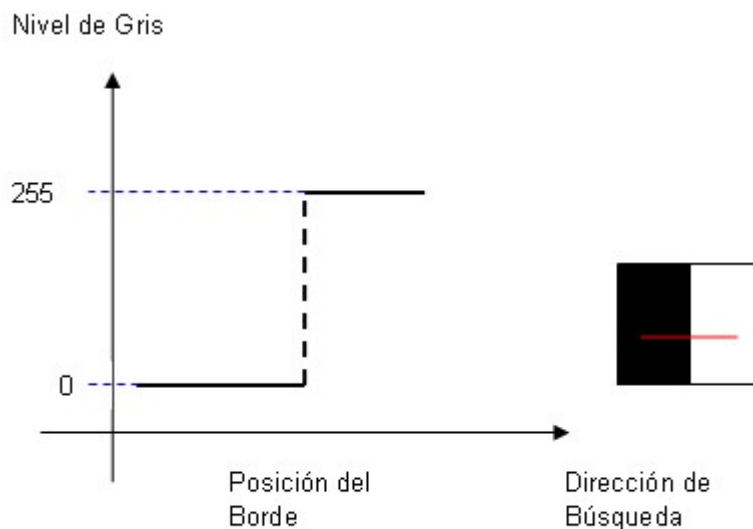


Figura 26. Borde ideal.

- Borde rampa: forman un conjunto de píxeles conexos en los que se produce, en una determinada dirección, una variación gradual en el nivel de gris.

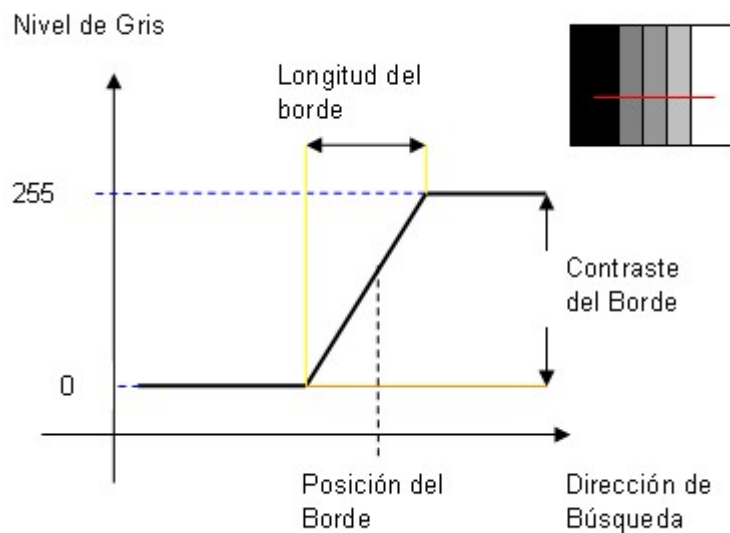


Figura 27. Borde rampa.

La idea en la que se basan la mayor parte de las técnicas de detección de bordes es el cálculo de un operador local de derivación, ya que un píxel pertenece a un borde si se produce un cambio brusco entre niveles de grises con sus vecinos.

Un problema a tener en cuenta es que, en la búsqueda de los cambios bruscos para detectar bordes, también se detectará el ruido. De manera que se debe realizar una suavizado de la imagen para reducir este problema y posteriormente detectar los posibles candidatos a ser puntos del borde.

Matlab incorpora varios métodos de detección de bordes de primer orden, de los cuales se han seleccionado los operadores de Canny, Prewitt y Sobel, los cuales se explicarán a continuación:

- Canny: es el detector de bordes más potente que existe actualmente. El proceso que sigue es el siguiente:
 1. Se suaviza la imagen mediante una convolución usando un filtro gaussiano, de esta forma se eliminan los ruidos.
 2. Se calcula el gradiente de la imagen suavizada, para detectar los píxeles donde se produce una variación importante.
 3. Se eliminan aquellos píxeles que no son máximos locales en la dirección del gradiente.
 4. Se realiza un proceso de umbralización para determinar los píxeles del borde.

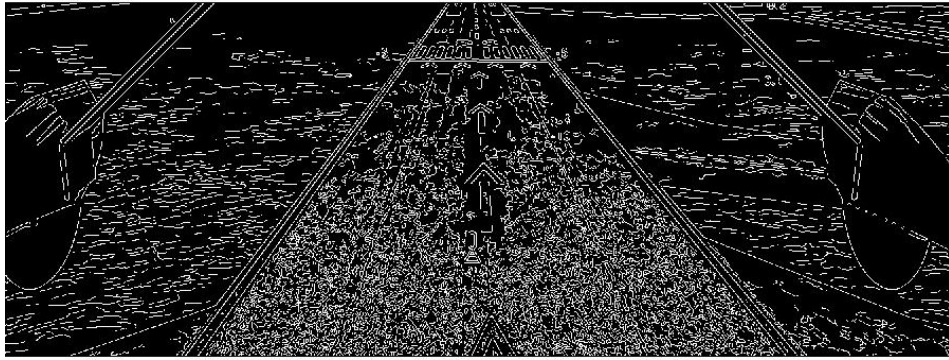


Figura 28. Detección de bordes mediante el algoritmo de Canny.

- Prewitt: detecta cambios en los límites de la respuesta al ruido, así como los cambios abruptos en la intensidad de la imagen. Consta de ocho matrices que se aplican pixel a pixel en la imagen. Cada matriz toma el nombre de un punto cardinal. Por último, se suman las imágenes para obtener bordes bien marcados.

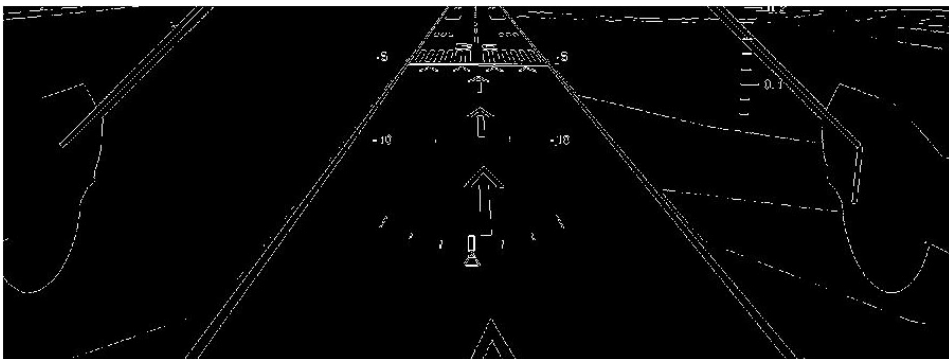


Figura 29. Detección de bordes usando el algoritmo de Prewitt.

- Sobel: calcula el gradiente de la intensidad de una imagen en cada punto. De este modo, para cada punto devuelve la magnitud del mayor cambio posible. Matemáticamente, el gradiente de una función de dos variables para cada punto es un vector bidimensional. Para cada punto de la imagen, el vector gradiente apunta en dirección del incremento máximo posible de la intensidad, y la magnitud del vector gradiente corresponde a la cantidad de cambio de la intensidad en esa dirección.

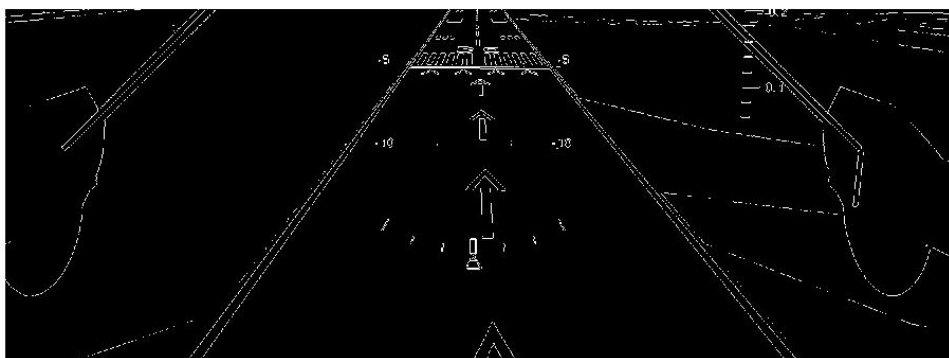


Figura 30. Detección de bordes usando el operador de Sobel.

Debido a las imperfecciones y a la complejidad, ya sea de la imagen captada o del detector de bordes, existen muchos puntos resaltados que no son de nuestro interés. Por lo tanto, se debe obtener una imagen en la que las diferencias entre los puntos a resaltar y los que no son objeto de interés sean mayores.

Observando la figura 19, vemos que las líneas son puramente blancas, siendo el único elemento de la imagen con este color. Se concluye que la mejor opción sería convertir a negro todo aquello que no es de color blanco en la imagen inicial.

Por lo tanto, una vez obtenida la imagen de la zona deseada y ha sido convertida a escala de grises, se realiza un binarizado de la misma para facilitar el procesamiento posterior, ya que de este modo únicamente se trabajará con una imagen en la que cada píxel es un 1 o un 0, tal y como se ha explicado anteriormente.

Para comenzar con esta tarea se realizó un histograma de la imagen. Un histograma es una gráfica que representa los niveles de intensidad del color de una imagen con respecto al número de píxeles presentes en ella con cada intensidad de color. El histograma nos ayudará a hacernos una idea de cómo es de correcta la exposición de una imagen. Una imagen que contenga todos los tonos de grises tendrá un histograma más o menos uniforme a lo largo de todo el eje horizontal.

El paquete Image Processing Toolbox incorpora también una función que representa el histograma de una imagen.

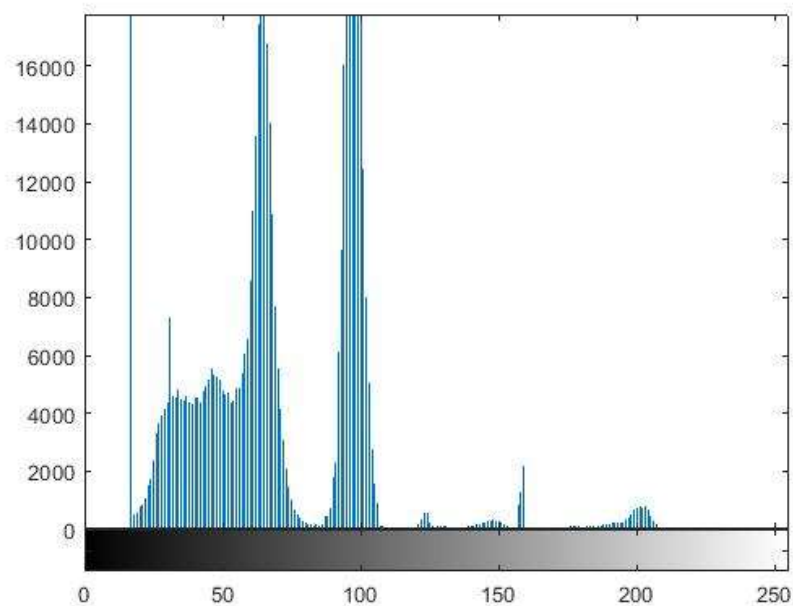


Figura 31. Histograma de la figura 19 obtenida mediante Matlab

Una vez obtenido el histograma es posible precisar más en el valor del umbral que se utilizará para asignar el color negro o el color blanco a cada píxel de la imagen.

En este caso se considera que, si el píxel presenta una intensidad superior a 170, se le asignará un valor de 1 (color blanco), quedando el resto de los píxeles menores que 170 con un valor de 0 (color negro).

Como ejemplo, se muestra a continuación el binarizado de la imagen de la Figura 19.



Figura 32. Binarizado de la figura 19.

Observando esta imagen, se puede comprobar que las líneas de la pista resaltan sobre cualquier otro elemento que se pudiera encontrar previamente en la imagen.

Por lo tanto, el paso siguiente consistirá en detectar tales líneas, con la intención de obtener información sobre ellas.

Existen varios métodos para la detección de segmentos y patrones en una imagen determinada. Sin embargo, en este proyecto esa detección se ha realizado mediante la transformada de Hough.

La transformada de Hough es una herramienta utilizada para detectar figuras en una imagen digital que pueden ser expresadas matemáticamente, tales como rectas, círculos o elipses.

El objetivo de la transformada de Hough es encontrar puntos alineados que puedan existir en la imagen, es decir, puntos en la imagen que satisfagan la ecuación de la recta, para distintos valores de ρ y θ . Para poder representar todas las posibles rectas que puedan aparecer en la imagen, se utiliza la ecuación de la recta en coordenadas polares:

$$x \cdot \cos \theta + y \cdot \sin \theta = \rho$$

Por tanto, hay que realizar una transformación entre el plano de la imagen, expresado en coordenadas x-y, y el plano de los parámetros (ρ , θ).

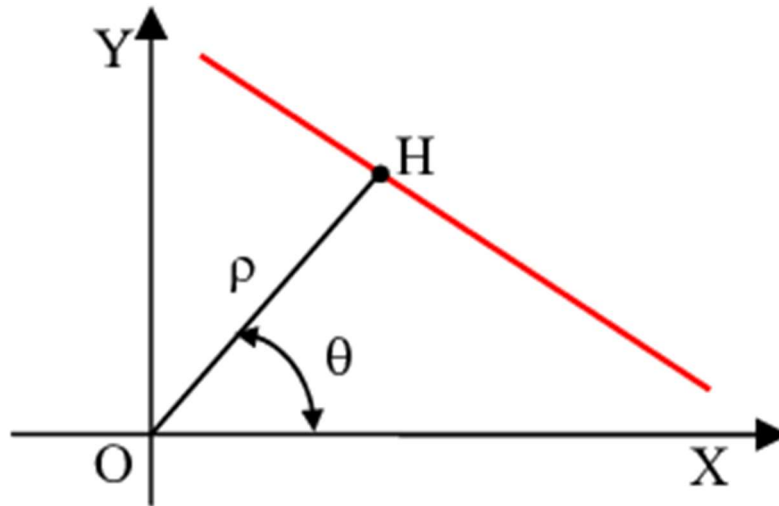


Figura 33. Recta en el espacio x-y con los parámetros de Hough.

Para un punto arbitrario en la imagen con coordenadas (x_0, y_0) , las rectas que pasan por ese punto son los pares (ρ, θ) , donde ρ está determinado por θ . Esto corresponde a una curva sinusoidal en el plano (ρ, θ) que es única para ese punto.

Si las curvas correspondientes a dos puntos se intersecan, el punto de intersección en el espacio de Hough corresponde a una línea en el espacio de la imagen que pasa por estos dos puntos. El algoritmo de la transformada de Hough usa una matriz, llamada acumulador, cuya dimensión es igual al número de parámetros desconocidos del problema, siendo este valor igual a dos en el caso específico de una recta.

Para construir el acumulador es necesario discretizar los parámetros que describen la figura. Cada celda del acumulador representa una figura cuyos parámetros se pueden obtener a partir de la posición de la celda. Cada punto en la imagen vota por las posibles rectas a las que puede pertenecer ese punto. Esto se logra buscando todas las posibles combinaciones de valores para parámetros que describen la figura. Si es correcto, se calculan entonces los parámetros de esa figura, en nuestro caso dos, y después se obtiene la posición en el acumulador correspondiente a la figura definida, incrementando el valor que hay en esa posición.

Un número de votos elevado indica que el punto pertenece a la recta.

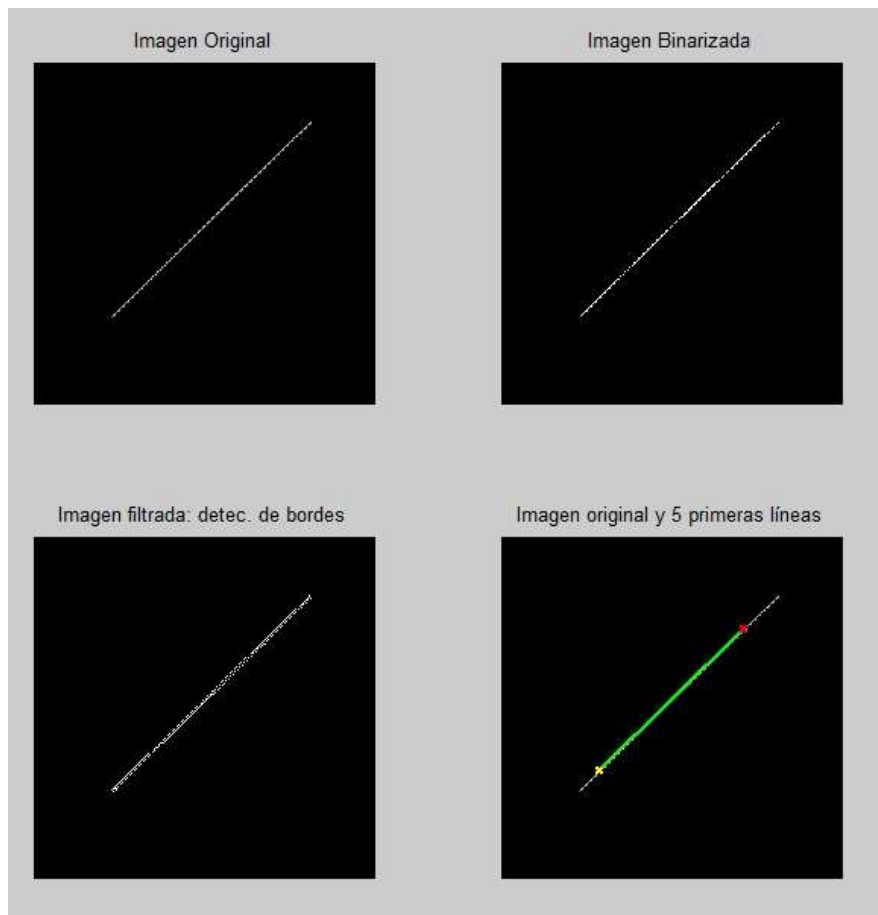


Figura 34. Ejemplo simple de uso de la transformada de Hough.

Existe en Matlab una función que aplica la transformada de Hough a una imagen con los siguientes parámetros de entrada y salida:

$$[H, T, R] = \text{hough}(I);$$

- I: imagen binaria.
- H: transformada de Hough
- T: θ , vector que contiene el ángulo correspondiente a cada columna de H.
- R: ρ , vector que contiene los valores de ρ correspondiente a cada fila de h

Aplicando esta transformada a la imagen binarizada se puede observar como las líneas de la pista son detectadas y resaltadas sobre la imagen.

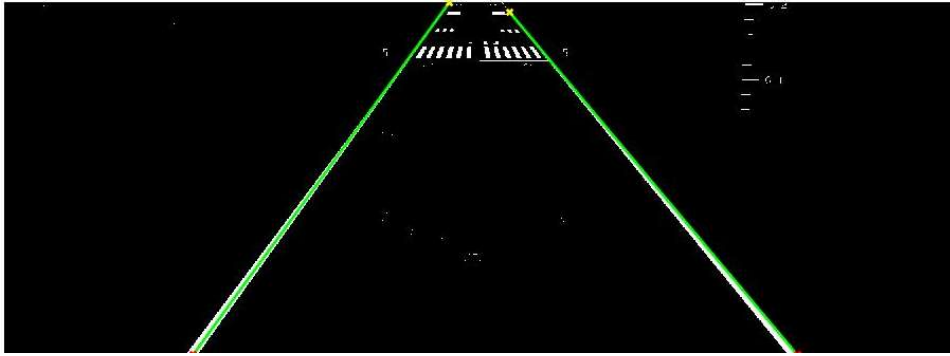


Figura 35. Líneas detectadas tras aplicar la transformada de Hough

Una vez obtenidas las líneas en la imagen a tratar y su ubicación dentro de la misma, se debe proceder a encontrar un patrón que nos permita obtener información sobre la orientación y la altitud de la aeronave con respecto a la pista de aterrizaje.

A modo de resumen, se especifica el proceso seguido con cada imagen que es adquirida mediante los sistemas instalados en el UAV.

1. Adquisición de la imagen.

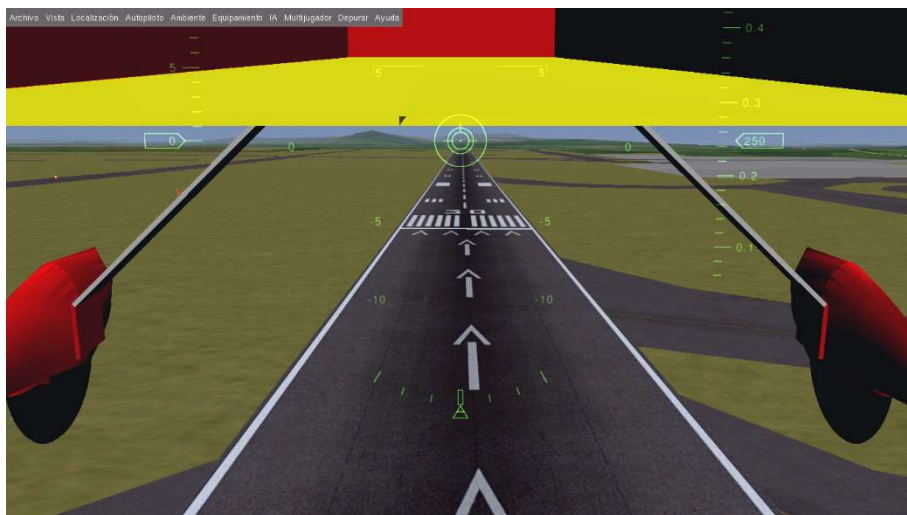


Figura 36. Imagen inicial.

2. Recorte de la imagen.



Figura 37. Imagen recortada.

3. Transformación a escala de grises.



Figura 38. Imagen transformada a escala de grises.

4. Binarizado de la imagen.



Figura 39. Imagen binarizada.

5. Aplicación de la transformada de Hough para la detección de las líneas de la pista.

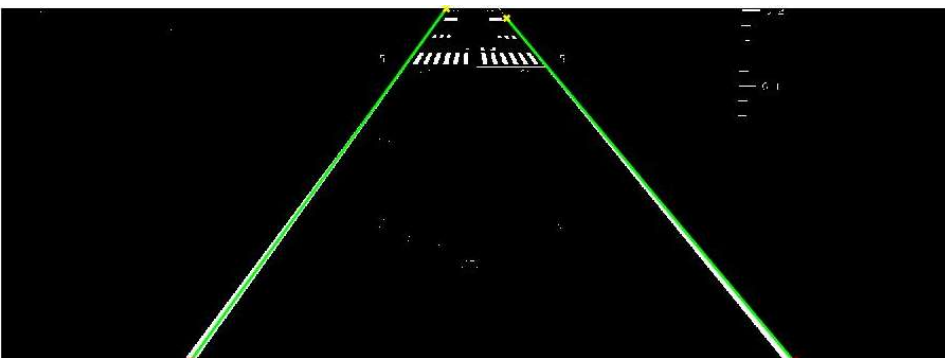


Figura 40. Resultado de la aplicación de la transformada de Hough.

4.3. Diseño de algoritmos mediante Matlab para aterrizaje autónomo.

El algoritmo diseñado ofrece un valor de la distancia y de la altitud de vuelo basándose en la posición de las líneas de la pista de aterrizaje.

La idea del algoritmo es que la aeronave se oriente con la pista, reduciendo la distancia entre el centro de la pista y el centro de la aeronave variando el roll, tras lo cual, en función de la separación de las líneas se identificará la altura de vuelo para comenzar el descenso.

Tras realizar el tratamiento de imagen, se obtienen dos líneas principales, las cuales se corresponden con los límites de la pista de aterrizaje. Cada línea está formada por dos puntos: el punto inicial y el punto final. Las posiciones exactas de estos puntos se conocen, ya que las hemos obtenido mediante el algoritmo de Hough.

En la siguiente figura podemos observar dichas líneas, la línea A y la B, con los correspondientes puntos 1 (en rojo), para el punto inicial, y 2 (en amarillo), para el punto final.

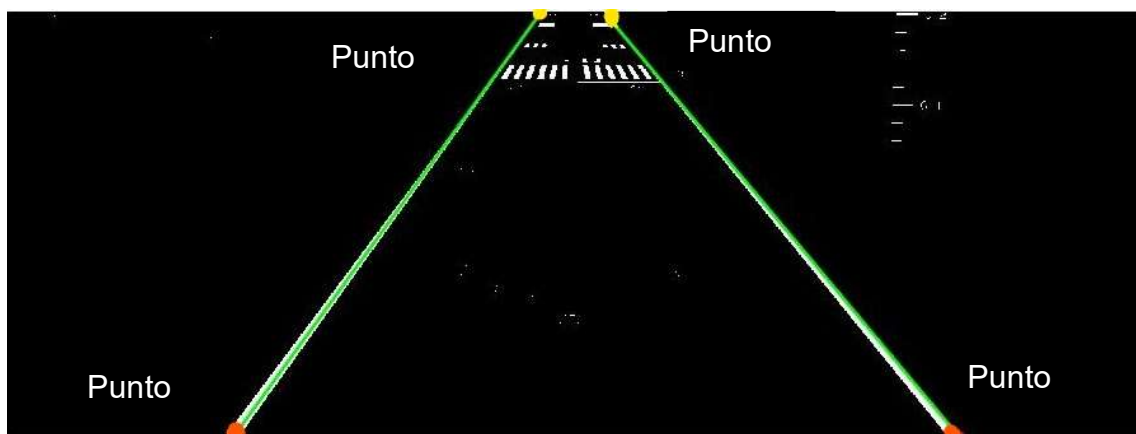


Figura 41. Puntos y líneas en imagen binarizada.

Con los puntos A1 y B1, es posible calcular la posición del centro de la pista con respecto a la imagen en la parte inferior, y con los puntos A2 y B2, la posición del centro de la pista con respecto a la imagen en la parte superior.

De este modo, se obtiene una línea imaginaria que se correspondería con el centro exacto de la pista de aterrizaje.

Como la imagen se corresponde con la de una cámara fija situada en la parte central inferior de la aeronave, se puede afirmar que el centro de la imagen equivale al centro de la aeronave, por lo que la misión del algoritmo consiste en alinear el centro de la pista que hemos obtenido con el centro de la imagen, variando el roll.

Es necesario trabajar con dos puntos del eje de la pista, ya que podría darse el caso de centrar la parte inferior de la imagen, pero que el rumbo no fuera el correcto. Eso se corrige calculando la posición de un segundo punto central de la pista, más alejado del primero que se calculó.

Además, se debe tener en cuenta si la aeronave se encuentra más a la derecha o a la izquierda de la pista. Simplemente basta con considerar el signo de las distancias relativas que hemos obtenido. En este caso, se ha convenido que si el valor de la distancia es positivo el UAV estará desplazado hacia la izquierda. Si por el contrario es negativo, el UAV estará desplazado a la derecha.

Estos valores de las distancias se utilizarán para estudiar cómo se comportan en función de la variación del roll para posteriormente obtener una relación que permita reducir estos valores durante el vuelo.

Las distancias obtenidas son distancias en píxeles, no son distancias reales, pero esto no es un problema ya que únicamente se necesita reducir su valor, no analizar o tratar con valores de distancias de longitud.

Para el caso de la altitud de vuelo es diferente, puesto que sí que es necesario un valor real, ya que en función de esta altitud la aeronave sabrá la orientación de cabeceo que deberá asignar para realizar el aterrizaje.

Por lo tanto, se debe establecer una relación entre las distancias de las líneas de la pista a partir de las cuales se consiga obtener un valor de la altitud de vuelo. El proceso seguido para esta tarea es el siguiente.

Primero se tomaron instantáneas a diferentes altitudes con la pista ya alineada, dichos valores fueron: 200, 250, 500 y 1000 pies. Estas imágenes fueron procesadas hasta obtener los puntos A1 y B1 que se han mencionado anteriormente. La Figura 34 se corresponde con la imagen procesada a 250 pies.



Figura 42. Imagen procesada de la pista a 200 pies de altitud.

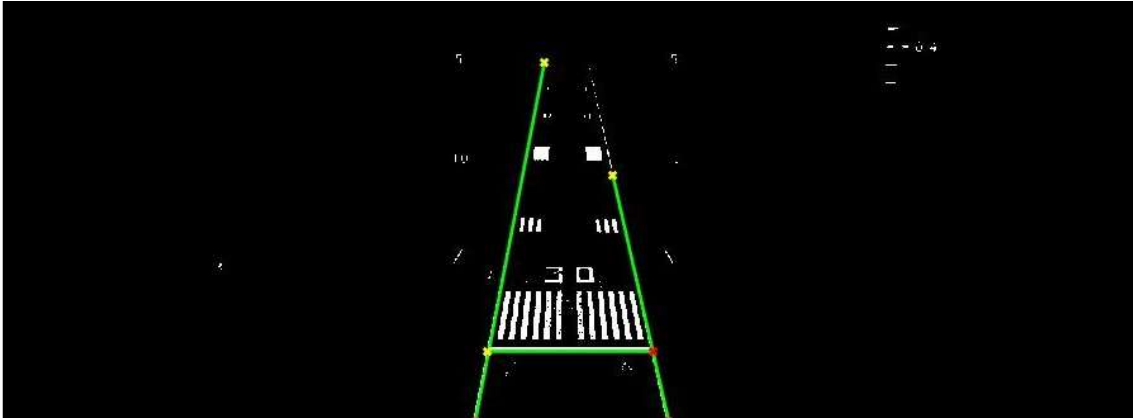


Figura 43. Imagen procesada de la pista a 500 pies de altitud.

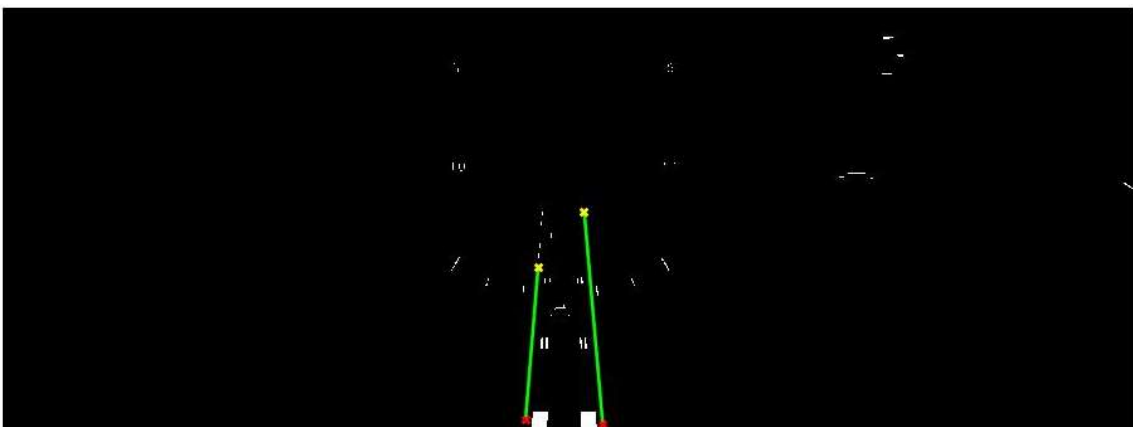


Figura 44. Imagen procesada de la pista a 1000 pies de altitud.

A partir de los puntos mencionados se calculó la distancia horizontal que los separaba y dicha distancia se relacionó con la altitud de vuelo correspondiente, obteniendo la siguiente tabla.

Separación (píxeles)	Altitud (ft)
856	250
229	500
92	1000
1304	218

Tabla 6. Relación de las altitudes de vuelo y las distancias.

Con estos datos, se puede establecer una regla que proporcione el valor de la altitud de vuelo a partir de la distancia de las líneas de pista obtenidas de la imagen captada por la aeronave. Para establecer esta regla se probó un ajuste potencial y un ajuste logarítmico, los cuales se muestran a continuación.

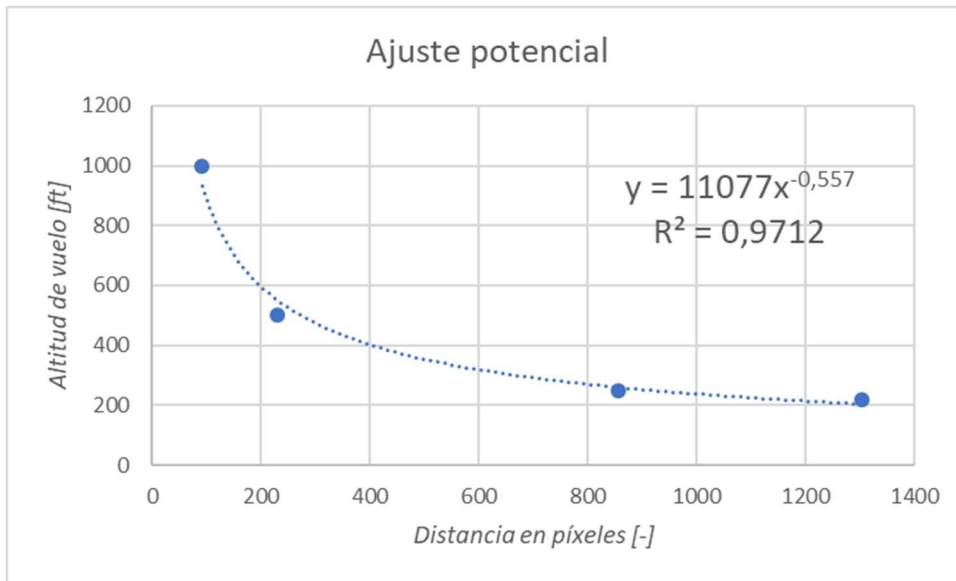


Figura 45. Ajuste potencial de la altitud de vuelo.

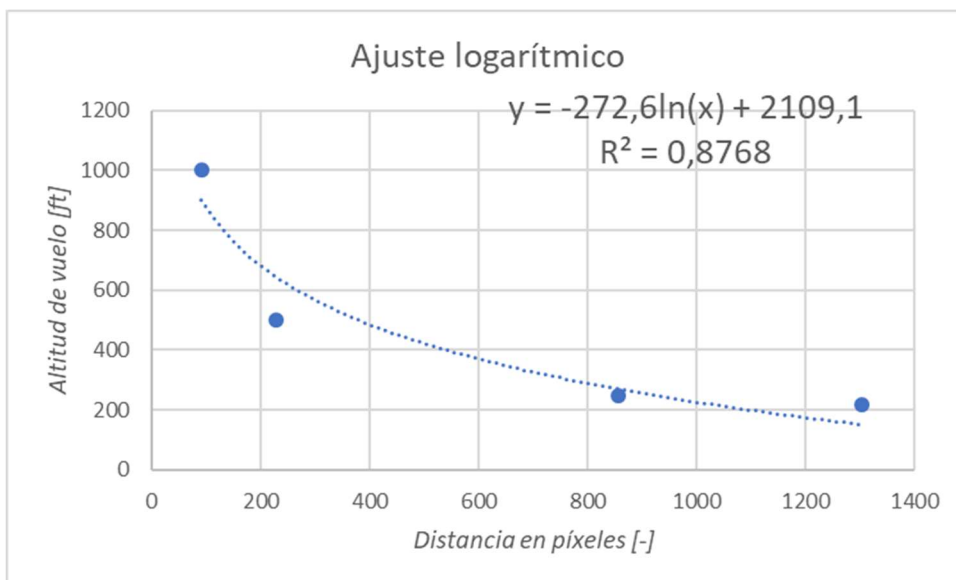


Figura 46. Ajuste logarítmico de la altitud de vuelo.

En vista de los resultados se concluye que el ajuste más adecuado es el ajuste potencial, estableciendo la siguiente ecuación:

$$H = 11077 \cdot D^{-0,557}$$

Donde D es la distancia en píxeles y H la altitud de vuelo en pies.

5. TEST Y RESULTADOS.

En este punto del proyecto se cuenta con un algoritmo que proporciona la distancia en pixeles del Kadett 2400 con respecto al centro de la pista y la altitud de vuelo cuando dicha aeronave se encuentra alineada con la misma.

Sin embargo, el objetivo del proyecto es que el aterrizaje se realice de forma autónoma, reduciendo la distancia entre el UAV y el centro de la pista mediante la variación del roll.

Para lograrlo, se debe disponer de un controlador que realizará las correcciones pertinentes, generando los comandos de movimiento oportunos.

Existen dos estrategias principales para realizar este tipo de operaciones de control:

- Controlador PID: es el mecanismo por excelencia utilizado en el control industrial. Se basa en la realimentación y calcula la desviación entre el valor medido y el valor deseado.

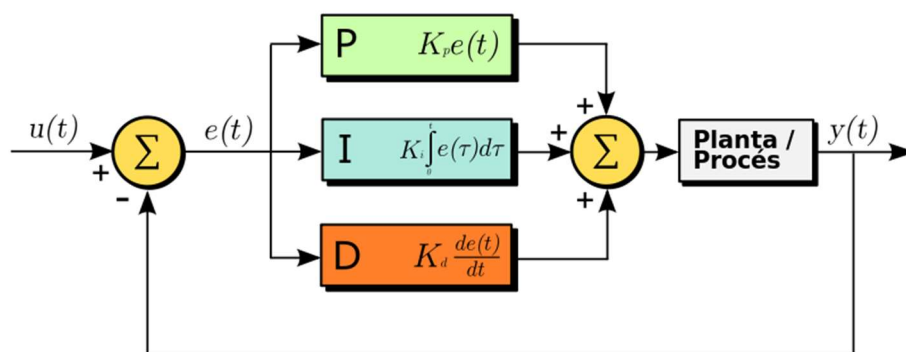


Figura 47. Diagrama de un PID.

Los controladores PID constan de tres módulos: el control proporcional (P), el control integral (I) y el control derivativo (D). Cada una de estas acciones tiene una función determinada en la estabilización del sistema. El valor proporcional depende del error actual, el integral de los errores pasados, y el derivativo es una predicción de errores futuros.

El controlador PID es una buena opción cuando no se conoce el proceso. De esta forma, ajustando las tres variables, el controlador es capaz de predecir una acción de control diseñada para los requerimientos del proceso específico.

- Controlador basado en lógica difusa: este tipo de controlador permite introducir valores intermedios entre la afirmación completa o la negación absoluta, se basa en la lógica multivaluada y es la técnica utilizada en el campo de la Inteligencia Artificial. Estos controladores requieren un coste computacional bajo,

presentando una opción interesante en el campo de los UAV de pequeña envergadura.

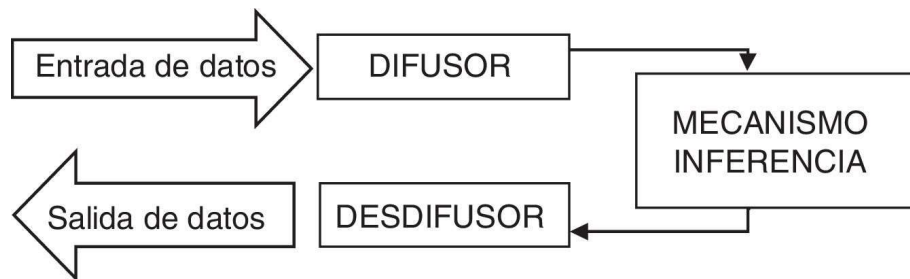


Figura 48. Diagrama de un controlador basado en lógica difusa.

Para su comprensión, el bloque difusor es ubicado de acuerdo con el grado de pertenencia a cada uno de los conjuntos difusos mediante la función característica. Luego se ingresan los datos de la variable a analizar con sus valores concretos y se obtienen como datos de salida los grados de pertenencia a los conjuntos estudiados.

El bloque de inferencia representa las reglas que definirán el sistema y la forma en la que se relacionan los conjuntos difusos de entrada y de salida.

Se entiende por desdifusor el método que permite obtener el resultado concreto con grados de seguridad a partir de los conjuntos difusos, tras aplicar métodos matemáticos de desdifusión (Rico y Tinto, 2008).

Existen diversos entornos de desarrollo que facilitan la creación de los bloques de control difuso mencionados anteriormente, ya que suelen ser sistemas muy complejos de manejar.

Se concluye que la mejor estrategia será diseñar un PID, el cual será ajustado a partir de la función de transferencia.

La función de transferencia vendrá dada por dos entradas, que serán la distancia inferior del centro de la pista al centro de la imagen, es decir al centro de la aeronave; y la distancia superior. La salida, será el roll o ángulo de alabeo en español.

Para obtener la función de transferencia se debe partir de una tabla de valores en la que se relacionen la variación temporal con el comportamiento de las distancias inferior y superior cuando se varía el roll.

Esta tabla de valores se obtendrá mediante una simulación en FlightGear en la que se definirán los Waypoints de la aeronave, los cuales estarán situados sobre la pista de aterrizaje del aeropuerto de Valencia (LEVC).

Dichos Waypoints son los siguientes:

Waypoints		
Latitud	Longitud	Altitud
39.4810361	-0.46198056	200
39.4838222	-0.466089	300
39.4843333	-0.4691278	400
39.4855889	-0.471925	400
39.4900056	-0.482083	300
39.4922194	-0.49048056	300
39.4947944	-0.496325	400

Tabla 7. Waypoints definidos para la simulación.



Figura 49. Ruta seguida en la simulación.

El procesado de la imagen está realizado en un script de Matlab y para poder obtener datos en el tiempo de simulación mientras esta se realiza, se debe incluir un bloque en Simulink que incluya una función que proporcione los valores de las distancias que van obtenidas en cada imagen. Dicho bloque se incorpora de manera que invoque el tratamiento de imagen y almacene para cada tiempo esos valores medidos de las distancias, además del valor del roll en cada momento, presente en el bloque StateVector.

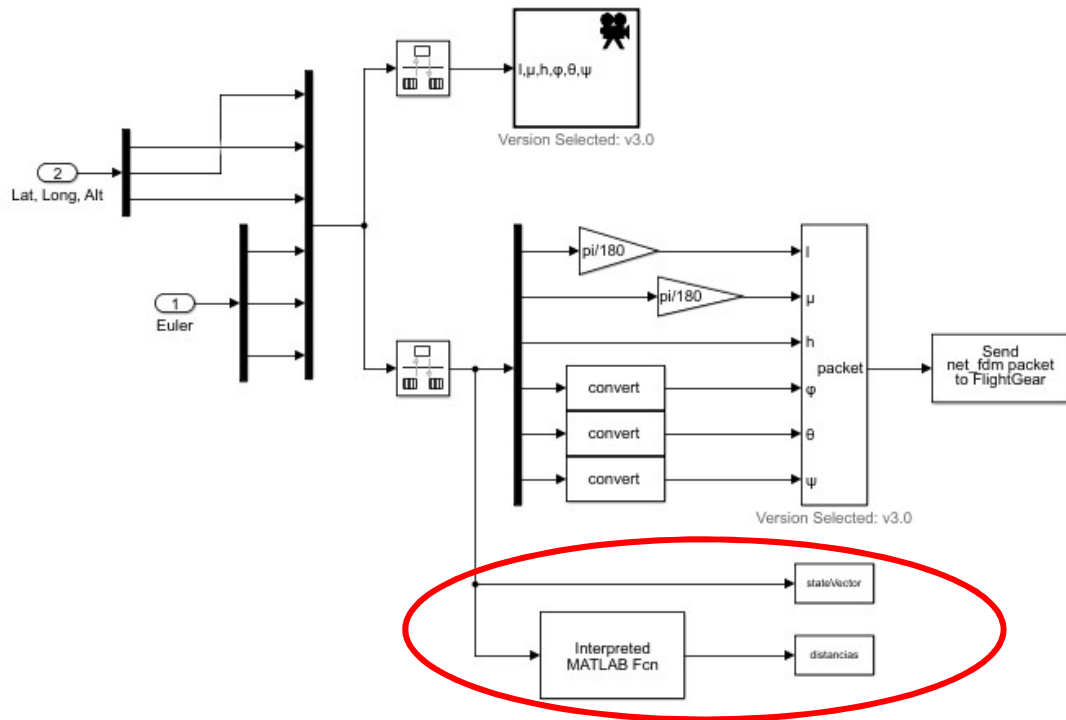


Figura 50. Esquema de Simulink con los bloques mencionados añadidos.

Los valores de las variables implicadas se envían a Matlab tras terminar la simulación mediante bloques “To Workspace”. De esta forma se obtendrá la tabla de valores necesaria para la búsqueda de la función de transferencia que relacione estas variables.

Sin embargo, los resultados de la simulación no fueron del todo satisfactorios, ya que los Waypoints deben estar más alejados y no se puede programar una maniobra “zig-zag” de este modo.

Por lo tanto, se optó por generar un script que, teniendo la aeronave sobre la pista, generara pulsos positivos y negativos en el roll, de manera que se comportara como un entrada escalón. Estos valores fueron introducidos en el FlightGear mediante Simulink, obteniendo de este modo la maniobra que se deseaba sobre la pista.

De esta forma ya se contaba con resultados satisfactorios para comenzar a obtener la función de transferencia del sistema.

En las siguientes figuras se puede observar la variación temporal del Roll durante la simulación, así como la variación de las dos variables de distancia de la aeronave al centro de la pista, la distancia superior y la inferior.

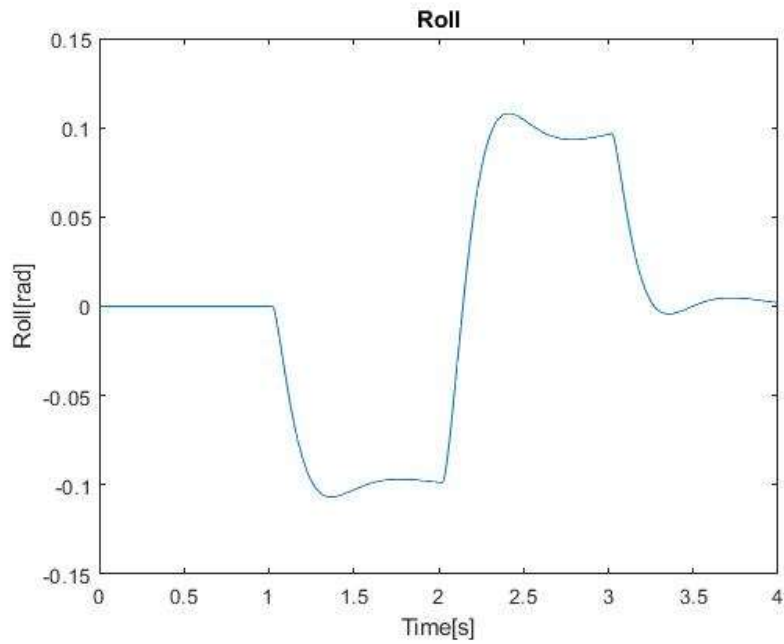


Figura 51. Variación temporal del Roll.

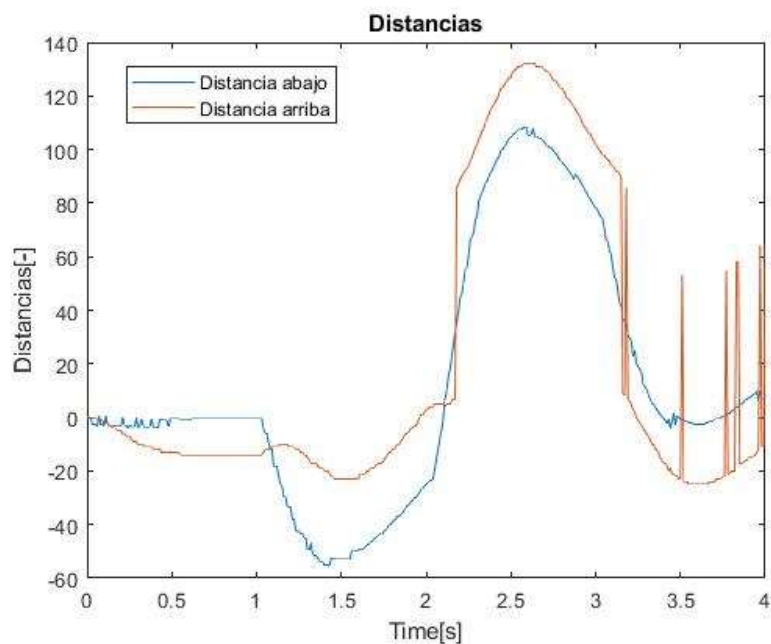


Figura 52. Variación temporal de las Distancias.

Utilizando la herramienta “ident” de Matlab, se seleccionó para cada conjunto de datos las zonas de datos óptimas con las que poder trabajar y a partir de ahí obtener distintas funciones de transferencia, variando el número de polos.

A partir de aquí el trabajo se centrará en orientar el avión en la pista utilizando la distancia inferior en la imagen, ya que el proceso siguiente será alinear la parte superior de la imagen y se hará del mismo modo que esta primera.

A la función de transferencia se le debe añadir un integrador, ya que el sistema con el que estamos trabajando es inestable. Este hecho es comprobable, ya que para un valor

del roll cualquiera la actitud de la aeronave será alejarse de la pista en algún momento, aumentando la distancia con la pista hasta el infinito.

De este modo la misión ahora es calcular el ángulo que hay entre el centro de la pista y nuestra aeronave, pues este dará el nuevo heading que debe seguir la aeronave para iniciar la maniobra de aterrizaje.

Teniendo en cuenta lo explicado y en función de los datos obtenidos se obtiene la función de transferencia del sistema. Se ha probado con un sistema con un integrador y un sistema con un integrador y un polo, obteniendo los siguientes resultados con su respectivo nivel de ajuste tras realizar una validación con los resultados obtenidos de la simulación.

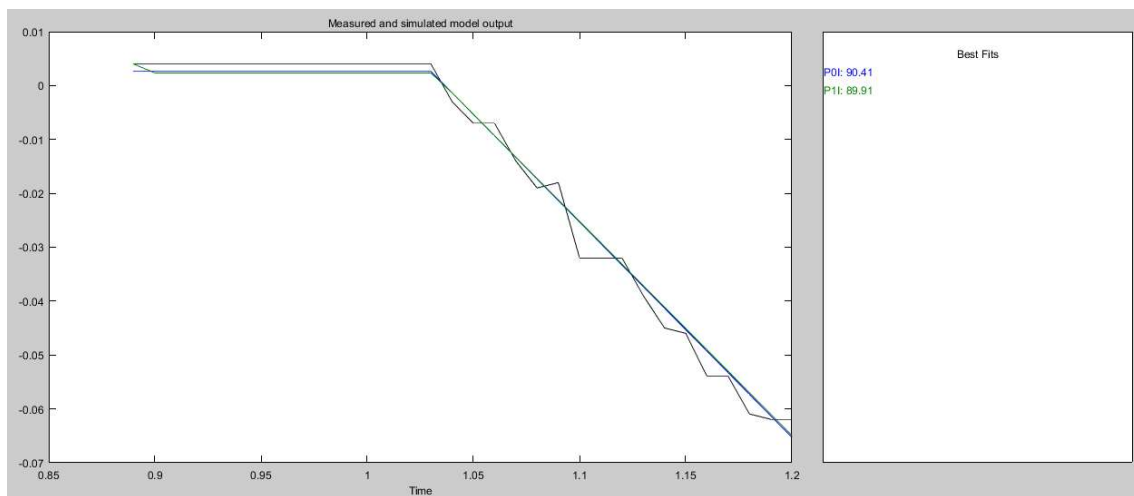


Figura 53. Validación de la función de transferencia usando 'ident'

Como se puede ver los resultados obtenidos tienen un nivel de ajuste alto. Siendo de color azul la respuesta de la función de transferencia son polos y verde la que incorpora un polo. Las funciones de transferencia obtenidas son las que se muestran a continuación:

$$\frac{3.995}{s} \quad y \quad \frac{3.973}{s(0.0006583 s + 1)}$$

Una vez se tienen las dos funciones de transferencia, se debe pasar a la programación del PID, el cual será integrado en Simulink en la parte encargada de controlar el Roll, y que se utilizará cuando comience la maniobra de aterrizaje.

La programación del PID se lleva a cabo mediante la herramienta de Matlab "rltool", es decir, utilizando el lugar de las raíces del sistema a controlar. Se escoge la función de transferencia con un solo integrador, puesto que el nivel de fiabilidad es alto y supone controlar un sistema más simple.

Como requerimientos del sistema se establece que la sobreoscilación sea nula y un tiempo de establecimiento de 2 segundos.

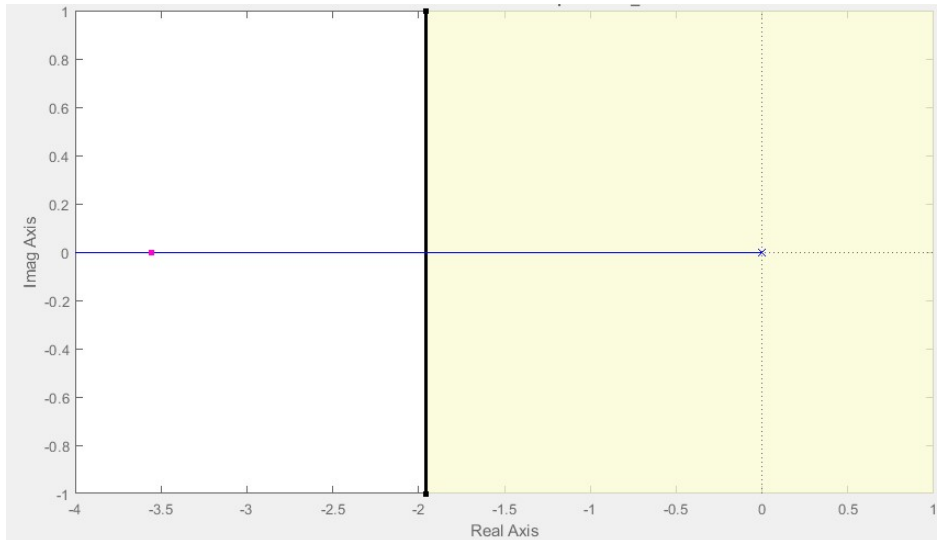


Figura 54. Lugar de las raíces del sistema.

Por lo que únicamente es necesario incluir un proporcional con valor $K = 0.76158$, y cuya respuesta a una entrada tipo escalón es la siguiente:

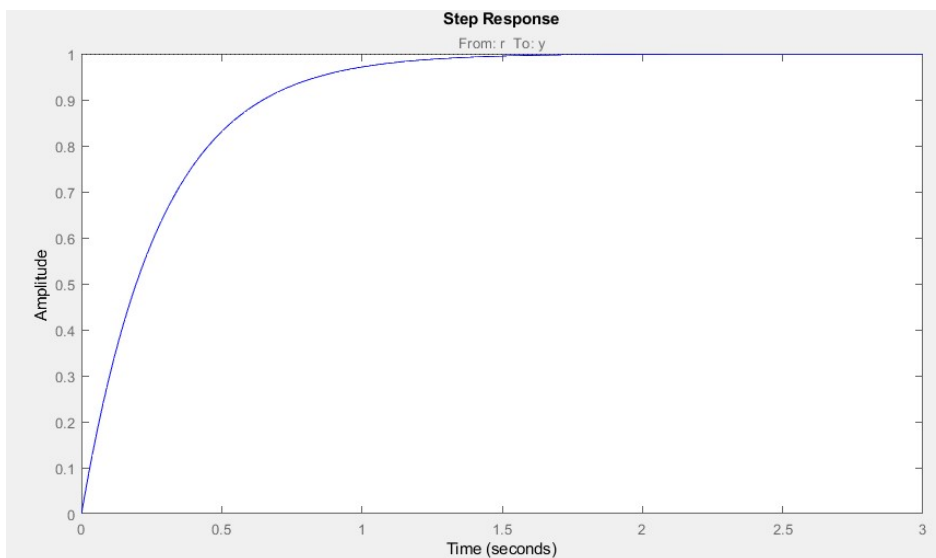


Figura 55. Respuesta del sistema a una entrada tipo escalón.

El control obtenido se incorpora al entorno de Simulink, en el cual se programa la variable del ángulo entre la pista y la aeronave que se utilizará para efectuar el guiado hasta la misma y comenzar la maniobra del descenso.

Además, se incluye en el sistema el PID obtenido para controlar esta variable hasta el valor deseado, que en este caso será 0 puesto que se trata de un ángulo relativo.

El PID se incorpora en la variación del heading, el cual se controla mediante el roll, que es la variable de entrada con la que se ha realizado el control.

Para comprobar la validez del modelo desarrollado, se realiza una simulación, en la que se iniciará una maniobra de aproximación a la pista desde un punto cercano a la misma al que se ha llegado mediante coordenadas GPS.

Para ello, el roll se ha variado en función del ángulo que se reconoce mediante la visión artificial, este ángulo se va reduciendo hasta que se alcanza el valor deseado, el cual es 0 y que resultaría en el avión orientado en la pista.

Se muestra a continuación la variación de este ángulo, α , durante la simulación realizada.

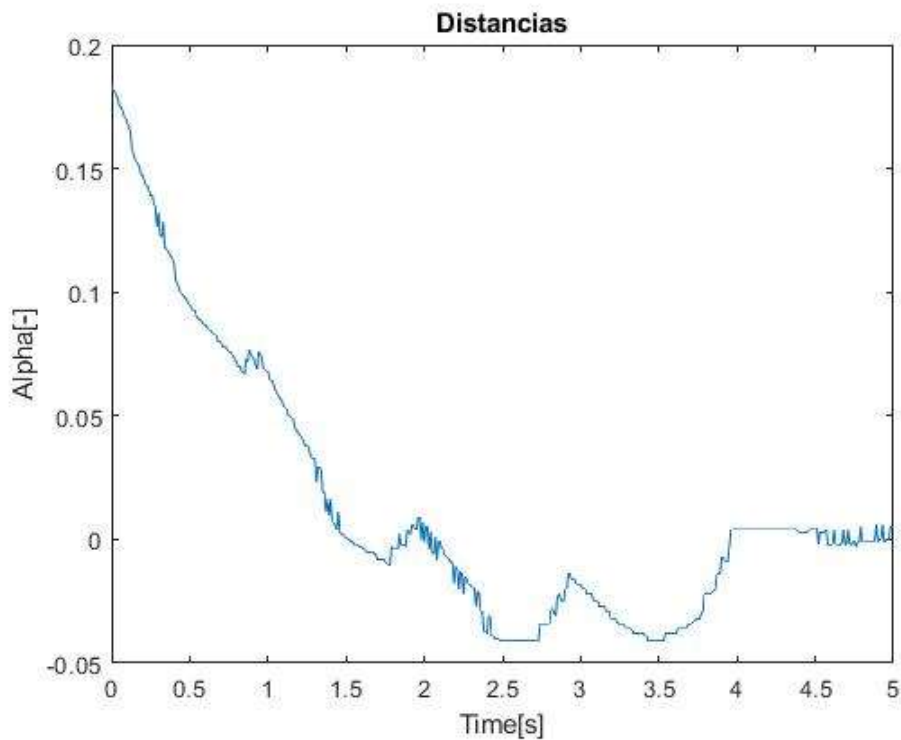


Figura 56. Variación del ángulo a controlar.

Se puede observar que existe un punto en el que el ángulo es 0, pero la aeronave lleva un roll residual que no es 0, por lo que este ángulo vuelve a aumentar, hasta que se consigue su estabilización.

6. CONCLUSIONES

Se concluye tras la realización del trabajo y las diferentes pruebas realizadas que se ha obtenido un sistema de guiado en la maniobra de aterrizaje basado en visión artificial.

Tras la primera familiarización con el entorno de simulación, el tratamiento de imagen y los sistemas existentes basados en la visión artificial, se comenzó a aplicar dicho tratamiento a una imagen que era obtenida mediante la conexión de la plataforma de simulación FlightGear y el sistema desarrollado en Matlab y Simulink de la plataforma de vuelo.

Dicha imagen ha sido procesada mediante diferentes métodos, hasta llegar a la conclusión de que se debe binarizar la imagen hasta encontrar un factor que se pudiera utilizar como referencia. Se concluyó que la mejor referencia a utilizar eran las líneas de la pista, las cuales eran encontradas aplicando el algoritmo de Hough a la imagen tratada previamente.

Gracias a las líneas es posible obtener información utilizable en la creación de un algoritmo que consiga orientar la aeronave en pista y calcular la altitud de vuelo para comenzar el descenso.

Se considera como trabajo futuro la posibilidad de incorporar una nueva variable, el ángulo de cabeceo o pitch, para realizar el descenso y completar el aterrizaje de la aeronave.

Existe una carencia evidente de resultados de pruebas de simulación, las cuales no se han realizado por falta de tiempo, de esta forma se podrá mejorar el algoritmo y probar situaciones más desfavorables desde las que iniciar el aterrizaje.

7. BIBLIOGRAFÍA

- [1] Barrientos, J. del Cerro, P. Gutiérrez, R. San Martín, A. Martínez, C. Rossi. *Vehículos aéreos no tripulados para uso civil. Tecnología y aplicaciones.*
- [2] https://es.wikipedia.org/wiki/Veh%C3%ADculo_a%C3%A9reo_no_tripulado
- [3] https://en.wikipedia.org/wiki/History_of_unmanned_aerial_vehicles
- [4] A. Arévalo. *Vehículos aéreos no tripulados, descripción y capacidades para la obtención de información.*
- [5] J. L. Asensio, F. Pérez, and P. Morán, *UAVs. Beneficios y límites.* I.E.S. Leonardo Da Vinci
- [6] <http://www.designation-systems.net/dusrm/app1/oq-14.html>
- [7] Taylor, John W. R., ed. (1982). *Jane's All The World's Aircraft*
- [8] J. Martinez-Carranza, L. Valentín, F. Márquez-Aquino, J. González-Islas, N. Loewen. *Detección de obstáculos durante vuelo autónomo de drones utilizando SLAM monocular.*
- [9] Joel Hermansson, Andreas Gising, Martin Skoglund, Thomas B. Schön. *Autonomous Landing of an Unmanned Aerial Vehicle.*
- [10] Institut La Garrotxa, *Aplicación práctica de la visión artificial en el control de procesos industriales.*
- [11] Saripalli, S., Montgomery, J. F., & Sukhatme, G. (2003). *Visually-Guided Landing of an Unmanned Aerial Vehicle.* IEEE.
- [12] TravTigerEE. *Controlador PID.* De Wikimedia Commons, el repositorio multimedia libre.
- [13] <http://wiki.flightgear.org/Es>
- [14] <https://www.sbg-systems.com/>
- [15] Baira, I. (2014). *Sistema de Detección, Localización y Seguimiento de objetos dinámicos mediante UAV.* Escuela Técnica Superior de Ingenieros Industriales – UPM.
- [16] Sapiro, G. *Digital Image Processing.* University of Duke.

- [17] Sharp, C. S., Orriid, S., – Sastry, S. S. *A Vision System for Landing o fan Unmanned Aerial Vehicle*. University of California Berkeley.
- [18] Carballo Martinez, R. M. *Desarrollo de software para la ayuda en el aterrizaje autónomo mediante visión por computador*. ETSII. Sevilla: Universidad de Sevilla.



UNIVERSITAT
POLITÀCNICA
DE VALÈNCIA



Escuela Técnica Superior de Ingeniería del Diseño



DISEÑO DE ALGORITMOS DE GUIADO PARA EL ATERRIJAZE AUTÓNOMO DE VEHÍCULOS AÉREOS NO TRIPULADOS BASADO EN VISIÓN ARTIFICIAL.

Presupuesto

Trabajo Final de Grado

Grado en Ingeniería Aeroespacial

Autor: Adrián Ruiz García

Tutor: Sergio García-Nieto Rodríguez

ÍNDICE

1. INTRODUCCIÓN	5
2. MANO DE OBRA	6
3. PRECIOS DE LOS MATERIALES	7
4. PRECIOS POR UNIDADES DE OBRA.....	8
4.1. Labor de investigación	8
4.2. Diseño de la visión artificial.	8
4.3. Implementación en Matlab y Simulink.	8
5. PRESUPUESTO TOTAL.....	9

ÍNDICE DE TABLAS

Tabla 1. Resumen de mano de obra.....	7
Tabla 2. Materiales usados.....	7
Tabla 3. Tareas de investigación.....	8
Tabla 4. Tareas del desarrollo de la visión artificial.....	8
Tabla 5. Tareas de implementación.	8
Tabla 6. Resumen del presupuesto total.....	9

1. INTRODUCCIÓN

Este documento tiene como objetivo detallar el coste de la ejecución del proyecto: “DISEÑO DE ALGORITMOS DE GUIADO PARA EL ATERRIZAJE AUTÓNOMO DE VEHÍCULOS AÉREOS NO TRIPULADOS BASADO EN VISIÓN ARTIFICIAL”.

En primer lugar, se definen los precios de la mano de obra y los materiales que se han utilizado. Después se detallará los precios en cada fase del proyecto, dividiéndolas en:

- Labor de investigación: incluye las horas invertidas en búsqueda de información, artículos, documentación.
- Diseño de la visión artificial.
- Implementación en Matlab y Simulink.

2. MANO DE OBRA

En este apartado se plantea cual sería el salario justo y acorde con la formación de un Ingeniero que realice estas tareas. En este proyecto, se tendrá como referencia al Colegio Oficial de Ingenieros Técnicos Industriales de Bizkaia. Concretamente se ha utilizado un artículo publicado por el colegio de Ingenieros en el año 2014, en el que se calculaban el precio por hora que se debía cobrar por un trabajo. El nombre del artículo es: “¿Sabes calcular el precio por hora de tu trabajo?”.

Se procede a continuación al desglose de los cálculos.

Cálculo de horas anuales (laborales):

40 horas semanales x 52 semanas = 2080 horas anuales

21 días de vacaciones y 14 festivos x 8 horas al día = 280 horas anuales

Total: 1800 horas anuales

También se deben estimar las horas facturables. Se va a suponer que se trabaja para una entidad, por lo que el 100% de las horas serán facturables.

Cálculo del ratio precio/horas base al año:

Estableciendo un salario base de unos 50.000,00 € netos al año, se tiene:

Entre 1800 horas facturables al año

Total: 28,00€/hora

Se debe tener también en cuenta que existen gastos que la empresa debe afrontar. Estos gastos, teniendo en cuenta un blog dedicado a la economía consultado (<http://albertojuven.blogcanalprofesional.es/cuanto-paga-empresa-nomina/>), se pueden desglosar los siguientes gastos:

Gastos:

Aporte de la empresa a la seguridad social: 30.9%

Aporte del trabajador a la seguridad social: 6.35%

Total: 37.25% de impuestos

Por lo tanto, para que el salario neto realmente sea 50000,00€, la empresa deberá asumir un salario bruto que deberá ser calculado, dando el coste bruto por hora calculado a continuación:

$$\frac{5000}{1-3,25\%} = 79.681,00\text{€ brutos.}$$

Entre las 1800 horas trabajadas.

Total: 44,00€/hora

En la siguiente tabla se puede observar un resumen del presupuesto por mano de obra.

Salario anual neto	Horas laborables anuales	Ratio base al año	Gastos de seguridad social	Salario final por hora
50.000,00€	1800 horas	28,00€/hora	30.000€	44,00€/hora

Tabla 1. Resumen de mano de obra.

3. PRECIOS DE LOS MATERIALES

En la tabla 2 se puede observar un desglose de los materiales y licencias que se han utilizado para este proyecto.

Material	Coste/Unidad	Periodo de utilización	Coste/hora
Windows 10 Core i5	700€	3 años = 5400 horas	0,13€/hora
Licencia de software Matlab for Academic Use	500€	1 año = 1800 horas	0,28€/hora
Subtotal			0,31€/hora

Tabla 2. Materiales usados.

Se debe calcular para cada caso el coste/hora, ya que después de realizar este proyecto se podrán seguir utilizando.

4. PRECIOS POR UNIDADES DE OBRA

En este apartado se detalla el tiempo requerido por cada tarea dentro de las unidades de obra que se han reflejado en la introducción, indicando el precio por tarea en base al precio que se obtuvo de mano de obra.

4.1. Labor de investigación

Incluye las tareas de búsqueda de información y documentación, así como la familiarización con el entorno de trabajo.

Tarea	Días	Horas	Mano de obra	Importe total
Estudio inicial de los UAVs	2	16	44,00€	704,00€
Estudio sobre visión artificial	4	32	44,00€	1.408,00€
Familiarización con el entorno de simulación	5	40	44,00€	1.760,00€
Total	11	88	44,00€	3.872,00€

Tabla 3. Tareas de investigación.

4.2. Diseño de la visión artificial.

Tareas realizadas en base al diseño de la visión por computador utilizada en el proyecto.

Tarea	Días	Horas	Mano de obra	Importe total
Conexión FlightGear-Matlab	2	16	44,00€	704,00€
Selección de patrones	2	16	44,00€	704,00€
Tratamiento de imagen	10	80	44,00€	3.520,00€
Implementación cálculos	5	40	44,00€	1.760,00€
Total	19	152	44,00€	6.776,00€

Tabla 4. Tareas del desarrollo de la visión artificial.

4.3. Implementación en Matlab y Simulink.

Finalmente, las tareas llevadas a cabo tanto en Matlab como en el entorno de la plataforma de simulación, Simulink.

Tarea	Días	Horas	Mano de obra	Importe total
Simulaciones y recogida de datos	4	32	44,00€	1.408,00€
Interpretación de resultados y funciones de transferencia	2	16	44,00€	704,00€
Diseño PID	2	16	44,00€	704,00€
Implementación en Simulink	4	32	44,00€	1.408,00€
Total	12	96	44,00€	4.224,00€

Tabla 5. Tareas de implementación.

5. PRESUPUESTO TOTAL

Asunto	Horas	Coste/hora	Coste total
Secciones			
Investigación	88	44€	3.872€
Visión artificial	152	44€	6.776€
Implementación en Matlab y Simulink	96	44€	4.224€
Materiales			
Windows 10 Core i5	336	0,12€/hora	40,32€
Licencia de software Matlab for Academic Use	336	0,28€/hora	94,08€
Presupuesto Total de Ejecución Material			15.006,40€
Gastos generales (15%)			2.250,96€
Beneficio industrial (6%)			900,38€
Presupuesto total de inversión			18.157,74€
IVA (21%)			3.813,13€
TOTAL			21.970,87€

Tabla 6. Resumen del presupuesto total.

De este modo, el coste del proyecto sería 15.006,40€, y ocuparía un total de 336 horas de trabajo. Cabe destacar que el grueso de los gastos proviene del personal, ya que el proyecto no requiere una inversión cuantiosa en cuanto a materiales, siempre que se entienda este proyecto como una sección independiente de la plataforma de vuelo Kadett 2400.



UNIVERSITAT
POLITÀCNICA
DE VALÈNCIA


Escuela Técnica Superior de Ingeniería del Diseño



DISEÑO DE ALGORITMOS DE GUIADO PARA EL ATERRIJAJE AUTÓNOMO DE VEHÍCULOS AÉREOS NO TRIPULADOS BASADO EN VISIÓN ARTIFICIAL.

Pliego de condiciones

Trabajo Final de Grado

Grado en Ingeniería Aeroespacial

Autor: Adrián Ruiz García

Tutor: Sergio García-Nieto Rodríguez

ÍNDICE

1.	DESCRIPCIÓN DE LAS OBRAS.....	5
1.1.	Unidades de obra.....	5
1.1.1.	Desarrollo de un ámbito de investigación.....	5
1.1.2.	Desarrollo e implementación del sistema de aterrizaje	5
2.	CONDICIONES GENERALES	5
2.1.	Disposiciones generales.....	5
2.1.1.	Documentación del contrato de obra	5
2.2.	Condiciones Generales Facultativas	6
2.2.1.	Funciones a desarrollar por el contratista.....	6
2.2.2.	Funciones a desarrollar por el Ingeniero Director	7
2.2.3.	Libro de Órdenes	7
2.3.	Condiciones Generales de la Ejecución	7
2.3.1.	Ritmo de las obras	7
2.3.2.	Orden de los trabajos	8
2.3.3.	Ampliación del Proyecto por Causas Imprevistas	8
2.3.4.	Prórroga por Causas de Fuerza Mayor	8
2.3.5.	Condiciones Generales de Ejecución de los trabajos	8
2.3.6.	Trabajos Defectuosos	8
2.3.7.	Pruebas y Ensayos	8
2.3.8.	Recepción	8
2.4.	Condiciones Generales Económicas	9
2.4.1.	Principio General	9
2.4.2.	Fianzas	9
2.4.3.	Precios	10
2.4.4.	Valoración de los trabajos	11
2.4.5.	Penalizaciones	12
2.5.	Condiciones Legales Generales.....	12
2.5.1.	El Contratista	12
2.5.2.	El contrato	13
2.5.3.	Arbitraje.....	13
3.	CONDICIONES PARTICULARES.....	13
3.1.	Desarrollo de un ámbito de investigación.....	13
3.1.1.	Investigación teórica de la visión artificial.	13

3.1.2.	Implementación en Matlab de las herramientas de Visión artificial.	14
3.2.	Desarrollo e implementación del sistema de aterrizaje.....	14
3.2.1.	Desarrollo del sistema de visión artificial.....	14
3.2.2.	Implementación.....	14

1. DESCRIPCIÓN DE LAS OBRAS

En el presente documento se exponen las especificaciones técnicas, legales y económicas para la ejecución del proyecto desarrollado.

1.1. Unidades de obra

Las obras a ejecutar se dividen en:

- Desarrollo de un ámbito de investigación.
- Desarrollo e implementación del sistema de aterrizaje.

1.1.1. Desarrollo de un ámbito de investigación

Esta unidad engloba el trabajo desarrollado para el ámbito de investigación, es decir, la parte puramente teórica del proyecto, en el que se ha desarrollado un sistema de visión artificial para la aeronave. Principalmente habrá dos unidades en este grupo:

- Investigación de la visión artificial.
- Implementación en Matlab.

1.1.2. Desarrollo e implementación del sistema de aterrizaje

Esta unidad de obra consta de la creación de un sistema de visión artificial para el Kadett 2400, y su posterior implementación usando Matlab y Simulink. Por lo que las partes bien diferenciadas de esta unidad son:

- Desarrollo de un sistema de visión artificial.
- Implementación en Matlab y Simulink.

2. CONDICIONES GENERALES

2.1. Disposiciones generales

Durante este apartado se pretende regular la ejecución de las obras, distinguiendo y delimitando las funciones que corresponden al constructor, al dueño de la obra, al ingeniero, así como las relaciones ente todos ellos. En él se recogen los aspectos legales del proyecto y se fijan las condiciones que regirán la ejecución y puesta en marcha de este: características de los materiales, técnicas a emplear en la ejecución, controles de calidad exigidos, normas y leyes que rigen el proyecto. Constituye la especificación de carácter facultativo, económico y legal que regirá el desarrollo de las obras.

2.1.1. Documentación del contrato de obra

El contrato de obra se encuentra formado por los siguientes documentos:

- Condiciones fijadas en el documento de contrato.
- Pliego de condiciones técnicas particulares.
- Pliego general de condiciones.
- El resto de documentación del proyecto (memoria, manual de programación, manual de usuario y presupuesto).

Las órdenes e instrucciones de la dirección de obra se incorporan al proyecto como interpretación o precisión de este. En cada documento las especificaciones literales prevalecen sobre las gráficas.

2.2. Condiciones Generales Facultativas

En este apartado se describen y regulan las relaciones entre la contrata y la dirección facultativa para ejecución de las obras.

2.2.1. Funciones a desarrollar por el contratista

Derechos y obligaciones:

- Conocer las leyes y verificar los documentos del proyecto. El constructor debería indicar que la documentación del proyecto le resulta suficiente para la comprensión de la obra o solicitar aclaraciones pertinentes.
- Elaborar el plan de seguridad e higiene para la aprobación por parte de la dirección facultativa.
- El constructor debe comunicar a la propiedad la persona designada como delegado suyo en la obra la cual asumirá las funciones planas del constructor.
- El jefe de obra, o sus encargados, deben estar presente durante la jornada de trabajo y acompañar al ingeniero en las visitas a las obras, poniéndose a disposición de estos y suministrando los datos precisos para la comprobación de mediciones y liquidaciones.
- La contrata debe ejecutar los trabajos necesarios para la buena construcción y aspecto de las obras, aun cuando estos no se hallen expresamente determinados, siempre que lo disponga el ingeniero dentro de los límites de las posibilidades de los presupuestos. Requiere reformado del proyecto con consentimiento de la propiedad toda variación que suponga el incremento de los precios de alguna unidad de obra en más del 20 % del total del presupuesto en un 10 %.
- Las aclaraciones, interpretaciones y modificaciones de los preceptos del pliego de condiciones o de las indicaciones de los planos se comunicará por escrito al constructor, debiendo este devolver los originales comunicando el enterado mediante su firma al pie de todas las instrucciones, aviso u órdenes que reciba.
- El constructor podrá requerir del ingeniero o ingeniero técnico cuantas instrucciones o aclaraciones precise para la correcta ejecución del proyecto. Asimismo, recibirá solución a los problemas técnicos no previstos en el proyecto que se presenten durante la ejecución de las obras.
- Las reclamaciones del contratista contra ordenes o instrucciones de la dirección facultativa las presentará ante la propiedad a través del ingeniero, si son de orden económico y de acuerdo con las condiciones estipuladas en el pliego de condiciones correspondiente. Contra disposiciones de orden técnico no podrá interponer reclamación alguna, pudiendo el contratista a fin de salvar su responsabilidad exponer razonablemente al ingeniero, el cual puede limitar su contestación al acuse de recibo.
- El contratista no podrá recusar al ingeniero o ingeniero técnico o personal encargado por estos para el control de las obras, ni pedir la designación de otros facultativos para los reconocimientos o mediciones.

- En caso de desobediencia, manifiesta incompetencia o negligencia grave que comprometa o perturben la marcha de las obras, el ingeniero puede requerir al contratista que aparte a los operarios causantes de la perturbación.
- El contratista puede subcontratar capítulos o unidades de obra con sujeción a lo estipulado en el pliego de condiciones y sin perjuicio de sus obligaciones como contratista general de la obra.
- El contratista no iniciará una unidad de obra sin la autorización de la dirección. El contratista está obligado a cumplir las indicaciones del libro de órdenes.

2.2.2. Funciones a desarrollar por el Ingeniero Director

Es el máximo responsable de la ejecución del proyecto, decide sobre el comienzo, ritmo y calidad de los trabajos. Velará por el cumplimiento de estos y por las condiciones de seguridad del personal de la obra. Las funciones que corresponden al ingeniero director son:

- Redactar los complementos o rectificaciones del proyecto que precisen.
- Asistir a las obras las veces que la naturaleza y complejidad de las mismas requieran a fin de resolver las contingencias que se produzcan e impartir las instrucciones necesarias.
- Coordinar la intervención en obra de un tercero.
- Aprobar certificaciones parciales de obra, la liquidación final y asesorar al promotor en el acto de recepción.
- Comprobar instalaciones provisionales, medios auxiliares y sistemas de seguridad e higiene en el trabajo.
- Ordenar y dirigir la ejecución con arreglo al proyecto, normas técnicas y reglas de la buena construcción.
- Realizar o disponer de las pruebas y ensayos de materiales, instalaciones y demás unidades de obra según el plan de control, así como los controles necesarios para asegurar la calidad constructiva de acuerdo con el proyecto y la normativa técnica aplicable. Informar al constructor de los resultados de las pruebas e impartir, en su caso, las órdenes oportunas.
- Planificar el control de calidad y el control económico de las obras.

2.2.3. Libro de Órdenes

Es obligatoria la existencia a pie de obra de un libro de órdenes e incidencias, visado por los colegiados profesionales correspondientes donde se recogerán las órdenes y modificaciones que se dicten en cada momento.

2.3. Condiciones Generales de la Ejecución

2.3.1. Ritmo de las obras

El instalador o contratista iniciará las obras en el plazo marcado por el pliego de condiciones particulares, desarrollándolas para que queden ejecutados los trabajos dentro de los periodos parciales establecidos y así ejecutar la obra dentro del plazo exigido en el contrato. El contratista dará cuenta por escrito al ingeniero del comienzo de los trabajos con, al menos, tres días de antelación.

2.3.2. Orden de los trabajos

La determinación del orden de los trabajos es facultad de la contrata, salvo en casos que por circunstancias técnicas estime conveniente su variación la dirección facultativa.

2.3.3. Ampliación del Proyecto por Causas Imprevistas

Cuando haya que ampliar el proyecto, bien por motivos imprevistos o razones de fuerza mayor, no se interrumpirán los trabajos, continuándose según las instrucciones dadas por el ingeniero en tanto se formula o tramita el proyecto reformado. El constructor deberá realizar los trabajos necesarios de carácter urgente, anticipando ese servicio, el cual le será consignado en un presupuesto adicional o abonado directamente.

2.3.4. Prórroga por Causas de Fuerza Mayor

Si por causas de fuerza mayor o independiente de la voluntad del constructor no pudiesen iniciarse las obras, o fuesen suspendidas o no se acabasen en los plazos prefijados, se otorgará una prórroga para el cumplimiento de la contrata previo informe favorable del ingeniero.

2.3.5. Condiciones Generales de Ejecución de los trabajos

Los trabajos se ejecutarán con estricta sujeción al proyecto, a las modificaciones de este que hayan sido aprobadas y a las órdenes e instrucciones que entreguen por escrito bajo su responsabilidad el ingeniero o ingeniero técnico.

2.3.6. Trabajos Defectuosos

El constructor deberá emplear materiales que cumplan las condiciones exigidas en las condiciones técnicas generales y particulares del pliego de condiciones y realizar los trabajos de acuerdo con lo especificado en el pliego. Hasta la recepción definitiva es el responsable de la ejecución y de los defectos derivados de una mala ejecución. Por esto cuando el ingeniero técnico advierta defectos en los trabajos, o que los materiales o los aparatos colocados no reúnen las condiciones exigibles, entonces antes de la recepción de la obra podrá disponer la reposición de las partes defectuosas.

2.3.7. Pruebas y Ensayos

Los gastos ocasionados por pruebas y ensayos corren por cuenta de la contrata, pudiéndose repetir aquellos que no ofrezcan las suficientes garantías. Los ensayos para cada instalación se encuentran especificados en el capítulo de cada instalación.

2.3.8. Recepción

- Recepción provisional: tres días antes de dar fin a las obras, el ingeniero comunicará a la propiedad la proximidad de la terminación a fin de convenir la fecha para el acto de recepción provisional. Esta se realiza con la participación de la propiedad, el ingeniero. Se practicará un detenido reconocimiento de las obras, se extenderá un acta para cada interviniente firmada por todos ellos. Desde esta fecha corre el periodo de garantía si las obras son admitidas. Seguidamente los técnicos de la dirección facultativa extenderán el certificado de final de obra. En caso de estar acabado se darán las instrucciones para remediar los defectos, fijando un plazo para subsanarlos, pasado el cual se hará un nuevo reconocimiento.

- Documentación final de la obra: el ingeniero director facilitara a la propiedad la documentación final con las especificaciones y contenidos por la legislación vigente.
- Recepción definitiva: se verificará después del plazo de garantía, el cual se especifica en el pliego de condiciones particulares y nunca será inferior a nueve meses. El formulismo será el mismo que para la provisional. A partir de esta fecha cesa la obligación del constructor de reparar desperfectos propios de la normal conservación de la obra.
- Prórroga de la garantía: si la obra no se encuentra en las condiciones debidas se aplazará la recepción definitiva, el ingeniero director indicará al constructor los plazos para realizar las obras necesarias. De no cumplirse estos plazos perderá la fianza.
- Recepción de trabajos cuya contrata ha sido rescindida: en caso de resolución del contrato, el contratista deberá retirar las herramientas, medios auxiliares, etc. En el plazo indicado en el pliego de condiciones y dejar la obra en condiciones de ser reanudada por otra empresa. Los trabajos terminados se recibirán provisionalmente, y definitivamente una vez transcurrido el periodo de garantía.

2.4. Condiciones Generales Económicas

2.4.1. Principio General

En este apartado se describen y regulan las relaciones económicas entre la propiedad y la contrata, así como la función de control de la dirección facultativa. Todos los intervinientes en el proceso de construcción tienen derecho a percibir puntualmente las cantidades devengadas por su correcta actuación con arreglo a las condiciones contractuales establecidas. La propiedad, el contratista y los técnicos pueden exigirse recíprocamente las garantías adecuadas al cumplimiento puntual de sus obligaciones de pago.

2.4.2. Fianzas

La fianza es el porcentaje del valor total de las obras que debe depositar la contrata como garantías a la firma del contrato. El contratista prestará las siguientes fianzas:

- Depósito en metálico o aval bancario por importe del 10% del precio total de contrata, salvo especificación contraria en el contrato.
- Retención de un 5% en las certificaciones parciales o pagos que se van librando.

Con cargo a la fianza se aplican las penalizaciones por demoras y las reparaciones con cargo a la contrata. Si el contratista se negase a realizar los trabajos precisos para ultimar la obra en las condiciones contratadas, el ingeniero en representación del propietario ordenará su ejecución a un tercero, abonando su importe con la fianza depositada, sin perjuicio de las acciones que el propietario lleve a cabo en caso de que el importe de la fianza no cubra el importe de los gastos. La fianza será devuelta al contratista en un plazo inferior a treinta días después de firmada el acta de recepción definitiva de la obra. La propiedad podrá exigir que el contratista acredite la liquidación y finiquito de sus deudas causadas por la ejecución de esta.

2.4.3. Precios

Composición de los precios: el cálculo de los precios es el resultado de sumar los costes directos, los indirectos, los gastos generales y el beneficio industrial.

Los costes directos son:

- Mano de obra con pluses, cargas y seguros sociales que intervienen directamente.
- Los materiales a los precios resultantes a pie de obra que sean necesarios para su ejecución.
- Equipos y sistemas técnicos de seguridad e higiene para prevención y protección de accidentes.
- Gastos de personal, combustible y energía derivados del funcionamiento de la maquinaria e instalaciones utilizadas en la ejecución de la unidad de obra.
- Gastos de amortización y conservación de la maquinaria, instalaciones, sistemas y equipos.

Los costes indirectos son:

- Gastos de instalación de oficinas a pie de obra, comunicaciones, edificación de almacenes, talleres, seguros, personal administrativo adscrito a la obra. Se cifran como porcentaje de los costes directos e indirectos.

El beneficio industrial:

- El beneficio del contratista se establece en un 6 % sobre la suma de las anteriores partidas.

Precio de ejecución material.

- El resultado obtenido por la suma de las anteriores partidas exceptuando el beneficio industrial.

El precio de contrata:

- Es la suma de costes directos, indirectos y beneficio industrial. El IVA se aplica a este precio, pero no lo integra.

Precios Contradictorios: se producen cuando la propiedad mediante el ingeniero introduce unidades o cambios de calidad en algunas de las unidades previstas o bien es necesario afrontar situaciones imprevistas. El contratista está obligado a efectuar los cambios. El precio se resolverá entre el contratista y el ingeniero antes de comenzar la ejecución de los trabajos. Si el contratista no reclama los precios antes de la firma del contrato, no podrá reclamar aumento de los precios fijados en el cuadro de presupuesto que sirve de base para la ejecución.

Revisión de los precios contratados: no se admite la revisión de los precios si el incremento de estos en las unidades que faltan por realizar no alcanza un montante superior al del presupuesto del contrato. Ante variaciones al alza se efectúa la revisión de acuerdo con la fórmula establecida en el pliego de condiciones particulares. El

contratista percibe la diferencia que resulte por la variación del IPC superior a alguna de las fórmulas de revisión propuestas en la ley de contratos del estado.

2.4.4. Valoración de los trabajos

Formas de abono: salvo indicación contraria en el pliego de condiciones particulares el abono de los trabajos se efectuará en una de las siguientes formas:

- Tipo fijo tanto alzado por unidad de obra, con el precio invariable fijado de antemano, pudiendo variar únicamente el número de unidades ejecutadas previa medición y aplicando al total de unidades de obra ejecutadas el precio fijado.
- Tanto variable por unidad de obra, según las condiciones en que se realice y los materiales empleados en su ejecución de acuerdo con las órdenes del ingeniero director.
- Mediante listas de jornales y recibos de materiales realizados en la forma que el pliego general de condiciones económicas determina.
- Por hora de trabajo según las condiciones determinadas en el contrato.

Certificaciones: lo ejecutado se valorará aplicando al resultado de la medición de los precios señalados en el presupuesto por cada una de ellas, considerando además lo establecido en el pliego general de condiciones económicas respecto a mejoras o sustituciones de material. El contratista puede presenciar las mediciones necesarias para la elaboración de la relación, así mismo, el ingeniero técnico enviará al contratista los resultados de las mediciones para que este lo examine y devolverlos firmados con su conformidad o efectuar las oportunas reclamaciones. El ingeniero aceptará o rechazará las reclamaciones dando cuenta al contratista de su resolución, pudiendo este reclamar al propietario contra la resolución del ingeniero. A partir de la relación valorada, el ingeniero expedirá la certificación de obras ejecutadas. La certificación se remitirá al propietario en el periodo de un mes posterior al que referencia la certificación y tendrá el carácter de documento sujeto a variaciones derivadas de la liquidación final, no suponiendo dichas certificaciones aprobación ni recepción de las obras que comprenden.

Pagos: los pagos los efectuará el propietario en los plazos previamente establecidos y su importe corresponderá al de las certificaciones de obra conformadas por el ingeniero director.

Trabajos Ejecutados Durante el Periodo de Garantía: el abono de estos trabajos se llevará a cabo de la siguiente forma:

- Si los trabajos están especificados en el proyecto y no fueron realizados a su debido tiempo serán valorados a los precios que se figuren en el presupuesto y abonados de acuerdo a lo establecido en el proyecto.
- Si los trabajos tienen como objeto la reparación de desperfectos ocasionados por el uso, estos se abonarán a los precios del día previamente acordados.

- Si el objeto de los trabajos es la reparación de desperfectos ocasionados por deficiencias de la instalación o de la calidad de los materiales, no se abonará nada al contratista.

2.4.5. Penalizaciones

Hay tres tipos de penalizaciones, por demora en la ejecución, por incumplimiento de contrato y por demora en los pagos.

- Por retraso no justificado en el plazo de terminación: la indemnización por retraso no justificado en la terminación de las obras se establecerá en un 10 por mil del importe total de los trabajos contratados por cada día natural de retraso a partir del día de terminación fijado. Esta cantidad será descontada y retenida con cargo a la fianza. No se computan como días perdidos los debidos a razones de fuerza mayor como huelgas, catástrofes o causas administrativas.
- Penalización por incumplimiento de contrato: se establecerá en las condiciones del contrato una penalización por incumplimiento del mismo o mala ejecución de los trabajos.
- Demora en los pagos: si el propietario no efectuase el pago dentro del mes siguiente al plazo convenido, el contratista tendrá el derecho de percibir el abono de un cuatro y medio por ciento anual en concepto de intereses de demora. Si pasasen dos meses a partir del término de dicho plazo, el contratista tiene derecho a la resolución del contrato, procediéndose a la liquidación de las obras ejecutadas y materiales acopiados.

2.5. Condiciones Legales Generales

Ambas partes comprometen en sus diferencias al árbitro de amigables componedores.

2.5.1. El Contratista

El contratista es el responsable de la ejecución de las obras en las condiciones establecidas en el contrato y en los documentos del proyecto excluida la memoria. Por tanto, está obligado a deshacer y volver hacer todo lo mal ejecutado durante las obras, aunque estas partidas hayan sido abonadas. Asimismo, se obliga a lo establecido en la ley de contratos de trabajo y dispuesto en la de accidentes de trabajo, subsidio familiar y seguros sociales. El contratista se responsabiliza de los accidentes que se produzcan por inexperiencia o descuido donde se efectúan las obras y en las contiguas. Será el único responsable y correrá de su cuenta el abono de las indemnizaciones puesto que se incluyen en los precios los gastos para seguros y medidas de seguridad. Corre a cargo de la contrata el pago de impuestos y arbitrios municipales cuyo abono deba efectuarse durante el tiempo de ejecución de las obras y por concepto inherente a las obras. El contratista tiene derecho, a su costa, de sacar copias de los manuales de programación, presupuestos, pliego de condiciones y demás documentos del proyecto.

Serán causas de rescisión del contrato:

- Muerte o incapacidad del contratista.
- La quiebra del contratista.

- Alteraciones de contrato debidas a modificaciones del proyecto con alteraciones fundamentales a juicio del ingeniero director, y siempre que alguna modificación represente como mínimo el 40% del valor de alguna de las unidades del proyecto modificadas.
- No comenzar los trabajos dentro del plazo señalado en las condiciones del contrato o del proyecto.
- El incumplimiento de las condiciones del contrato cuando suponga descuido o mala fe con perjuicio de los intereses de la obra.
- El abandono de la obra sin causa justificada.

2.5.2. El contrato

El contrato se establece entre a propiedad o promotor y el contratista. Existen varias modalidades:

- A precio alzado, se estipula una cantidad para las obras que no se modificará, aunque el volumen de las obras se modifique. Sirve para obras pequeñas.
- Contrato por unidades de obra.

2.5.3. Arbitraje

En casos de litigio o desavenencia entre la propiedad y la contrata se recurrirá en primer lugar a la Dirección Facultativa de la obra. En caso de que el desacuerdo subsista, cada parte nombrará un perito, sometiéndose las partes al acuerdo entre estos. En última instancia se acudirá a los tribunales.

3. CONDICIONES PARTICULARES.

Se procede a describir las características que han de tener los materiales, los controles de calidad, así como las pruebas y ensayos a los que van a ser sometidos.

Se han organizado las obras de la siguiente manera:

- Desarrollo de un ámbito de investigación
 - Desarrollo de la visión artificial.
 - Validación de los modelos desarrollados.
- Desarrollo e implementación del sistema de aterrizaje
 - Desarrollo de un sistema de visión artificial.
 - Implementación en Matlab y Simulink.

3.1. Desarrollo de un ámbito de investigación

Se desarrollará al comenzar el proyecto una labor de investigación con la que se describa el funcionamiento de la visión artificial. Para ello se desarrollarán dos partes: la investigación teórica de la visión artificial y su implementación en Matlab.

3.1.1. Investigación teórica de la visión artificial.

Esta parte es puramente teórica, lo que requerirá la lectura de un gran número de artículos de diversas fuentes. Los requisitos son los siguientes:

- Amplia bibliografía.
- Comparativa de distintos sistemas basados en visión artificial.
- Manejo con la herramienta Image Processing Toolbox de Matlab.

3.1.2. Implementación en Matlab de las herramientas de Visión artificial.

Tras familiarizarse con las herramientas de las que se dispone y la tarea a realizar se debe implementar el tratamiento de imagen para recabar la información necesaria de esta. Los requisitos son los siguientes:

- Selección de la imagen.
- Implementación de herramientas de procesado de imagen.
- Selección de patrones útiles en la imagen.
- Información útil.

3.2. Desarrollo e implementación del sistema de aterrizaje.

Trata de diseñar el algoritmo que permita el aterrizaje autónomo y su implementación en el sistema desarrollado por el grupo de Optimización y Heurística de la Universidad Politécnica de Valencia.

3.2.1. Desarrollo del sistema de visión artificial.

Los requisitos de este apartado serán:

- Utilización de la información obtenida para los cálculos.
- Desarrollo del algoritmo.
- Simulación para la toma de datos.
- Selección de los datos útiles y almacenamiento.

3.2.2. Implementación

Una vez recabada la información de los experimentos se tienen los siguientes requisitos:

- Obtención de la función de transferencia del sistema.
- Diseño de un PID que controle las variables de interés.
- Inclusión del sistema en el modelo desarrollado.