

## PROYECTO FIN DE GRADO

**TÍTULO:** Optimización de algoritmos de videoseguimiento de objetivos móviles con cámaras PTZ

**AUTOR/A:** Gonzalo Salgado Rivera

**TITULACIÓN:** Grado en Ingeniería Telemática

**DIRECTOR/A:** José María Peña Espartero

**TUTOR/A:** Javier Martín Rueda

**DEPARTAMENTO:** Departamento de Ingeniería Telemática y Electrónica

VºBº TUTOR/A

**Miembros del Tribunal Calificador:**

**PRESIDENTE/A:** María Luisa López Ibáñez

**TUTOR/A:** Javier Martín Rueda

**SECRETARIO/A:** Eloy Portillo Aldana

**Fecha de lectura:** 15-07-2024

**Calificación:**

El Secretario/La Secretaria,



---

## Resumen

Los algoritmos de control de movimiento en sistemas de video seguimiento son fundamentales para mantener los objetivos en su campo de visión y hacer posible su identificación. El éxito de estos algoritmos depende de su capacidad para mantener el blanco centrado dentro de su campo de visión el máximo tiempo posible garantizando una estabilidad óptima.

En este Trabajo de Final de Grado se diseña un algoritmo de control orientado al seguimiento de objetivos móviles mediante cámaras PTZ (Pan-Tilt-Zoom) y una metodología de evaluación de prestaciones y comparación entre distintos algoritmos.

En primer lugar, se genera un entorno simulado que permite evaluar las prestaciones de distintos algoritmos de control en las mismas condiciones para diferentes escenarios. Los escenarios representan distintas dinámicas de blancos dentro del mismo entorno.

Se diseña un algoritmo con el objetivo de mejorar las prestaciones de los algoritmos empleados actualmente. La mejora del algoritmo se sostiene sobre principios lógicos basados en la minimización de la diferencia entre la posición del objetivo y el centro de la imagen. La implementación se realiza sobre un sistema de seguimiento de drones y aves, por lo que se diseñan escenarios, y se ajustan los parámetros configurables del algoritmo, basándose en este tipo de dinámicas.

En la fase de pruebas, se somete el algoritmo propuesto junto con otros algoritmos empleados para este tipo de aplicación, a los diferentes escenarios planteados, empleando los mismos elementos externos, como cámaras y elementos de software, que se utilizan en los sistemas reales. A partir de los resultados obtenidos en las simulaciones se realiza una comparativa entre respuestas según la metodología establecida para la evaluación de prestaciones de los algoritmos.

Finalmente, se pone a prueba la viabilidad del algoritmo propuesto en un sistema en operación, a través de la evaluación de una serie de procesos de vídeo seguimiento.

**Palabras clave:** Video seguimiento, Algoritmos, PTZ, PID, Lógica difusa

---

---

## Abstract

Motion control algorithms in video tracking systems are critical to keep targets in their field of view and make identification possible. The success of these algorithms depends on their ability to keep the target centered within their field of view for as long as possible while ensuring optimal stability.

In this Final Degree Project, a control algorithm for tracking moving targets using PTZ (Pan-Tilt-Zoom) cameras is designed, in addition to a methodology for performance evaluation and comparison between different algorithms.

First, a simulated environment is generated to evaluate the performance of different control algorithms under the same conditions for different scenarios. The scenarios represent different target dynamics within the same environment.

An algorithm is designed with the aim of improving the performance of the currently used algorithms. The improvement of the algorithm is based on logical principles based on the minimization of the difference between the target position and the center of the image. The implementation is performed on a drone and bird tracking system, so scenarios are designed, and the configurable parameters of the algorithm are adjusted, based on this type of dynamics.

In the testing phase, the proposed algorithm, together with other algorithms used for this type of application, is subjected to the different scenarios proposed, using the same external elements, such as cameras and software elements, that are used in real systems. Based on the results obtained in the simulations, a comparison is made between responses according to the methodology established for the evaluation of the performance of the algorithms.

Finally, the viability of the proposed algorithm is tested in an operating system, through the evaluation of a series of video monitoring processes.

**Key words:** Video tracking, Algorithms, PTZ, PID, Fuzzy logic

---

---

## Índice de figuras

Figura 1.1. Elementos de una video detección.....	1
Figura 2.1. Diagrama de un controlador PID realimentado [4].....	7
Figura 2.2. Respuesta típica de un sistema PID en lazo cerrado [5].....	7
Figura 2.3. Respuesta al escalón [11].....	9
Figura 2.4. Oscilaciones del sistema [11].....	10
Figura 2.5. Señal de control sin restricciones [1].....	11
Figura 2.6. Señal de control limitada [16].....	11
Figura 2.7. Diagrama de compensación en un controlador PID [19].....	12
Figura 2.8. Modelo de control predictivo [24].....	13
Figura 2.9. Funciones de membresía [32].....	15
Figura 2.10. Diagrama de toma de decisiones en dominio difuso.....	16
Figura 4.1. Diagrama del sistema.....	22
Figura 4.2. Cámara AXISQ6265-LE PTZ [37].....	23
Figura 4.3. Escenario real.....	24
Figura 4.4. Escenario simulado.....	25
Figura 4.5. Software de mando y control.....	26
Figura 4.6. Software de pruebas.....	26
Figura 4.7. Software de visión artificial.....	27
Figura 4.8. Entorno de simulación.....	28
Figura 4.9. Diagrama de secuencia de la prueba.....	29
Figura 4.10. Medición de latencia.....	30
Figura 4.11. Diagrama del controlador PID.....	33
Figura 4.12. Oscilación del sistema.....	34
Figura 4.13. Aplicación antiwindup sobre el controlador.....	35
Figura 4.14. Diagrama de flujo de la acción antiwindup.....	35
Figura 4.15. Diagrama del controlador Fuzzy PID.....	36
Figura 4.16. Diferencia temporal entre acciones de control.....	38
Figura 4.17. Diferencia temporal entre acciones de control.....	38
Figura 4.18. Tasa de cambio del error a lo largo del tiempo.....	39
Figura 4.19. Funciones de pertenencia triangular [39].....	40
Figura 4.20. Función de pertenencia triangular modificada [39].....	41
Figura 5.1. Respuesta del controlador PID.....	44
Figura 5.2. Respuesta del controlador Fuzzy PID.....	45
Figura 5.3. Comparativa de respuestas.....	45
Figura 5.4. Respuesta del controlador PID.....	47
Figura 5.5. Respuesta del controlador Fuzzy PID.....	48
Figura 5.6. Comparativa de respuestas.....	48
Figura 5.7. Respuesta del controlador PID.....	50
Figura 5.8. Respuesta del controlador Fuzzy PID.....	51
Figura 5.9. Comparativa de respuestas.....	51

---

Figura 5.10. Captura del software de vídeo .....	53
Figura 5.11. Diferencia temporal .....	54
Figura 5.12. Movimiento sobre el eje de Pan .....	55
Figura 5.13. Estado del FoV .....	55
Figura 5.14. Estado del seguimiento y error de seguimiento .....	56
Figura 5.15. Movimiento sobre el eje de Pan .....	57
Figura 5.16. Estado del FoV .....	57
Figura 5.17. Estado del seguimiento y error de seguimiento .....	58

---

## Índice de tablas

Tabla 4.1. Parámetros PID .....	34
Tabla 4.2. Conjuntos difusos .....	37
Tabla 4.3. Parámetros difusos de entrada .....	40
Tabla 4.4. Parámetros difusos de salida .....	40
Tabla 4.5. Regla difusa para Kp [41] .....	42
Tabla 4.6. Regla difusa para Ki [41] .....	42
Tabla 4.7. Regla difusa para Kd [41] .....	42
Tabla 5.1. Parámetros de control .....	43
Tabla 5.2. Resultados simulación A .....	46
Tabla 5.3. Resultados simulación B .....	49
Tabla 5.4. Resultados simulación C .....	52
Tabla 5.5. Resultados pruebas de campo .....	56
Tabla 5.6. Resultados pruebas de campo .....	58
Tabla 6.1. Recursos hardware .....	59
Tabla 6.2. Recursos software .....	60
Tabla 6.3. Presupuesto final .....	60



---

## Lista de acrónimos

- **FoV** Field of View
- **FPS** Frames per second
- **GRPC** Google Remote Proto Call
- **ONVIF** Open Network Video Interface Forum
- **PTZ** Pan-Tilt-Zoom
- **PV** Process variable
- **RTSP** Real Time Stream Procotol
- **SP** Set Point
- **YOLO** You Only Look Once



---

# Índice de contenidos

<b>Resumen</b> .....	<b>i</b>
<b>Abstract</b> .....	<b>iii</b>
<b>Índice de figuras</b> .....	<b>v</b>
<b>Índice de tablas</b> .....	<b>vii</b>
<b>Lista de acrónimos</b> .....	<b>ix</b>
<b>1. Introducción</b> .....	<b>1</b>
1.1 Marco y motivación del proyecto.....	1
1.2 Objetivos técnicos y académicos .....	3
1.3 Estructura del resto de la memoria .....	3
<b>2. Marco tecnológico</b> .....	<b>5</b>
2.1 Controlador PID .....	6
2.1.1 Sintonización del PID .....	8
2.1.2 Windup .....	10
2.1.3 Feedforward .....	12
2.1.4 Ventajas y desventajas.....	12
2.2 Controlador MPC .....	13
2.2.1 Modelo del proceso.....	13
2.2.2 Función de costo.....	14
2.2.3 Ventajas y desventajas.....	14
2.3 Controladores de lógica difusa .....	15
2.3.1 Funciones de membresía.....	15
2.3.2 Fuzzificación y defuzzificación .....	16
2.3.3 Reglas difusas .....	16
2.3.4 Ventajas y desventajas.....	17
<b>3. Especificaciones y restricciones de diseño</b> .....	<b>19</b>
3.1 Especificaciones .....	19
3.2 Restricciones de diseño .....	19
<b>4. Descripción de la solución propuesta</b> .....	<b>21</b>
4.1 Componentes del sistema .....	22
4.1.1 Cámara PTZ.....	22
4.1.2 Módulo de visión artificial.....	23
4.1.3 Módulo de control.....	23
4.2 Definición del escenario y entorno de pruebas.....	24
4.3 Desarrollo del algoritmo .....	31
4.3.1 Desarrollo del controlador PID .....	31
4.3.2 Técnica Antiwindup .....	34
4.3.3 Aplicación de lógica difusa .....	36
<b>5. Resultados</b> .....	<b>43</b>
5.1 Simulación A: Respuesta al impulso .....	43

---

5.2	Simulación B: Desplazamiento y maniobras .....	46
5.3	Simulación C: Cambios de maniobra con velocidad máxima.....	49
5.4	Pruebas de campo.....	52
<b>6.</b>	<b>Presupuesto.....</b>	<b>59</b>
6.1	Recursos humanos .....	59
6.2	Recursos hardware.....	59
6.3	Recursos software .....	60
6.4	Presupuesto final.....	60
<b>7.</b>	<b>Impacto del proyecto .....</b>	<b>61</b>
<b>8.</b>	<b>Conclusiones .....</b>	<b>63</b>
8.1	Conclusiones.....	63
8.2	Trabajos futuros .....	64
<b>9.</b>	<b>Referencias .....</b>	<b>65</b>
<b>Anexo</b>	<b>.....</b>	<b>71</b>
A.1	Datos generados .....	71
A.1.1.	Parámetros de la cámara .....	71
A.1.2.	Datos del seguimiento.....	72
A.1.3.	Datos del controlador difuso.....	72

# 1. Introducción

## 1.1 Marco y motivación del proyecto

La identificación, clasificación y seguimiento de amenazas se encuentra presente en muchos sistemas como pueden ser los sistemas de vigilancia costera y de fronteras, sistemas antidrón, sistemas de control de avifauna u otras especies...

Actualmente uno de los únicos métodos fiables para realizar la evaluación de potenciales amenazas es mediante elementos visuales. Por tanto, se requiere un recurso multimedia que será evaluado por una persona o por un modelo de inteligencia artificial basado en imágenes. Se debe garantizar para la correcta identificación que los datos necesarios lleguen al elemento de clasificación y sean adecuados para que pueda realizar su función. En el caso de clasificadores visuales serán necesarias imágenes o vídeos que capturen el objetivo con la mayor nitidez posible.

Se diferencian por tanto en este contexto dos elementos:

- Un sistema optrónico con capacidad de control de posición (PTZ - Pan-Tilt-Zoom) que proporcione un flujo de video o imágenes.
- Un elemento clasificador que identifica potenciales amenazas en las imágenes recibidas y les asigna una clasificación.

El seguimiento de amenazas se realiza a través del control de movimiento de la unidad PTZ y se basa en la información recibida de un detector, que proporciona la posición del objetivo dentro de la imagen. El algoritmo de control debe evaluar la posición recibida y tratar de mantener el objetivo centrado en la imagen para lograr un seguimiento óptimo.

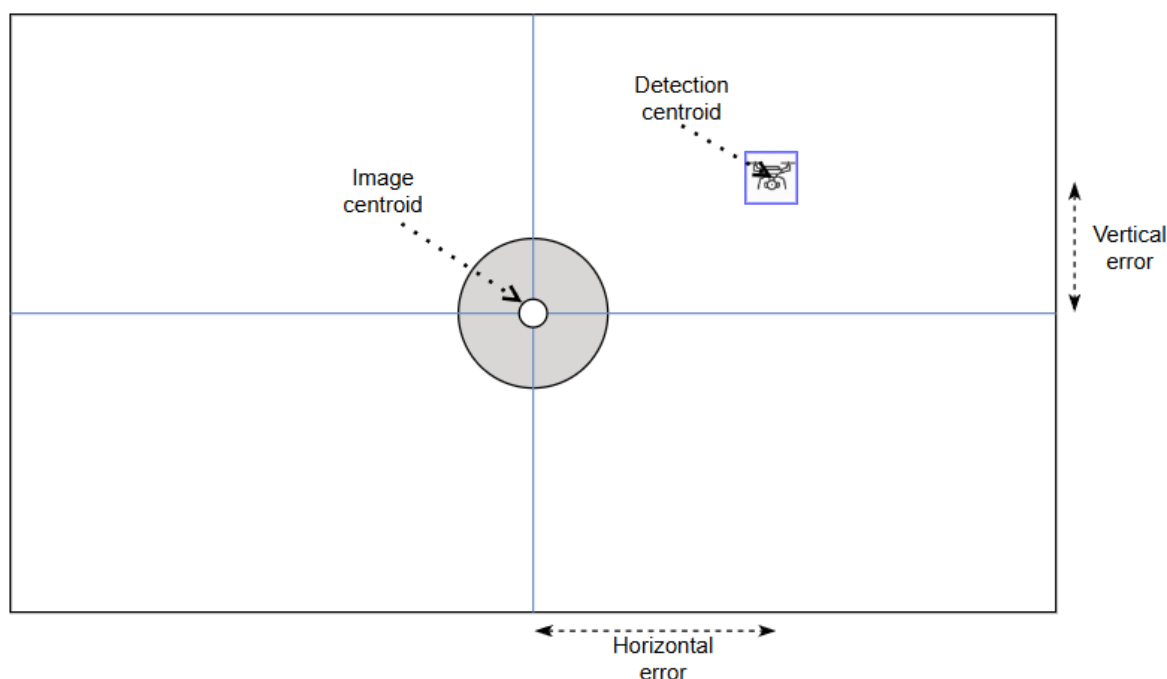


Figura 1.1. Elementos de una video detección

Los avances tecnológicos de las últimas décadas favorecen la aparición de objetivos de mayor velocidad y menor tamaño que requieren de algoritmos más reactivos capaces de anticiparse a cambios bruscos de velocidad o trayectoria. Algunos ejemplos podrían ser lanchas de gran velocidad en aplicaciones de vigilancia costera o drones ligeros que pueden alcanzar grandes velocidades en sistemas de vigilancia aérea.

Los algoritmos de control empleados para el seguimiento son habitualmente algoritmos pensados para controlar un sistema del que se desconoce su respuesta. Se pueden emplear también algoritmos basados en la respuesta del sistema, pero plantean la necesidad de obtener un modelo preciso de la respuesta del sistema, que puede variar en distintas condiciones al ser dependiente de la dinámica del blanco y hace que este tipo de controladores pueda no ser adecuado para esta aplicación o requiera de una optimización que lo ajuste al problema que se debe abordar.

Este proyecto analiza algunos de los algoritmos de control actuales, ajustando sus parámetros para aplicaciones de vigilancia de objetivos móviles a gran distancia y trata de implementar una mejora para el caso de estudio del seguimiento de objetivos móviles mediante cámaras PTZ.

Dado que el estudio se centra en los algoritmos de control, para la realización del proyecto se empleará un detector genérico y una cámara PTZ dentro del mismo escenario.

Durante la realización del proyecto, se distinguen dos fases de evaluación de algoritmos:

- Una primera fase de simulación, en la cual se realizan pruebas de los algoritmos en escenarios simulados previamente definidos y donde se realimenta el algoritmo para realizar los ajustes necesarios.
- Una vez evaluados los distintos algoritmos, se realizan pruebas en campo con datos reales y se extraen métricas para determinar que el algoritmo es consistente y robusto.

Todas las pruebas realizadas parten de una posición en la cual el objetivo se encuentre dentro del campo de visión de la cámara. Esto puede hacerse de forma manual o mediante algún dispositivo como un radar, que permita realiza el apuntamiento inicial de la cámara.

## 1.2 Objetivos técnicos y académicos

El objetivo principal de este proyecto consistirá en definir un algoritmo de control que suponga una mejora frente a los algoritmos actuales en el campo del seguimiento de objetivos móviles.

En aplicaciones de vigilancia, las altas velocidades y el pequeño tamaño de los objetivos hace que sean necesarios algoritmos de control de unidades PTZ muy reactivos con el fin de mantener el objetivo dentro del campo de visión.

Para lograr el objetivo principal del trabajo, se definen los siguientes hitos a alcanzar:

- Selección de un algoritmo de control e implementación de este.
- Definición de un entorno de trabajo que permita simular fácilmente distintos escenarios para evaluar los algoritmos.
- Ajuste de los algoritmos actuales y el algoritmo propuesto.
- Extracción de métricas para evaluar el algoritmo propuesto y los algoritmos actuales.
- Evaluación de los resultados obtenidos para los distintos escenarios simulados y realimentación de la fase de ajuste para optimizar el algoritmo.
- Evaluación en campo de la consistencia y robustez del algoritmo propuesto en aplicaciones reales.

## 1.3 Estructura del resto de la memoria

Este proyecto se estructura en nueve capítulos:

- Capítulo 1. Introducción: Se exponen la motivación y los objetivos del proyecto.
- Capítulo 2. Marco tecnológico: Descripción de soluciones actuales que se emplean para abordar el problema propuesto.
- Capítulo 3. Especificaciones y restricciones de diseño: Especificaciones técnicas para el desarrollo del proyecto y restricciones.
- Capítulo 4. Desarrollo de la solución: Descripción de la solución propuesta detallando su implementación y ajuste de parámetros.
- Capítulo 5. Resultados: Se exponen los experimentos planteados y se evalúan los resultados obtenidos.
- Capítulo 6. Presupuestos: Detalle de los costos de realización del proyecto.
- Capítulo 7. Impacto del proyecto: Impacto del proyecto en varios ámbitos.
- Capítulo 8. Conclusiones: Conclusiones tras la realización del proyecto y líneas de trabajo futuras.
- Capítulo 9. Referencias: Referencias bibliográficas consultadas.



## 2. Marco tecnológico

Los sistemas Pan-Tilt son sistemas de posicionamiento de movimiento en dos ejes y se utilizan para orientación y movimiento de dispositivos. El eje de Pan controla el movimiento sobre un plano horizontal y el eje de tilt controla el movimiento sobre un plano vertical. [1]

Los sistemas Pan-Tilt se emplean habitualmente en aplicaciones de percepción espacial controlada para aportar movilidad a la obtención de información del entorno y al rango de operación del sistema. Algunas de estas aplicaciones son:

- **Elementos ópticos:** Todo tipo de cámaras como pueden ser cámaras de seguridad, de monitorización de especies aviarias o en producción de películas, telescopios para observación astronómica o sensores ópticos para orientación de robots.
- **Defensa:** Apuntamiento de torretas con armas o dispositivos de inhibición de frecuencia.
- **Industria y robótica:** Para el movimiento de elementos robóticos en procesos industriales.

Algunas aplicaciones pueden incluir una componente de profundidad pudiendo controlar también el Zoom, son los denominados sistemas PTZ o Pan-Tilt-Zoom, entre los que se encuentran muchos sistemas de cámaras.

Los sistemas Pan-Tilt deben moverse con rapidez, suavidad y precisión, por lo que los controladores deben ser precisos y robustos. [2] Entre los principios básicos del control Pan-Tilt para el control de cámaras destacan:

- **Control de posición:** Controlar la posición del sistema y la posición que se busca alcanzar periódicamente.
- **Control de velocidad:** Ajustar la velocidad de movimiento del Pan-Tilt para evitar movimientos bruscos e inestabilidad.
- **Compensación de errores:** Realimentar el controlador con cada escaneo del estado del sistema y ajustar sus parámetros para obtener la máxima precisión.

En el marco de utilización de sistemas de control de Pan-Tilt se deben tener en cuenta ciertas consideraciones externas que pueden afectar al funcionamiento adecuado del controlador. Se debe garantizar la nivelación adecuada de la plataforma sobre la que se instalan, ya que pequeños errores angulares pueden resultar en grandes errores a mucha distancia, así como la correcta calibración entre el Pan-Tilt y el dispositivo que proporciona la detección del objetivo, pudiendo ser este un modelo de visión artificial o un dispositivo independiente como un radar. Para el caso de los modelos de visión artificial es importante que garanticen una detección estable del objetivo para el correcto funcionamiento del algoritmo. Las inestabilidades en otros bloques del sistema afectarán al correcto funcionamiento del controlador. [3]

A continuación, se exponen una serie de soluciones de control empleadas habitualmente para este tipo de sistemas.

## 2.1 Controlador PID

Un controlador PID (Proporcional, Integral, Derivativo) es un elemento de control realimentado que controla la variable de un proceso ("Process variable" o PV) a lo largo del tiempo tratando de minimizar el error con respecto al valor de consigna conocido como "Set Point" (SP). El error en un instante de tiempo se expresa matemáticamente según la ecuación de la figura siguiente. [4]

$$e(t) = SP - PV(t) \quad (\text{Ec. 1})$$

El controlador PID cuenta con tres bloques que generan distintas acciones: El bloque proporcional, el bloque integral y el bloque derivativo. En la caracterización del controlador PID se definen los valores de las constantes  $K_p$  (proporcional),  $K_i$  (integral) y  $K_d$  (derivativa) que se emplearán para el cálculo de cada uno de los términos.

El ajuste de la variable de proceso se obtiene de la suma de la acción de cada uno de los tres bloques: [5][6]

- **Proporcional:** La salida del controlador es proporcional al error. Una constante proporcional alta aumentará la velocidad de respuesta del sistema, pero si su valor es muy alto provocará oscilaciones en la variable de proceso. [7]

$$u(t) = K_p * e(t) \quad (\text{Ec. 2})$$

- **Integral:** La salida del controlador es proporcional al error acumulado. El término integral aumentará con el tiempo mientras exista error, pero puede saturar el controlador provocando el efecto de "Windup". [8]

$$u(t) = K_i * \int_0^t e(\tau) d\tau \quad (\text{Ec. 3})$$

- **Derivativa:** La salida del controlador es una acción predictiva sobre la tasa de cambio del error. Disminuye la salida ante aumentos rápidos de la variable de proceso controlando inestabilidad y oscilaciones, pero si la tasa de refresco del control es muy lenta generará inestabilidad.

$$u(t) = K_d * \frac{de(t)}{dt} \quad (\text{Ec. 4})$$

La figura siguiente muestra el diagrama de un controlador PID.

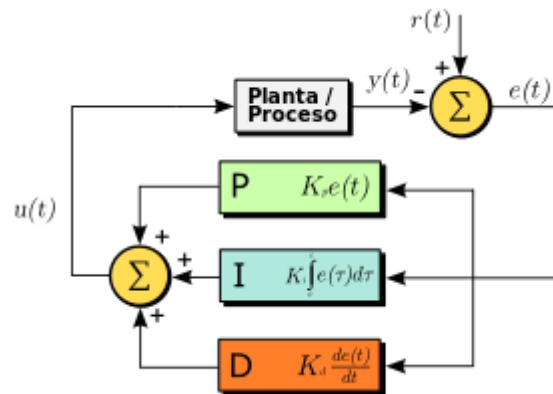


Figura 2.1. Diagrama de un controlador PID realimentado [4]

Los requisitos de rendimiento de un controlador PID se miden con la respuesta del controlador ante cambios de estado. Habitualmente se aplica una función escalón al sistema y se mide el estado de la variable de proceso para extraer ciertas métricas: [5]

- **Rise time** o tiempo de incremento: tiempo que tarda el sistema en pasar del 10% al 90% del SP.
- **Overshoot** o porcentaje de sobre impulso: porcentaje en que la variable de proceso excede el valor final del SP.
- **Settling time** o tiempo de estabilización: es el tiempo que transcurre hasta que la variable de proceso se estabiliza dentro de un umbral en torno al SP (habitualmente del 5%).
- **Steady-State-Error** o error de estado estable: es el error que permanece entre la variable de proceso y el SP una vez estabilizado el sistema.

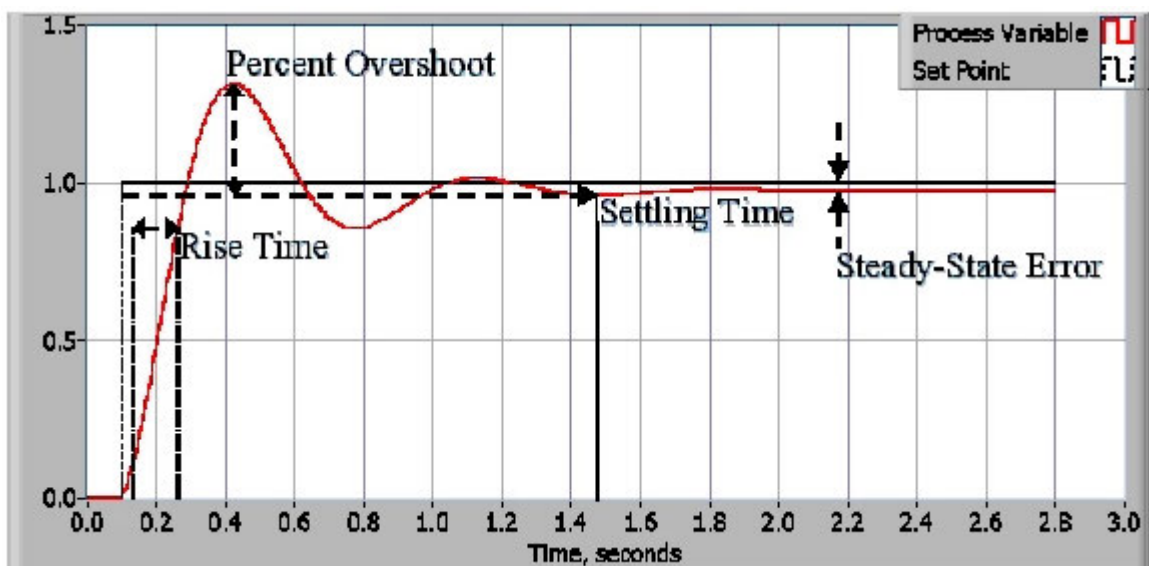


Figura 2.2. Respuesta típica de un sistema PID en lazo cerrado [5]

Otro tipo de criterios para la evaluación del rendimiento del controlador comúnmente utilizados son los errores integrales. Este tipo de evaluación puede realizarse sobre la operación normal del sistema, sin necesidad de evaluar específicamente su respuesta al impulso. A continuación, se definen algunos de los criterios integrales. [9] [10]

$$IAE: \int_0^{\infty} |e(t)| dt \quad (\text{Ec. 5})$$

$$ITAE: \int_0^{\infty} t|e(t)| dt \quad (\text{Ec. 6})$$

$$IAE: \int_0^{\infty} |e(t)| dt \quad (\text{Ec. 7})$$

$$ISE: \int_0^{\infty} e(t)^2 dt \quad (\text{Ec. 8})$$

$$ITSE: \int_0^{\infty} te(t)^2 dt \quad (\text{Ec. 9})$$

$$ISTE: \int_0^{\infty} t^2 e(t)^2 dt \quad (\text{Ec. 10})$$

### 2.1.1 Sintonización del PID

El ajuste de las constantes proporcional, integral y derivativa ( $K_p$ ,  $K_i$ ,  $K_d$ ) es conocido como sintonización del PID. Existen numerosos métodos para el ajuste de parámetros de un PID, como el método de Tyreus y Luyben o el método de Cohen y Coon. Uno de los métodos comúnmente utilizados para la sintonización de controladores PID es el método de Ziegler-Nichols que se puede ajustar midiendo la respuesta del sistema en lazo abierto o en lazo cerrado.

Para el ajuste en lazo abierto se aplica una señal escalón de entre el 10% y el 20% del valor nominal de entrada como señal de control en lugar del controlador PID y se mide el tiempo de respuesta. [11]

Se traza una línea tangente en el cambio de pendiente de la señal de salida del sistema y se define el tiempo de retardo o tiempo muerto ( $L$ ) como el tiempo transcurrido desde la subida del escalón hasta el corte entre la recta tangente y el valor inicial del sistema. Por otro lado, se obtiene también el tiempo de subida ( $T$ ), que es el tiempo transcurrido desde el punto anterior y el punto que corta la recta tangente en el valor máximo de la respuesta del sistema. [12]

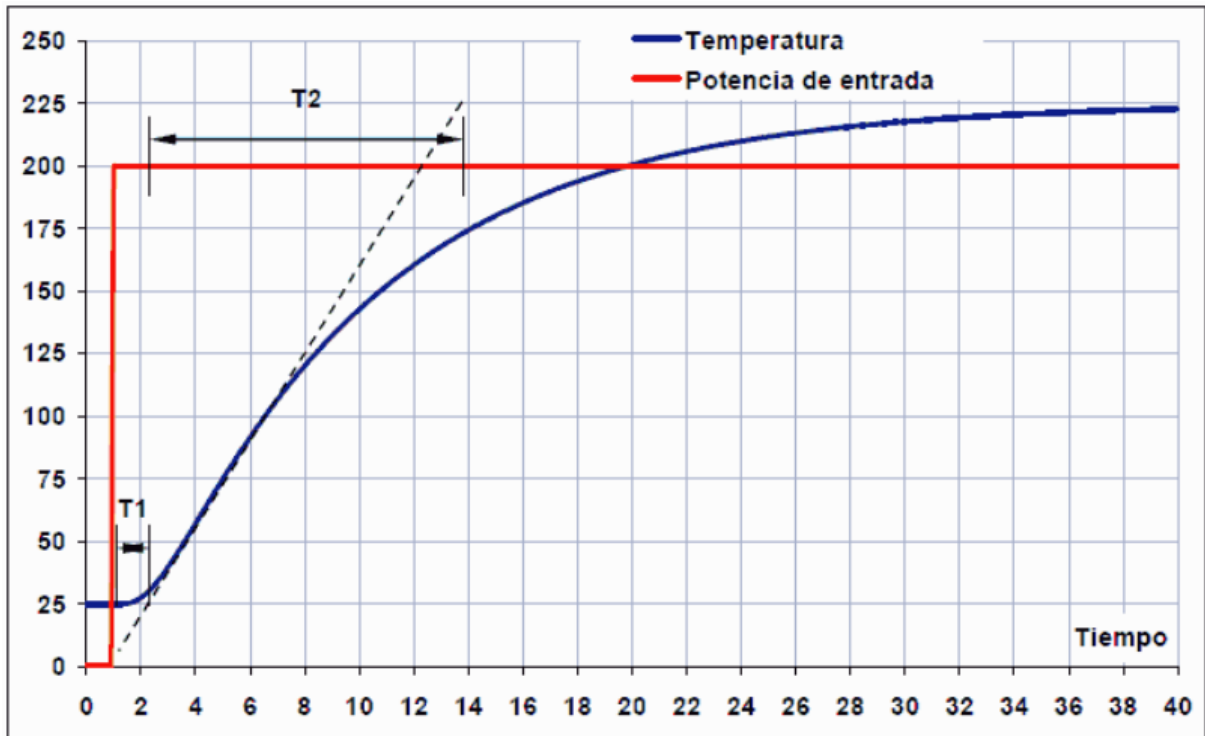


Figura 2.3. Respuesta al escalón [11]

Una vez obtenidos estos valores se emplean las ecuaciones de Ziegler-Nichols para obtener las constantes de un controlador PID. [13]

$$K_P = 1,2 * \frac{T}{L} \quad (\text{Ec. 11})$$

$$K_i = \frac{0,5}{L} \quad (\text{Ec. 12})$$

$$K_d = 0,5 * L \quad (\text{Ec. 13})$$

Para el caso de ajuste en lazo cerrado se reducen los términos integral y derivativo al mínimo y se aumenta el término proporcional hasta que el sistema oscile de forma constante. El valor de  $K_p$  que provoca este efecto se conoce como ganancia crítica ( $K_c$ ) y su periodo de oscilación como  $T_c$ . [11] [14]

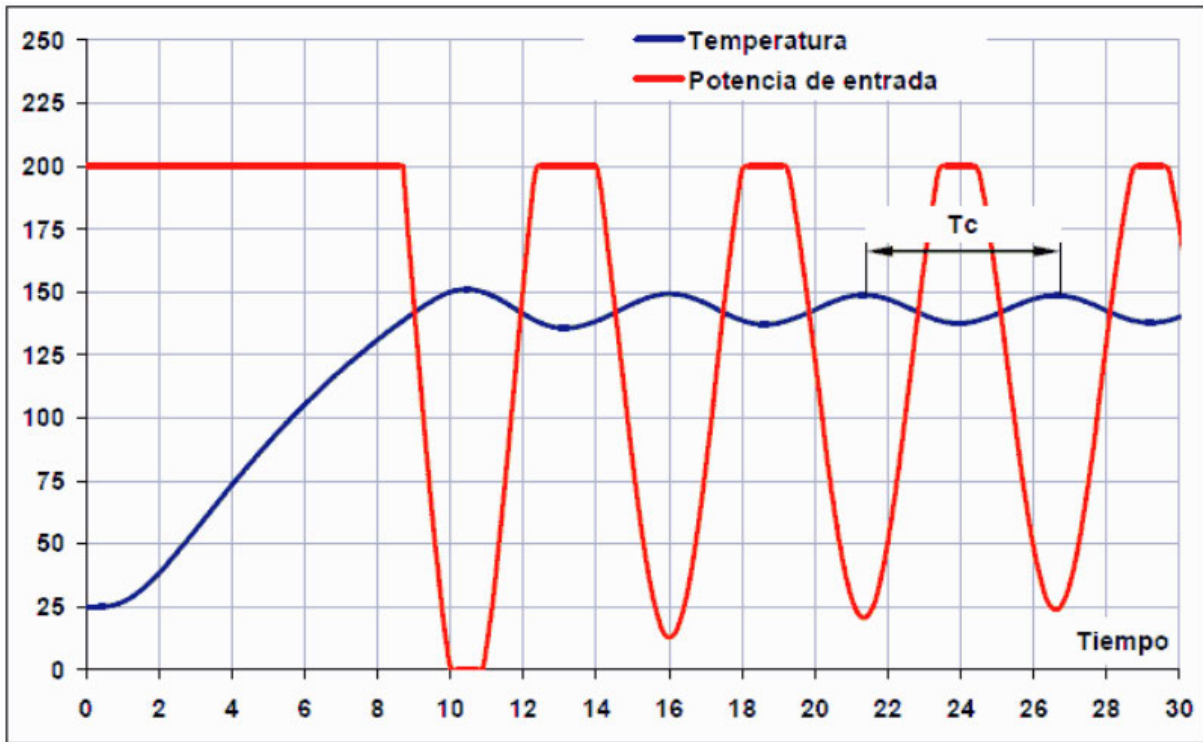


Figura 2.4. Oscilaciones del sistema [11]

Se aplican las fórmulas de Ziegler-Nichols para obtener los valores de las constantes para un controlador PID. [11]

$$Kp = 0,59 * Kc \quad (\text{Ec. 14})$$

$$Ki = 1,18 * \frac{Kc}{Tc} \quad (\text{Ec. 15})$$

$$Kd = 0,075 * Kc * Tc \quad (\text{Ec. 16})$$

Este método es utilizado en [15] para la sintonización del controlador PID en un robot de seguimiento de líneas basado en navegación visual.

### 2.1.2 Windup

El fenómeno conocido como “Windup”, “Integral Windup” o “Reset Windup”, es el efecto que se produce cuando el término integral se carga con un valor muy grande y provoca oscilaciones cuando el error cambia de signo. Esto sucede debido a la no linealidad del término integral, que aumenta positivamente en caso de tener un error positivo para intentar reducirlo, de tal forma que, se va cargando positivamente con el paso del tiempo, y cuando se alcanza el SP y el error cambia de signo se produce una oscilación generada por la carga del término integral, que sigue cargado positivamente y hace crecer el error negativo. El término integral continuará generando más error negativo hasta que la contribución negativa de los nuevos instantes de tiempo anule al acumulado de error positivo, lo cual generará oscilaciones hasta lograr estabilizarse. [16]

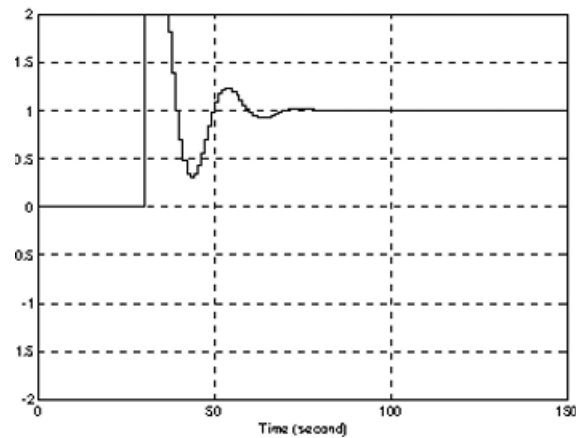


Figura 2.5. Señal de control sin restricciones [1]

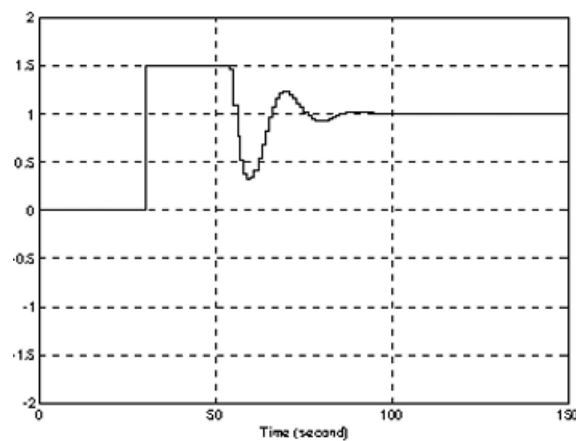


Figura 2.6. Señal de control limitada [16]

Existen diversas estrategias para limitar el efecto “Windup”. Algunas de las más empleadas son los métodos de integración condicionada. [17]

- Detener la integración cuando el error sea grande (cuando el valor absoluto del error supere un umbral definido).
- Detener la integración cuando el control sature.
- Detener la integración cuando el control sature y el error tenga el mismo signo que la señal de control.
- Limitar el integrador (cuando el término integral alcance un umbral definido).
- Detener la integración y asignar un valor predeterminado al término integral cuando se cumpla una condición.

En el estudio de estrategias “Antiwindup” [18], se evalúan distintos métodos de integración condicionada y se concluye que, aunque todos los métodos reducen significativamente el “Overshoot”, el método de integrador limitado es el que produce mayor oscilación.

### 2.1.3 Feedforward

La técnica de “Feedforward” es una técnica de compensación utilizada para anticiparse a errores debidos a perturbaciones del sistema que puedan ser modeladas. Si se conoce la respuesta del sistema ante determinadas perturbaciones se puede adicionar a la salida del controlador un coeficiente de compensación de error. Un ejemplo de esta técnica se aplica en [19] para compensar el error que generará el movimiento de un objeto conociendo su velocidad en un sistema de tiempo discreto.

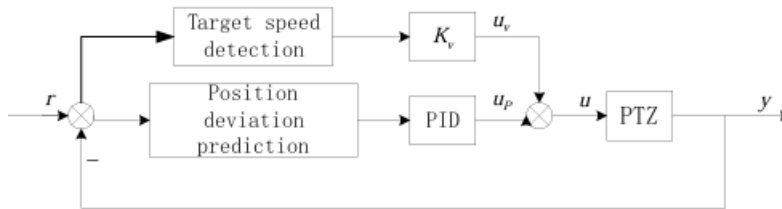


Figura 2.7. Diagrama de compensación en un controlador PID [19]

El coeficiente de “Feedforward” se define como  $K_v$  y las salidas del sistema quedan modeladas como: [19]

$$u(k) = u_p(k) + u_v(k) \quad (\text{Ec. 17})$$

$$u_p(k) = k_p * e(k) + k_i * \sum_{i=0}^k e(i) + k_d * [e(k) - e(k - 1)] \quad (\text{Ec. 18})$$

$$u_v(k) = k_v * \frac{[r(k) - r(k - 1)]}{T} \quad (\text{Ec. 19})$$

$$e(k) = r(k) - (y(k) + u(k - 1) * D) \quad (\text{Ec. 20})$$

### 2.1.4 Ventajas y desventajas

La simplicidad y facilidad de implementación del controlador PID constituye, junto a su poco coste computacional, una de sus principales ventajas. [21]

Se pueden configurar sus parámetros para obtener una respuesta rápida o una respuesta precisa en función del tipo de sistema. [20]

Adicionalmente, la capacidad de adaptación de este tipo de controladores permite su uso en diversas industrias y aplicaciones desde control de temperatura, control de velocidad de motores, robótica o en procesos químicos y científicos. [21]

La mayor desventaja de este tipo de controlador es su dificultad para manejar no linealidades del sistema. Su capacidad para manejar perturbaciones y cambios en la dinámica del sistema es limitada, al igual que retardos entre la salida y la respuesta del proceso. Otra desventaja de este tipo de controlador es la dificultad de sintonización que puede requerir de experiencia para lograr un ajuste fino. [21]

## 2.2 Controlador MPC

El control predictivo por modelo (Model Predictive Control) es un algoritmo de control basado en predecir el comportamiento futuro del sistema en una ventana de tiempo finita conocida como horizonte. A partir del estado predicho y del estado actual del sistema, el controlador actúa sobre las variables de control para alcanzar el estado deseado de la variable de control teniendo en cuenta las limitaciones del sistema. Tras un intervalo de tiempo el proceso se repite con un horizonte desplazado. [23]

La implementación del controlador se basa en iterar sobre el modelo de planta del sistema en el instante actual y resolver la estrategia de control que minimice el costo para un horizonte de tiempo, implementando únicamente el primer paso de la estrategia y repitiendo el proceso tras obtener el nuevo estado del sistema. [22]

La siguiente figura muestra un diagrama del funcionamiento del MPC. [24]

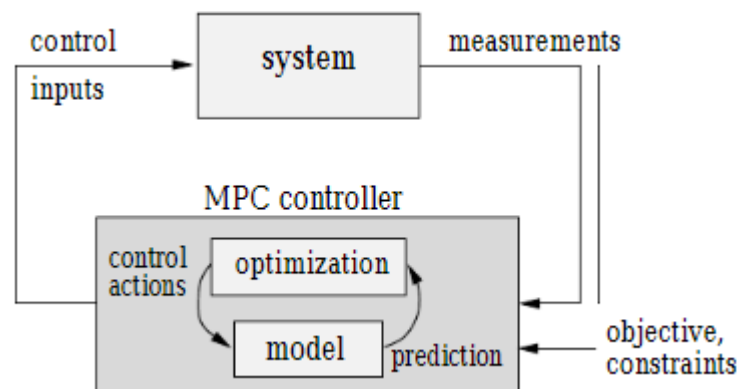


Figura 2.8. Modelo de control predictivo [24]

### 2.2.1 Modelo del proceso

Representación matemática de la salida del sistema en función de su entrada sobre las que se realizaran las predicciones de estados futuros. Es fundamental en el diseño de un controlador MPC, ya que todas las predicciones futuras del comportamiento del sistema dependen del modelo, y si éste tiene errores o no se ajusta con precisión al comportamiento del sistema, se puede llegar a predicciones incorrectas.

Habitualmente la complejidad de encontrar un modelo que se ajuste a la respuesta del sistema es uno de los principales inconvenientes del MPC. En [25], se emplean técnicas de Machine Learning (aplicando modelos de regresión y Random Forest) para construir un modelo dinámico a partir de datos históricos.

El modelo matemático que describe el proceso puede ser lineal o no lineal y en tiempo continuo o discreto. Entre los modelos de tipo lineal encontramos modelos regresivos o modelos que describen el sistema a través de un conjunto de ecuaciones diferenciales. Algunos de los modelos no lineales se basan en el uso de redes neuronales como en [26], donde se emplean redes neuronales para construir un modelo dinámico del sistema en un sistema embebido en tiempo real.

El modelo mejora considerando la respuesta del sistema a estrategias de control previamente aplicadas.

### 2.2.2 Función de costo

Es la función matemática que define el costo de una acción de control sobre el horizonte de predicción. Consiste en asignar un costo mayor a acciones que generen desviaciones en la variable de control. [27]

Generalmente se emplean funciones de costo cuadráticas para penalizar más las desviaciones en las variables de control.

Una función de costo típica puede ser: [23]

$$J = \sum_{k=0}^{N_p} (y_k - y_{ref})^2 + \sum_{k=0}^{N_c} \lambda * u_k^2 \quad (\text{Ec. 21})$$

Donde  $N_p$  representa el horizonte de predicción y  $N_c$  el horizonte de control. Los valores de  $y$  representan la salida predicha y la referencia y  $u_u$  representa la acción de control con un coeficiente de ponderación  $\lambda$ .

El primer término penaliza la desviación de la salida y el segundo término penaliza el esfuerzo de control.

### 2.2.3 Ventajas y desventajas

La principal ventaja de los controladores MPC es su capacidad para manejar procesos inestables y controles multivariable. Además, pueden configurarse para considerar los límites de los actuadores y pueden aumentar las ganancias para permitir una operación cercana a los límites del sistema. [28]

Entre algunas de las limitaciones de este tipo de controladores cabe destacar la dificultad para encontrar un modelo que se adapte adecuadamente a la respuesta real del sistema y la necesidad de ajustar correctamente los horizontes de predicción. Muchos modelos se limitan únicamente a procesos estables en lazo abierto o consideran únicamente perturbaciones en la salida y tienen problemas para manejar perturbaciones de entrada. [28]

## 2.3 Controladores de lógica difusa

El concepto de lógica difusa se asocia a cómo las personas perciben el medio y cómo clasifican propiedades del medio en conjuntos según su percepción. Podría aplicarse esta lógica a medidas de temperatura (si algo está frío o caliente), velocidad a la que se desplaza un objeto (rápido o lento), distancia a un punto (lejos o cerca) ...

De esta forma se caracterizan propiedades físicas del medio en conjuntos subjetivos basados en la percepción, por lo que estos conjuntos son ambiguos según el observador y según el elemento que se esté caracterizando. Los conjuntos difusos intentan modelar la ambigüedad con la que se percibe una variable, por la cual el elemento puede pertenecer a un conjunto en mayor o menor grado. [29]

Aplicando esta lógica en sistemas de control se puede regular el grado de actuación en función del grado de cumplimiento de una premisa. Cada conjunto difuso tiene una función de pertenencia que caracteriza en qué medida un elemento forma parte de un conjunto difuso. Los conjuntos se caracterizan tomando valores extremos que definen los límites del sistema y dentro de los cuales se ubicarán el resto de los conjuntos. [30] Los conjuntos extremos comprenden la actuación para valores fuera de los límites definidos.

### 2.3.1 Funciones de membresía

Una función de membresía es una curva que define el mapeado de cada valor de entrada a uno de los conjuntos definidos o a un grado de membresía. Los conjuntos en lógica difusa se caracterizan entre 0 y 1, ajustándose estos límites según los extremos definidos para el controlador. El espacio comprendido entre estos límites se conoce como universo. [31]

Existen distintos tipos de funciones de membresía. Algunos de los empleados habitualmente son: [32]

- Triangular
- Trapezoidal
- Valores únicos
- Gaussiana
- Lineal por trozos

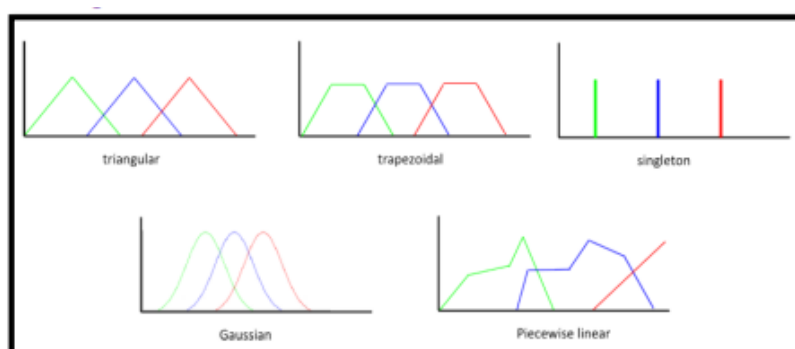


Figura 2.9. Funciones de membresía [32]

Las funciones de membresía se pueden modificar a través de la experimentación para ajustar el modelo difuso al comportamiento del sistema. En [33], se realiza una modificación de las funciones de membresía para obtener un mejor ajuste de sintonización y mejorar el rendimiento del controlador.

### 2.3.2 Fuzzificación y defuzzificación

Los procesos de fuzzificación y defuzzificación son los procesos de aplicar las funciones de membresía a los datos de entrada para generar los valores de membresía en la entrada y transformar la salida difusa en un valor real. [32]

La siguiente figura muestra el diagrama de toma de decisiones de un sistema de control de lógica difusa.

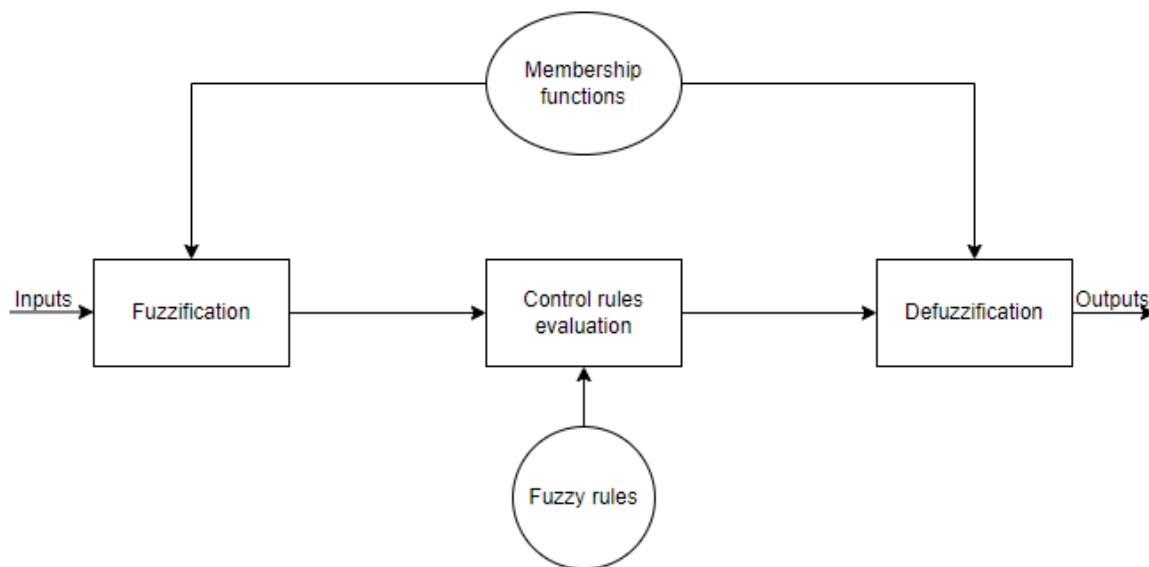


Figura 2.10. Diagrama de toma de decisiones en dominio difuso

### 2.3.3 Reglas difusas

Las reglas difusas constituyen el lenguaje de los controladores difusos. En sistemas de control difusos se definen reglas del tipo SI-ENTONCES para implementar determinadas acciones en función del error en un instante de tiempo y la tasa de cambio del error.

En [34] se diseña un controlador basado en reglas difusas con funciones de membresía triangulares para el control de un péndulo invertido con una rueda de reacción. La implementación de las reglas difusas se define en función del error angular y su derivada con 7 estados difusos (NB, NM, NS, ZE, PS, PM, PB), obteniendo un buen desempeño del sistema a pesar de la existencia de perturbaciones.

La selección del conjunto de reglas difusas y la selección del número de conjuntos difusos para la representación de los estados del sistema requiere de especial atención y se debe hacer cuidadosamente para evitar un efecto contrario al deseado. [34]

#### **2.3.4 Ventajas y desventajas**

La lógica difusa resulta muy potente para resolver problemas complejos, que no pueden ser resueltos por otros sistemas empleando ecuaciones muy complejas, de manera sencilla y con poco coste computacional. Se basa en conjuntos lógicos y razonamientos que pueden ser fácilmente entendidos por un usuario facilitando la configuración de parámetros en cualquier sistema. Una de las mayores ventajas de la lógica difusa es su capacidad de ofrecer resultados precisos con datos imprecisos. [35]

Algunas desventajas que plantea el uso de la lógica difusa en algunos casos son los problemas de implementación cuando no es posible un enfoque sistemático lógico. La configuración de la lógica difusa es muy difícil en algunos sistemas. [35]



### 3. Especificaciones y restricciones de diseño

A continuación, se detallan las especificaciones y las restricciones de diseño para la realización del proyecto.

#### 3.1 Especificaciones

En este apartado se enumeran algunos de los requisitos que debe cumplir la solución desarrollada para garantizar que se alcancen los objetivos marcados.

El objetivo principal del proyecto consiste en obtener un algoritmo de control para el movimiento de una cámara, que permita el correcto video seguimiento de objetivos móviles en diferentes escenarios, proporcionando rapidez de respuesta y estabilidad.

A continuación, se enumeran algunas especificaciones que debe cumplir el algoritmo de control para alcanzar el objetivo.

- **Error máximo:** El controlador debe mantener el objetivo dentro del ángulo de visión de la cámara, esto es, siendo la posición centrada del objeto en la imagen el estado ideal del sistema, mantener el error absoluto por debajo de la mitad del "Field of view" (FoV).

$$|e| \leq 0,5 * FoV \quad (\text{Ec. 22})$$

- **Tiempo de respuesta:** El sistema debe alcanzar la posición deseada en el menor tiempo posible, que variará en función de la distancia al objetivo y el ángulo de visión de la cámara, así como la velocidad del objetivo.
- **Estabilidad:** El seguimiento debe ser estable, manteniendo el objetivo centrado sin oscilaciones continuas.
- **Robustez:** El controlador debe ser robusto ante variaciones en la carga, entendidas como cambios bruscos de maniobra o de velocidad de los objetivos.

#### 3.2 Restricciones de diseño

En cuanto a las restricciones de diseño que deben ser consideradas para la realización del proyecto, encontramos de dos tipos: Restricciones a nivel de hardware y restricciones a nivel de software.

Las restricciones a nivel de hardware vienen dadas por elementos como la cámara sobre la que se vaya a realizar el control y la memoria del ordenador que procese el algoritmo.

Dado que el objetivo de este trabajo se centra en los algoritmos de control, no son tan importantes los recursos hardware empleados, aunque sí que pueden suponer una limitación a la hora de implantar ciertos algoritmos. Se emplean, por tanto, para el desarrollo del trabajo, una cámara PTZ y un ordenador que no degraden significativamente el funcionamiento del algoritmo de control.

La cámara empleada es una AXIS Q6225-LE PTZ [36]:

- Resolución de video: 1920x1080
- Máximo número de FPS: 50/60
- FoV horizontal: 63,8 – 2,2 °
- FoV vertical: 37 - 1,3 °

El ordenador empleado cuenta con una tarjeta gráfica NVIDIA GeForce RTX 3050 con 8GB de memoria RAM y procesa el vídeo de la cámara con un modelo de visión artificial conectado al software de mando y control donde se implementa el algoritmo.

A nivel de software no hay restricciones en cuanto a entorno de desarrollo o lenguaje de programación. Una vez definido el algoritmo puede implementarse en el lenguaje más conveniente. Se debe procurar únicamente que el modelo de visión artificial empleado para proporcionar las detecciones sea lo suficientemente estable.

## **4. Descripción de la solución propuesta**

La solución propuesta es un controlador “Fuzzy PID”, que consiste en emplear un controlador PID con lógica difusa para el manejo de las constantes proporcional, integral y derivativa. Se emplea en el controlador PID la técnica “antiwindup” de condicionamiento integral basada en detener la integración y asignar un valor al integrador cuando se cumpla una determinada condición, en este caso, que el signo del error sea distinto al signo del término integral y este exceda un umbral, que será ajustable.

Las decisiones de diseño a la hora de optar por este tipo de controlador se justifican por la necesidad de adaptación del controlador a distintos tipos de escenarios en cuanto a dinámica de objetivos (variaciones de velocidad y rumbo), ángulos de visión limitados...

Por ello se debe optar por un controlador flexible y robusto capaz de manejar no linealidades del sistema, para lo cual la lógica difusa presenta buenas capacidades. Además, se deben obtener tiempos de respuesta rápidos para lograr el seguimiento adecuado del objeto con ángulos de visión más limitados, y para ello la acción combinada del PID y la lógica difusa aporta mayor capacidad de reacción sin perder estabilidad.

Es importante la inclusión de una técnica “antiwindup” para aportar mayor estabilidad al sistema ante cambios de maniobra tras largos periodos de movimiento en la misma dirección y poder modular el término integral sin generar problemas de oscilaciones.

El sistema en el que se implementa el algoritmo se compone de una serie de elementos físicos y módulos de software que se detallan en siguiente apartado. La arquitectura del sistema se compone de una cámara PTZ manejada por un módulo de control integrado en un software de mando y control que es quien reúne la información recibida de los distintos elementos del sistema, y un módulo de visión artificial integrado en un software que se conecta al flujo de vídeo de la cámara.

El algoritmo se ha implementado en un sistema real junto con otros algoritmos comúnmente utilizados, como el controlador PID, y se han realizado pruebas en distintos escenarios simulados para obtener una comparativa entre las prestaciones de los controladores. Finalmente, se han evaluado las prestaciones del algoritmo propuesto en un sistema en operación.

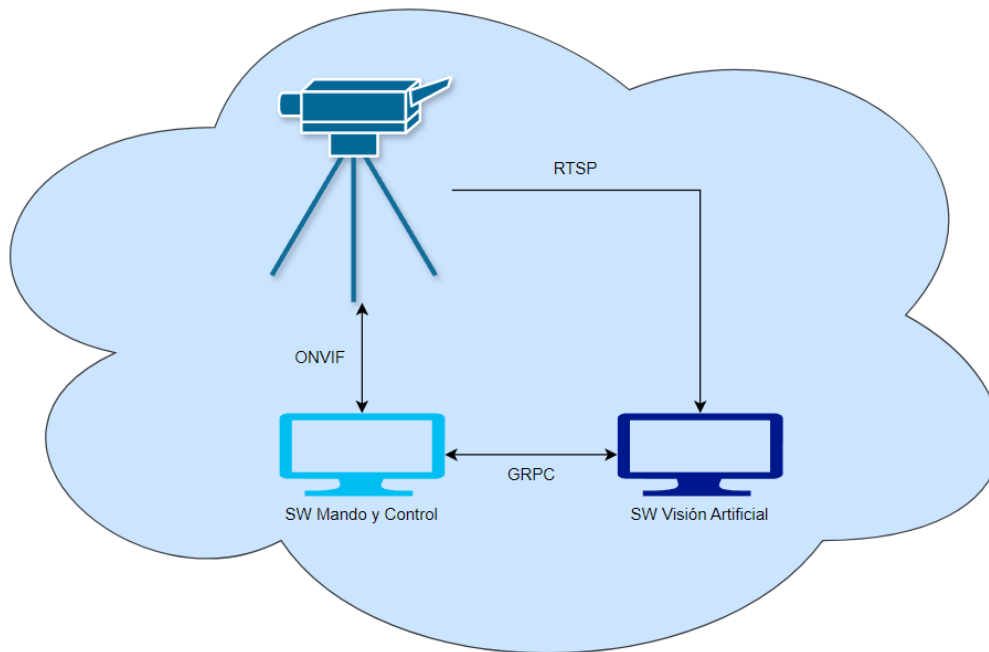


Figura 4.1. Diagrama del sistema

## 4.1 Componentes del sistema

En este apartado se detallan los componentes del sistema y cómo se comunican entre ellos. Estos elementos serán los mismos durante toda la realización del proyecto, con el fin de poder evaluar únicamente el comportamiento del algoritmo de control y comparar sus prestaciones contra otros algoritmos, bajo diferentes condiciones, o con distintos parámetros de configuración, sin interferencia de elementos externos.

### 4.1.1 Cámara PTZ

La cámara empleada es una AXIS Q6225-LE PTZ con las siguientes características: [36]

- Resolución de video: 1920x1080
- Máximo número de FPS: 50/60
- Movimiento horizontal: 360 ° limitado de 0,05 °/s a 150 °/s
- Movimiento vertical: -90 ° a +90 ° limitado de 0,05 °/s a 150 °/s
- Zoom: Óptico de 31x y digital de 12x
- Precisión: 0,10 °
- FoV horizontal: 63,8 – 2,2 °
- FoV vertical: 37 - 1,3 °



**Figura 4.2. Cámara AXISQ6265-LE PTZ [37]**

Es importante considerar las restricciones físicas de la cámara a la hora de limitar el ámbito de acción del algoritmo.

#### **4.1.2 Módulo de visión artificial**

Para que el controlador pueda comandar la cámara para centrar el objetivo en la imagen, debe haber un módulo que le proporcione una detección del objeto dentro de la imagen. Esta detección se obtiene mediante un modelo de visión artificial basado en el algoritmo de detección de objetos en tiempo real YOLO (You Only Look Once).

Se opta por este tipo de algoritmo debido a que es sencillo, aporta velocidad (hasta 45 FPS), precisión en la detección, tiene buena generalización y es de código abierto. [38]

El módulo de visión artificial se integra dentro de un software de visión artificial que se conecta al flujo RTSP (Real Time Streaming Protocol) de la cámara.

#### **4.1.3 Módulo de control**

El módulo de control se integra dentro de un software de mando y control que se conecta a la cámara por protocolo ONVIF (Open Network Video Interface Forum) para el envío de comandos y recepción de información, y al software de visión artificial por el protocolo GRPC (Google Remote Proto Call) para obtener las detecciones.

Dentro del software, un lazo de control se encarga de ejecutar el algoritmo en cada instante de tiempo, proporcionándole el valor de la variable de proceso y el tiempo transcurrido desde la última iteración de control, y obteniendo del algoritmo una velocidad que se comandará a la cámara. Previamente se inicializa el algoritmo asignándole los parámetros de configuración seleccionados, que pueden ser ajustados manualmente en cualquier momento.

El software está desarrollado en C# y, por tanto, el algoritmo se desarrolla en este lenguaje.

## 4.2 Definición del escenario y entorno de pruebas

Para la realización del proyecto se implementa un entorno simulado que represente una dinámica real de un objetivo. El sistema en el que se integra el controlador se emplea habitualmente en el seguimiento de drones de pequeño tamaño o aves, por lo que se define el escenario basándose en este tipo de dinámicas.

El objetivo consiste en simular mediante movimientos a poca velocidad a una distancia cercana, las condiciones de un movimiento a mayor velocidad y distancia.

Todas las pruebas y la caracterización del algoritmo se realizan sobre el eje de Pan, por ser el que ofrece mayor ángulo de visión para preparar el escenario simulado y por ser el que debe manejar dinámicas más rápidas.

A partir de la velocidad máxima de desplazamiento de un objetivo sobre la pantalla se evalúa la viabilidad del entorno simulado. Se define un escenario con desplazamiento a máxima velocidad, suponiendo un movimiento sobre el eje de Pan de 70 km/h, con Zoom máximo de la cámara (2 ° de FoV) y a una distancia de 1 km.

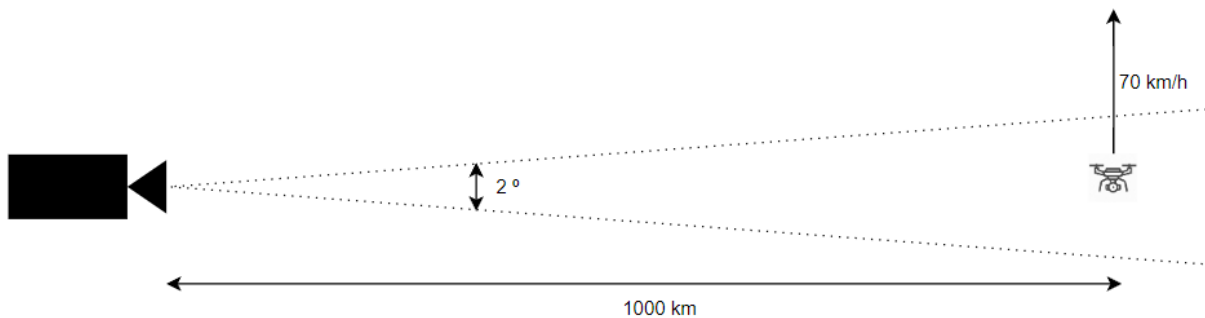


Figura 4.3. Escenario real

Bajo estas condiciones, la distancia del campo de visión en la posición del blanco se calcula de la siguiente manera:

$$\tan 1^\circ = \frac{\frac{d}{2}}{1000 \text{ m}}$$

$$d = 34.92 \text{ m}$$

A una velocidad de 70 km/h (20 m/s), se estima el tiempo que tarda el blanco en recorrer el campo de visión:

$$t = \frac{34.92 \text{ m}}{20 \text{ m/s}}$$

$$t \approx 1.75 \text{ s}$$

Por tanto, en el entorno simulado se deben preparar las condiciones para que un objeto que se desplaza en el eje de Pan, tarde 1.75 s en recorrer el campo de visión de la cámara (si permanece estática).

Las condiciones del entorno de simulación permiten colocar la cámara a una distancia de 3.7 m del objetivo. El objetivo será un objeto que se desplaza por una pantalla y para obtener una imagen nítida se considera necesario un FoV de 4.3 °.

En estas condiciones se evalúa la distancia del campo de visión en la posición del blanco:

$$\tan 2.15^\circ = \frac{\frac{d}{2}}{3.7 \text{ m}}$$

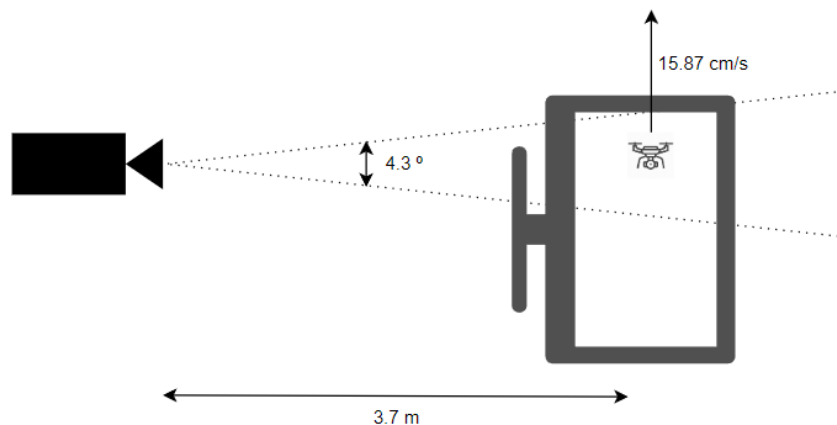
$$d \approx 27.78 \text{ cm}$$

Sabiendo que se deben cumplir las condiciones evaluadas para una distancia mayor, se obtiene la velocidad de movimiento para un objetivo rápido:

$$1.75 \text{ s} = \frac{27.78 \text{ cm}}{v}$$

$$v = 15.87 \text{ cm/s}$$

Se dispone de una televisión de 1.2 m de ancho sobre la que se reproduce el movimiento. Bajo estas condiciones de movimiento máximo, el objeto tardaría aproximadamente 8 s en recorrer la pantalla. Este tiempo se considera suficiente para que el controlador establezca el seguimiento de un objetivo en la misma dirección, y, por tanto, el escenario simulado resulta viable para la evaluación del controlador.



**Figura 4.4. Escenario simulado**

Para las simulaciones se dispone de un software independiente programado en C#. Se trata de un banco de pruebas en el que se implementan conexiones a dispositivos y se definen unas pruebas, con el objetivo de que cada prueba sea replicable y permita evaluar las prestaciones de los distintos algoritmos bajo las mismas condiciones.

## Descripción de la solución propuesta

Se ha implementado un dispositivo servidor dentro del software de mando y control, al que se conectará el banco de pruebas mediante un cliente por el protocolo GRPC. El cliente establecerá unos parámetros iniciales de para la ejecución de la prueba y los enviará al servidor, que será quien comande la cámara.

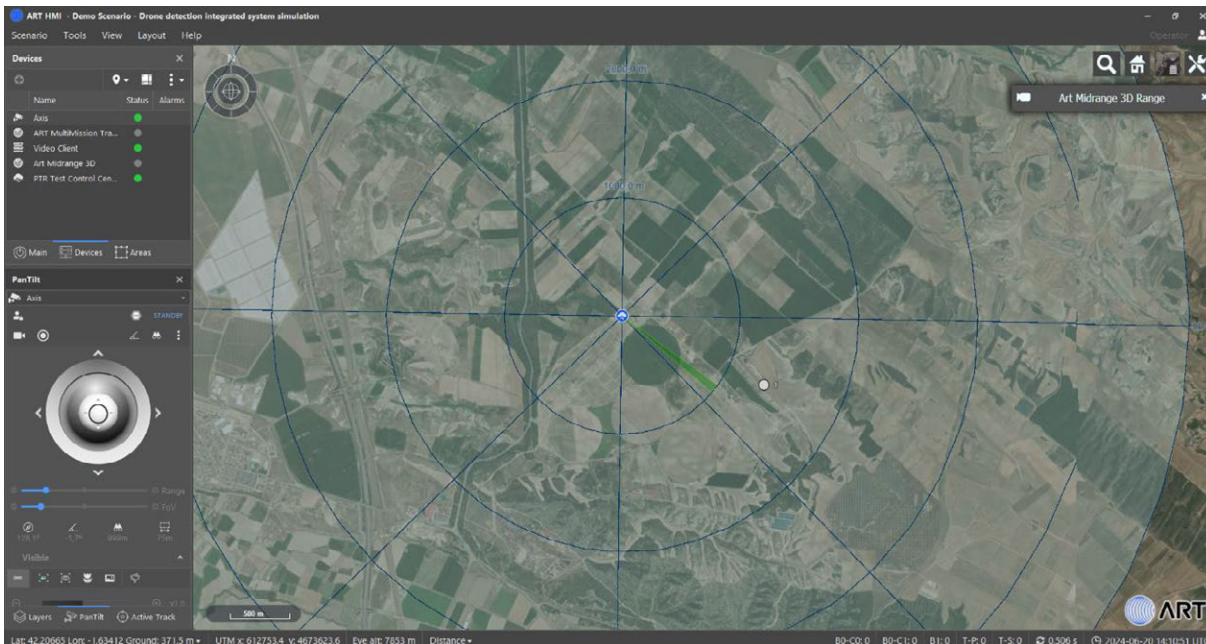


Figura 4.5. Software de mando y control

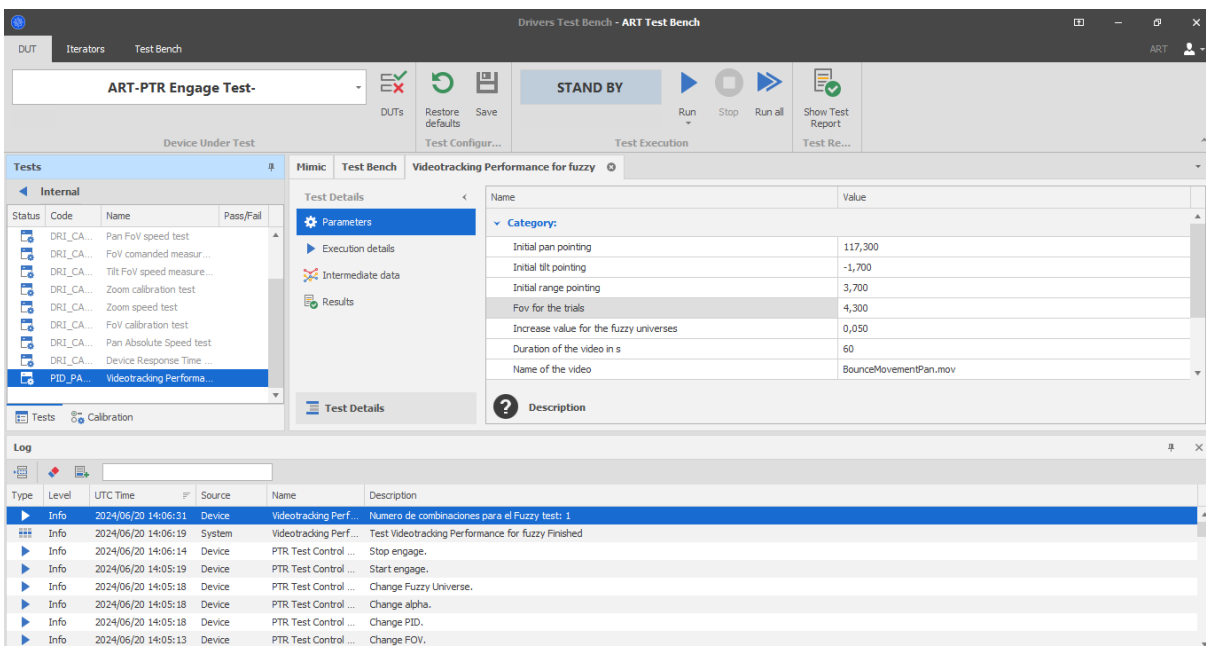
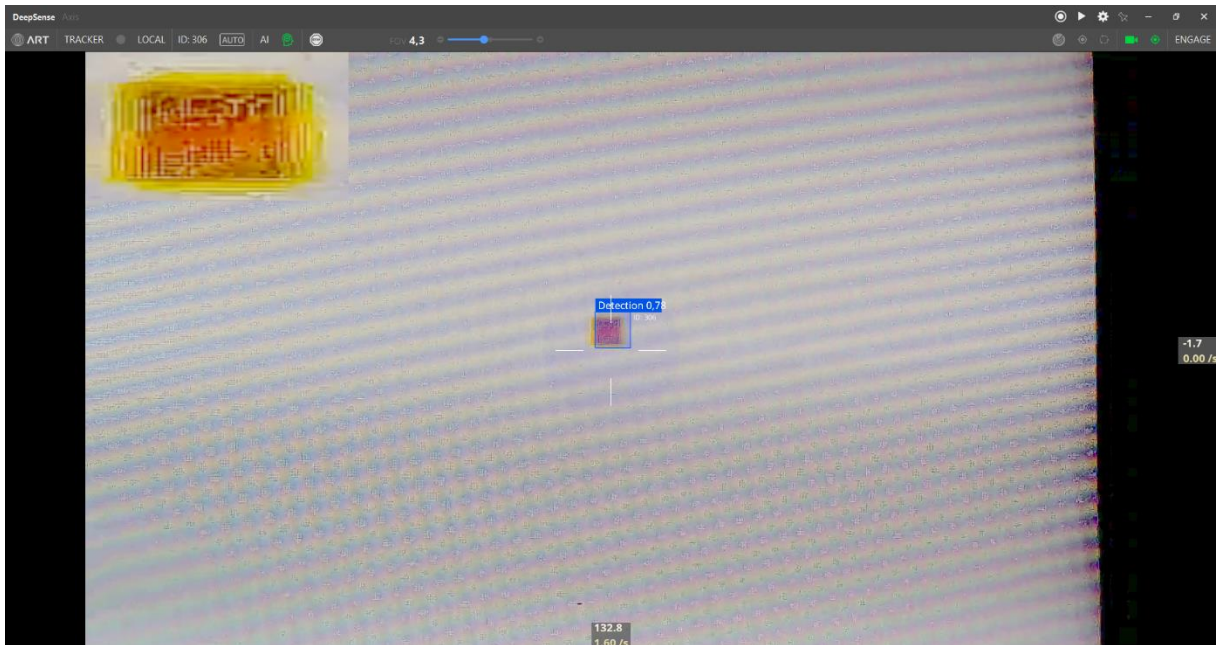


Figura 4.6. Software de pruebas

La simulación que se realiza consiste en posicionar una cámara apuntando a una televisión a cierta distancia y aplicar Zoom a la cámara para limitar su ángulo de visión. Sobre la pantalla se proyecta un vídeo generado con el software de edición de vídeo Da Vinci Resolve, de un pequeño cuadrado amarillo realizando distintos movimientos sobre un fondo blanco.

Se entrena un modelo de visión artificial con YOLO para la detección del cuadrado y se carga en el software de visión artificial.



**Figura 4.7. Software de visión artificial**

En un ordenador se ejecutan el software de mando y control y el software de visión artificial. En otro ordenador se ejecuta el software con la prueba, conectado al dispositivo servidor del software de mando y control del anterior ordenador.



**Figura 4.8. Entorno de simulación**

La prueba que se implementa permite lanzar una ejecución de VLC media player con un vídeo en pantalla completa. Los siguientes parámetros de la prueba son configurables:

- **Apuntamiento inicial:** Se comanda la cámara a una posición de Pan-Tilt predefinida.
- **FoV:** Se establece el field of view inicial de la cámara (ajuste de Zoom).
- **Incremento** (0 por defecto): Se trata de un incremento que puede ser aplicado sobre en la prueba para iterar sobre los universos difusos en caso de que se deseen ajustar estos parámetros, iterando sobre cada parámetro hasta ejecutar todas las combinaciones posibles dentro de un rango de valores (definido por código dentro de límites razonables basados en la experiencia).
- **Duración del vídeo:** Tiempo que dura la prueba hasta detener el video seguimiento.
- **Nombre del video:** Nombre del archivo que contiene el video a ejecutar.

La ejecución de la prueba se muestra en el siguiente diagrama.

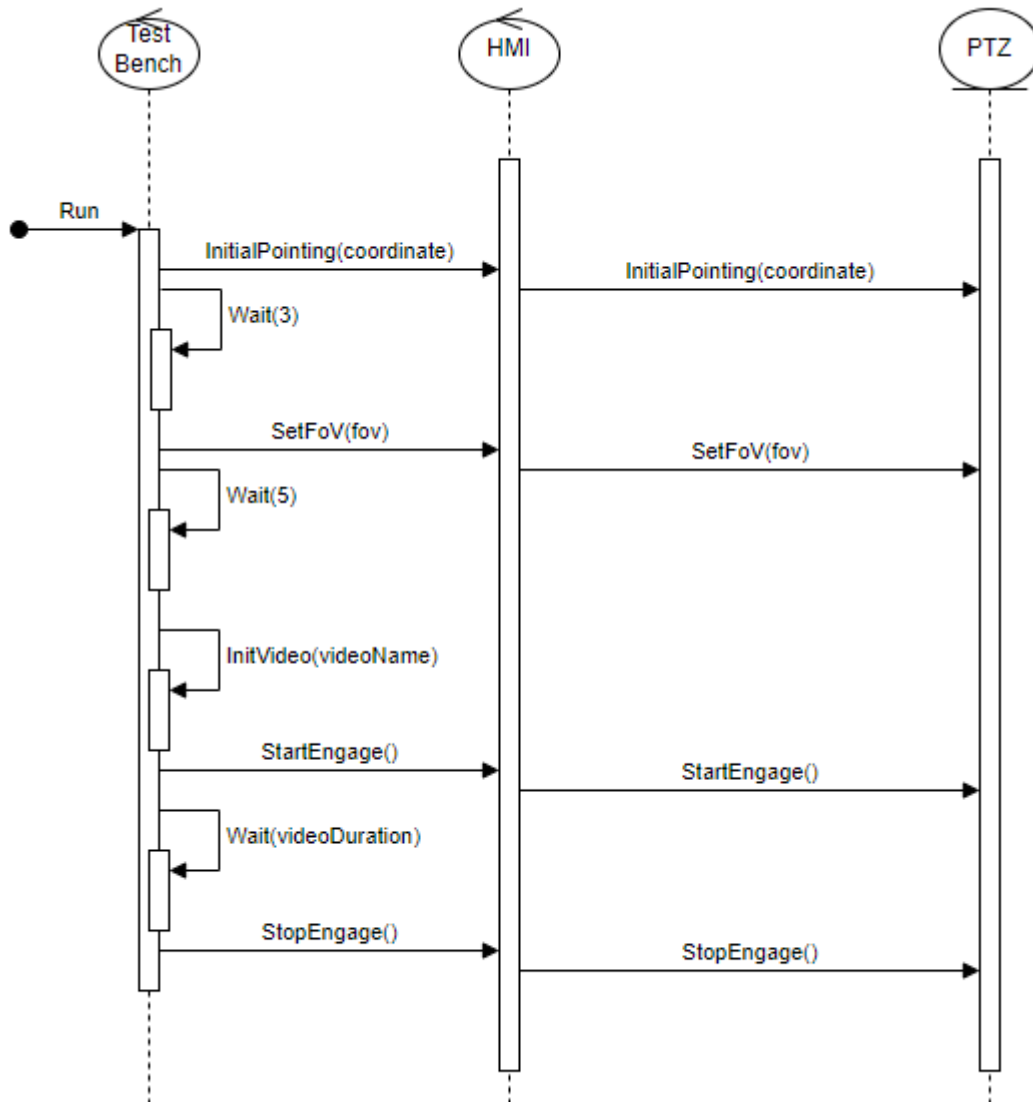


Figura 4.9. Diagrama de secuencia de la prueba

Pese a que se ha dotado al banco de pruebas con la capacidad de iterar sobre los parámetros de los universos difusos y en una prueba adicional, sobre los parámetros del PID (para su sintonización sin lógica difusa), se considera que, pese a que su ejecución es automática, la realización de estas pruebas llevaría gran cantidad de tiempo considerando la duración de un vídeo para sintonizar (por ejemplo, una respuesta al impulso de al menos 10 segundos) y el número de iteraciones, todo ello sumado al factor de que habría que realizar pruebas bajo diferentes condiciones para comprobar si estos parámetros son adaptables a otras dinámicas hacen que se considere impracticable la realización de estas pruebas debido al tiempo destinado para el proyecto. Se consideran este tipo de pruebas para sintonización como líneas de trabajo futuras.

Un punto importante del escenario es la latencia del flujo de vídeo, que supondrá prácticamente el total de la latencia del controlador. Este valor puede ser medido visualizando un reloj con milisegundos en internet y proyectando este valor en la televisión para apuntar la cámara y ver el mismo valor en el flujo de vídeo.

Tras la realización de esta prueba se obtiene el siguiente resultado, con la ventana del navegador a la izquierda y el flujo de vídeo a la derecha.

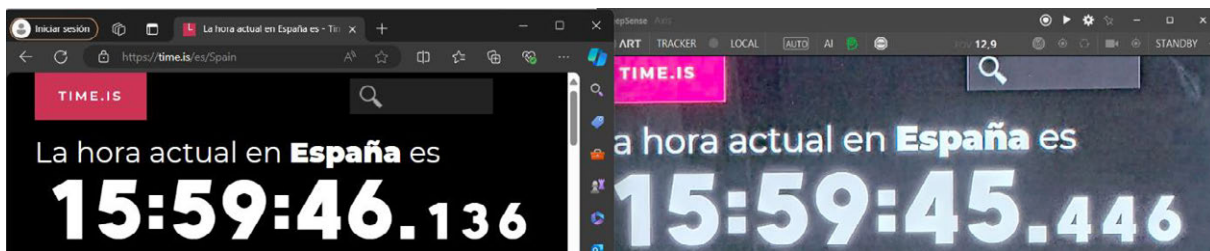


Figura 4.10. Medición de latencia

Con estos datos se estima el valor de latencia.

$$15:59:46.136 - 15:59:45.446 = 690 \text{ ms}$$

Por tanto, se debe tener en cuenta en los escenarios simulados que habrá una latencia mayor a 690 ms.

Tal y como se ha definido anteriormente en el escenario de velocidad máxima, un objetivo tarda 1.75 s en recorrer 1 FoV, que para un objetivo centrado hace que el tiempo de reacción sea de la mitad, es decir, 875 ms. Con el valor de latencia medido, y suponiendo que no hubiera otras latencias significativas, el margen de maniobra del controlador ya sería muy ajustado para responder a este tipo de dinámicas.

Esto se tiene muy en cuenta a la hora de ajustar los parámetros de los controladores, y a la hora de evaluar los resultados para los distintos escenarios.

### 4.3 Desarrollo del algoritmo

En esta sección se detallará el desarrollo del algoritmo de control en tres fases, comenzando por el desarrollo de un controlador PID sobre el que se aplica un control “antiwindup” y finalmente se añade el control de las constantes con lógica difusa.

El apuntamiento inicial de la cámara se realiza a través del software de mando y control con la posición del objetivo proporcionada por un radar alineado con la cámara. Una vez que la cámara se centra en la posición comandada realiza un escaneo en espiral hasta que el modelo de visión artificial detecte el objetivo, cediendo entonces el control de la cámara al algoritmo controlador. Para los casos simulados, el apuntamiento inicial lo realiza el software de la misma manera mediante detecciones simuladas.

Un lazo de control se encarga de gestionar las entradas y salidas del algoritmo de control.

- **Entradas:** Variable de proceso medida en FoV como posición en la imagen relativa al centro y tiempo transcurrido desde la última ejecución del algoritmo.
- **Salidas:** Velocidad a comandar a la cámara.

#### 4.3.1 Desarrollo del controlador PID

En la fase inicial se diseña e implementa un controlador PID para el eje de Pan y para el eje de Tilt. El desarrollo del algoritmo será el mismo en ambos casos, pero a nivel de software se debe instanciar un controlador distinto para cada eje ya que son independientes, y, además, tendrán parámetros distintos para las constantes.

El objetivo del controlador PID es el de minimizar el error entre el SP y la variable de proceso, entendiéndose esta como la posición del centroide de la video detección relativa al centro de la imagen expresada en FoV.

El estado ideal del sistema se encuentra cuando el objetivo queda situado en el centro de la imagen, y, dado que las posiciones dentro de la imagen se definen relativas al centro, esta sería la posición (0,0), quedando establecido el SP en pan y en tilt con valor 0.

$$SP = 0 \tag{Ec. 23}$$

Dado que el procesado de control de la variable de proceso se produce en instantes de tiempo a medida que se reciben video detecciones hay que plantear las ecuaciones de un PID en tiempo discreto. Para cada iteración de control de la variable, se determina el error como la diferencia entre el SP y la variable de proceso.

$$e[k] = SP - PV[k] \tag{Ec. 24}$$

El intervalo de tiempo transcurrido se obtiene en segundos y se comprueba que no sea menor que un valor mínimo ni mayor a un valor máximo para evitar posibles fallos en el algoritmo (como divisiones por 0). Si el valor fuera menor al mínimo se establecería el valor mínimo y si fuera mayor al máximo se establecería el valor máximo. Estos valores se definen como constantes.

$$\begin{aligned} \text{MINIMUM\_TIME} &= 1\text{ms} \\ \text{MAXIMUM\_TIME} &= 2\text{s} \end{aligned}$$

Se calcula la aportación del término proporcional como el producto del coeficiente proporcional con el error en ese instante de tiempo.

$$u_p[k] = K_p * e[k] \quad (\text{Ec. 25})$$

Se calcula la aportación del término integral como el sumatorio en el tiempo del producto del coeficiente integral con el error y el intervalo de tiempo transcurrido desde la última iteración de control.

$$u_i[k] = \sum_{i=1}^k K_i * e[k] * dt[k] \quad (\text{Ec. 26})$$

Es necesario restringir la acción integral definiendo unos límites para evitar saturar el controlador. Se limita la acción integral dentro del rango máximo de velocidad de movimiento de la cámara para cada eje con el objetivo de evitar valores infinitos.

El controlador almacena el valor de la variable de proceso de la última iteración para implementar la acción predictiva. Se obtiene la variación de la variable de proceso desde la última iteración y se calcula el término derivativo como el producto del coeficiente derivativo por la variación de la variable de proceso entre el tiempo transcurrido desde la última iteración.

$$u_d[k] = K_d * \frac{dPV}{dt} \quad (\text{Ec. 27})$$

La salida del controlador se obtiene sumando la acción de todos sus términos.

$$u[k] = u_p[k] + u_i[k] + u_d[k] \quad (\text{Ec. 28})$$

Para adaptar la salida a los límites físicos de la cámara se definen unos parámetros máximos y mínimos configurables y se aplican limitando la salida del controlador al igual que sobre la acción integral.

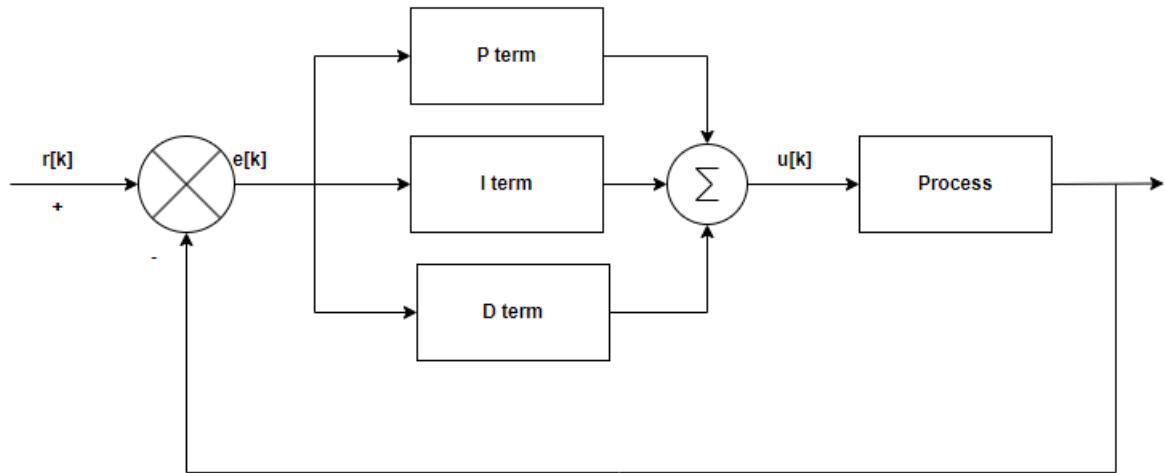


Figura 4.11. Diagrama del controlador PID

Se emplea el método de Ziegler-Nichols en lazo cerrado para sintonizar el PID.

En la realización de este método se anulan las componentes integral y derivativa estableciendo  $K_i = 0$  y  $K_d = 0$ , y se incrementa  $K_p$  hasta que el sistema oscile de forma continua y lineal, sin amortiguamiento ni incremento de la oscilación.

El valor de  $K_p$  que genera este efecto se conoce como ganancia crítica, y se identifica como  $K_c$ .

Para encontrar este valor se ejecuta una simulación como las que se definen en el apartado "5. Resultados" de este proyecto.

Se genera un vídeo de un pequeño cuadrado amarillo que se desplaza a lo largo de una pantalla y se apunta la cámara, situada a 3.7 metros de distancia y con un FoV (Field of View) de  $4.3^\circ$ , sobre una zona de la pantalla. En un momento dado, se comanda, desde el software de mando y control, el inicio del proceso de video seguimiento sobre una detección simulada y ubicada en el ángulo sobre el que se encuentra apuntando la cámara. Una vez que el cuadrado entra en el campo de visión de la cámara, es detectado y se inicia el video seguimiento. Se pausa el vídeo tras un pequeño periodo de seguimiento y la cámara intenta centrar el objetivo en la pantalla.

La simulación se realiza iterando sobre distintos valores de  $K_p$  que se van incrementando hasta que el controlador no sea capaz de corregir el error una vez que el objetivo se ha detenido y se mantenga una oscilación constante sobre el SP.

Tras la realización de las pruebas, el valor que se identifica como ganancia crítica es  $K_c = 3$ .

A continuación, se muestra una gráfica de la respuesta del sistema. En el eje Y se representan los valores de velocidad comandada y error expresados en FoV.

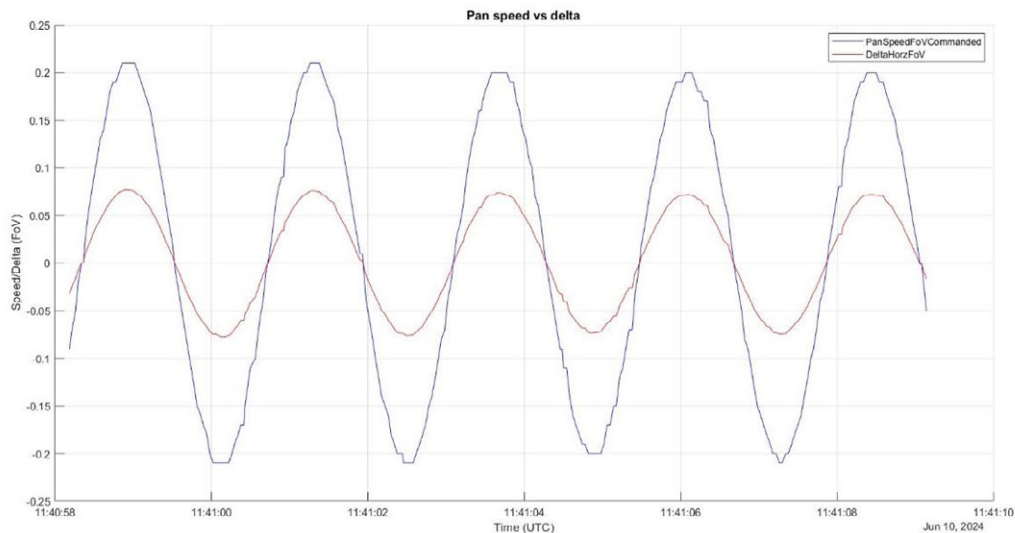


Figura 4.12. Oscilación del sistema

De la gráfica anterior se puede obtener el valor del periodo de oscilación expresado en segundos  $T_c = 2.322$ .

A partir de estos valores se obtienen los valores de las constantes para un controlador PID.

Ganancia proporcional:

$$K_p = 0.59 * K_c \quad (\text{Ec. 29})$$

Ganancia integral:

$$K_i = 1.18 * \frac{K_c}{T_c} \quad (\text{Ec. 30})$$

Ganancia derivativa:

$$K_d = 0.075 * K_c * T_c \quad (\text{Ec. 31})$$

Aplicando estas ecuaciones se obtienen los siguientes valores para las constantes y se configuran como parámetros del controlador.

Kp	Ki	Kd
1.77	1.52	0.522

Tabla 4.1. Parámetros PID

### 4.3.2 Técnica Antiwindup

Sobre el controlador se implementa un “Integral reset” para prevenir la saturación. Esta técnica previene la acumulación excesiva de error en el integrador cuando la acción integral contribuye negativamente al control del sistema, oponiéndose al error.

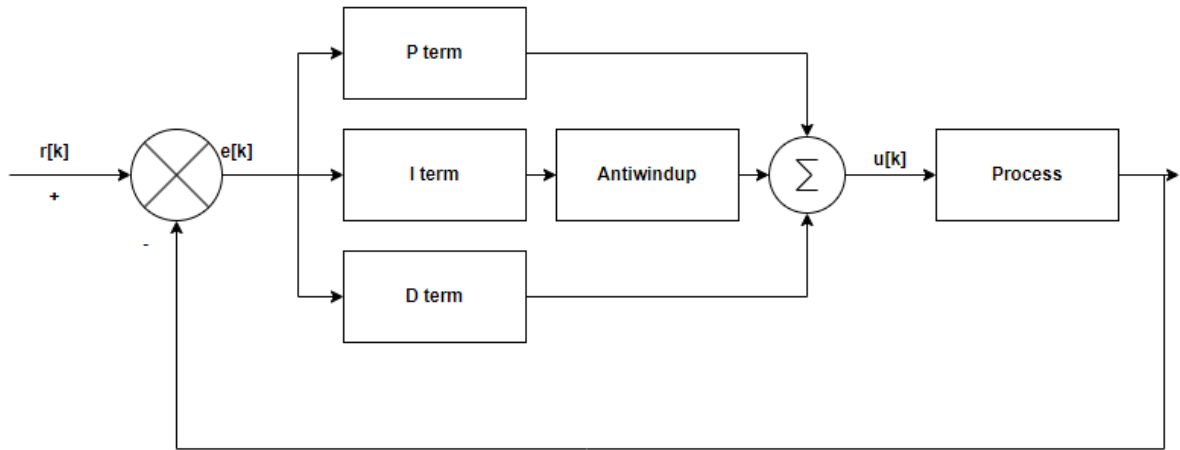


Figura 4.13. Aplicación antiwindup sobre el controlador

Cuando se cumple una determinada condición se reinicia el acumulado de la ganancia integral para mantener la estabilidad del sistema. Se definen dos condiciones que se deben cumplir para establecer a 0 la ganancia integral:

- El signo del término integral es distinto al signo del término proporcional.
- La magnitud del término proporcional es significativa, indicando que se ha producido un cambio en el sentido del error. El valor del umbral es un parámetro configurable, se debe encontrar el equilibrio entre optimizar el tiempo de reacción del sistema ante cambios bruscos de estado y evitar detener la integración durante el seguimiento normal por pequeñas oscilaciones.

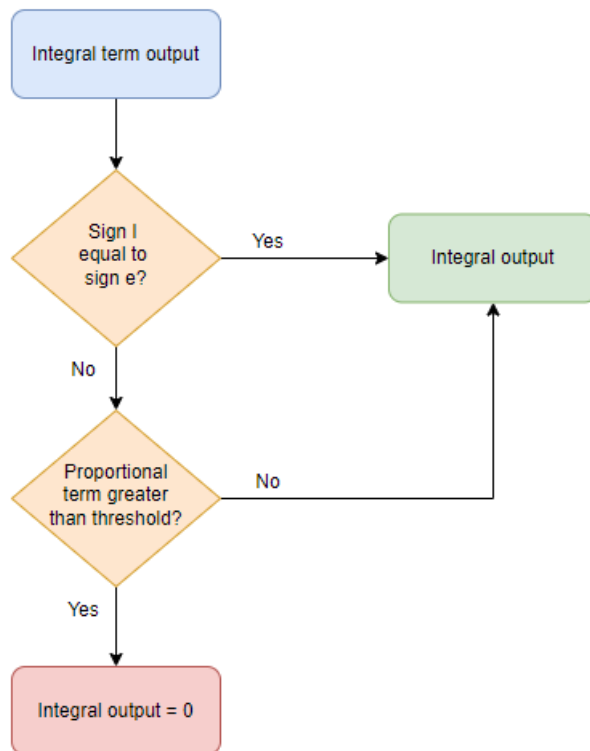


Figura 4.14. Diagrama de flujo de la acción antiwindup

### 4.3.3 Aplicación de lógica difusa

En esta sección se describe la aplicación de la lógica difusa sobre el controlador, con el objetivo de mejorar su adaptabilidad y rendimiento en distintos escenarios. El método de Ziegler-Nichols proporciona un método de sintonización sistemática para los parámetros del PID, con buena carga de atenuación de perturbaciones, pero con un tiempo de respuesta lento y poco robusto ante oscilaciones. La lógica difusa se aplica sobre los parámetros de las constantes del controlador PID proporcionando un ajuste dinámico en función del error y de la tasa de cambio de error.

Sobre los valores de las constantes del PID se aplican correcciones con términos de ganancia en cada instante de tiempo. [40] [41] [42]

$$K'_p = K_p + G_p(e, \Delta e) \quad (\text{Ec. 32})$$

$$K'_i = K_i + G_i(e, \Delta e) \quad (\text{Ec. 33})$$

$$K'_d = K_d + G_d(e, \Delta e) \quad (\text{Ec. 34})$$

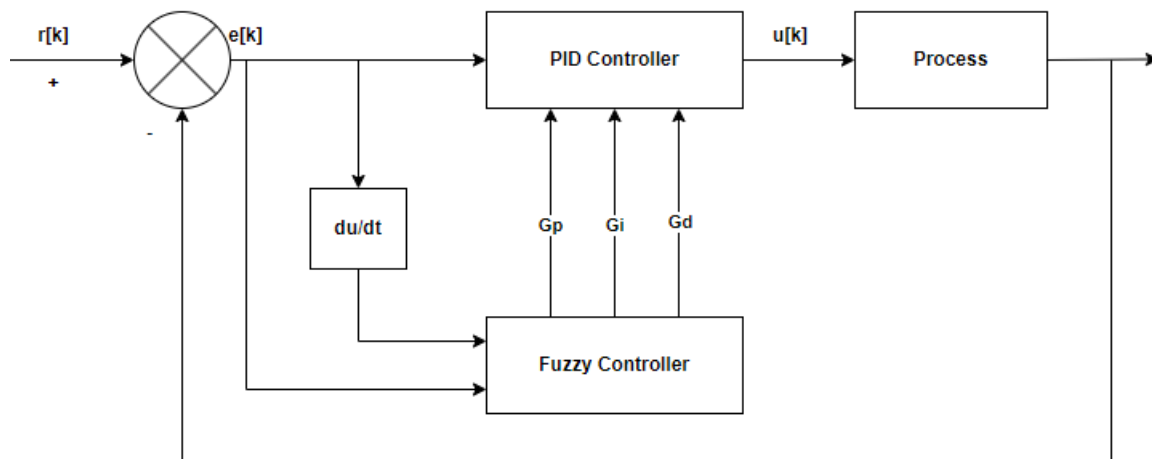


Figura 4.15. Diagrama del controlador Fuzzy PID

La aplicación de la lógica difusa viene de razonamientos intuitivos como por ejemplo [39]:

- Si la magnitud del error es grande, se desea comandar mayor velocidad en ese sentido para minimizar el error rápidamente.
- Si la magnitud del error es pequeña y el error está decreciendo rápidamente, se desea reducir el decremento del error para prevenir oscilaciones.

El concepto principal de la lógica difusa es el de que el estado lógico de una variable no tiene por qué ser verdadero o falso, puede pertenecer a varios conjuntos con distintos grados de pertenencia. Los conjuntos difusos se definen por medio de funciones de pertenencia, ajustados en un rango de valores definido por un universo de discurso. El universo es el rango de valores dentro del cual se definen los conjuntos difusos.

A continuación, se presentan conceptos que hay que definir en la caracterización de controlador difuso.

Los conjuntos difusos son los estados conceptuales que se pueden asignar a un valor de una variable real. Se definen dentro de un rango de valores limitado por el universo y se asignan grados de pertenencia entre [0, 1] de una variable a cada conjunto por medio de una función de pertenencia.

Para aplicar esta lógica al controlador se definen 7 conjuntos difusos. [41]

NB (Negative Big)	NM (Negative Medium)	NS (Negative Small)	ZO (Zero)	PS (Positive Small)	PM (Positive Medium)	PB (Positive Big)
-------------------------	----------------------------	---------------------------	--------------	---------------------------	----------------------------	-------------------------

**Tabla 4.2. Conjuntos difusos**

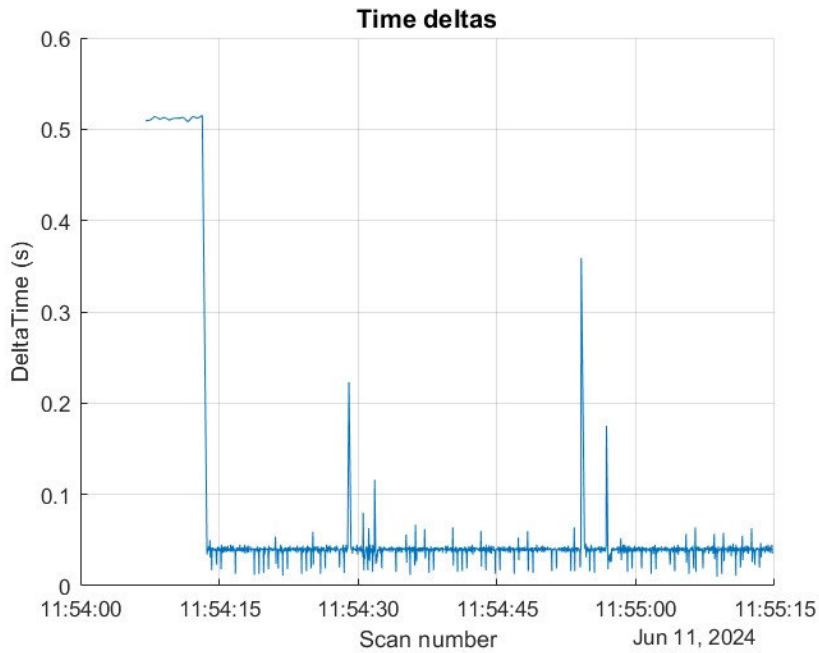
Por ejemplo, con un universo del error entre [-0.5, 0.5] (expresado en FoV), se representan todos los estados posibles del error, ya que el máximo error positivo que se puede tener es un objetivo que se encuentre en el límite de la imagen, es decir, la mitad del FoV. Este objetivo tendría el grado de pertenencia máximo (1) para el conjunto PB (positivo grande), mientras que un objetivo que se encuentre centrado tendrá el máximo grado de pertenencia para el conjunto ZO. El grado de pertenencia a cada conjunto dependerá de la función de pertenencia que se emplee.

Cada variable que controlar tendrá estos conjuntos construidos en cada una por un valor configurable de universo.

Se establecen los universos para la fuzzificación de las entradas, expresados en FoV, ya que los universos son proporcionales al campo de visión en cada instante. Dado que el SP se encuentra en el centro de la imagen y los errores a ambos lados tendrán las mismas magnitudes, pero con distinto signo, a la hora de introducir los parámetros de los universos en el software se introduce únicamente un valor, y se define el universo entre  $[-valor, valor]$ .

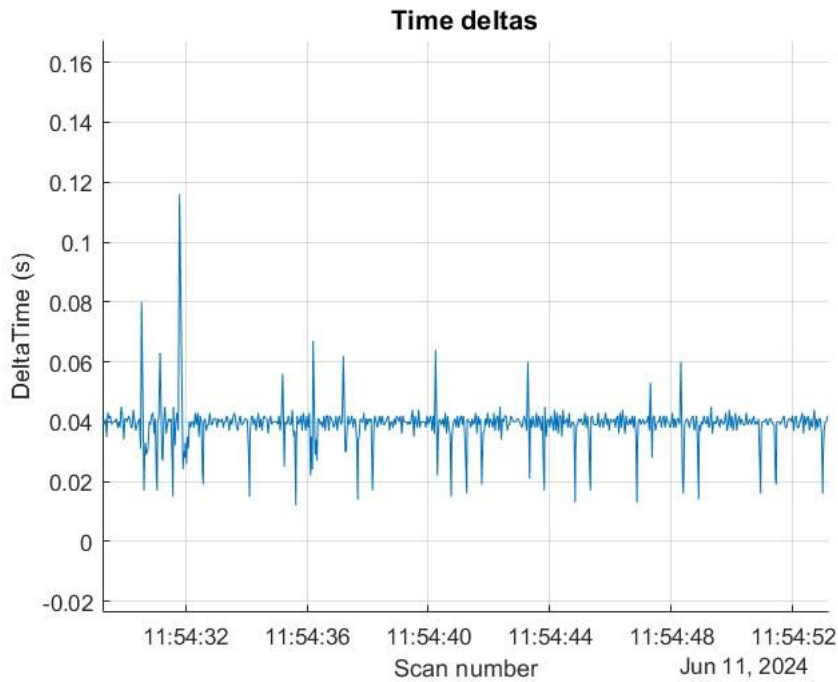
El universo del error queda definido por el máximo error que puede haber en un instante de tiempo. Es decir, la posición más alejada del centro de la pantalla es uno de los lados, a una distancia de 0.5 FoV. Este valor se ajusta a los límites difusos y se define entre [-0.2, 0.2] para aportar mayor capacidad de reacción al controlador ante desviaciones importantes, ya que se considera que 0.2 FoV ya puede ser un error considerable si hay latencia en el sistema.

Para establecer el universo de la desviación del error sobre el tiempo es necesario establecer un valor típico de diferencia temporal entre iteraciones del controlador. En la siguiente figura se muestra una gráfica de la evolución de la diferencia temporal a lo largo de las iteraciones de control en un proceso de seguimiento.



**Figura 4.16. Diferencia temporal entre acciones de control**

Enfocando más sobre la gráfica se aprecia que un valor nominal entre iteraciones de control es de 0.04 segundos.



**Figura 4.17. Diferencia temporal entre acciones de control**

Se caracteriza el universo de la diferencia de error en el tiempo con respecto al tiempo entre iteraciones de control (definido en 0.04 s), y al desplazamiento máximo de un objetivo en la imagen, relacionado con el tiempo entre medidas.

Basándose en las condiciones de movimiento establecidas en la definición del escenario, se calcula el valor de desplazamiento máximo a partir de la velocidad máxima. La velocidad máxima se define en FoV como:

$$v_{\max} = \frac{1 \text{ FoV}}{1.75 \text{ s}}$$

$$v_{\max} \approx 0.5714 \text{ FoV/s}$$

Y por tanto el desplazamiento máximo en 0.04 s será:

$$\text{disp}_{0.04 \text{ s}} = 0.5714 \frac{\text{FoV}}{\text{s}} * 0.04 \text{ s}$$

$$\text{disp}_{0.04 \text{ s}} \approx 0.02286 \text{ FoV}$$

El parámetro es muy dependiente del tiempo, y, por tanto, del entorno del controlador. La cámara tiene un parámetro configurable de tiempo mínimo entre comandos de movimiento, que se establece para las pruebas en 10 ms. Esto sumado a latencias provocadas por el envío de la detección y latencias en el flujo de vídeo, hacen que sea recomendable comprobar experimentalmente los valores de este parámetro.

Se realizan pruebas con un objetivo a máxima velocidad. En la gráfica siguiente se muestra la diferencia de error sobre el tiempo transcurrido desde la última acción de control en cada iteración del controlador.

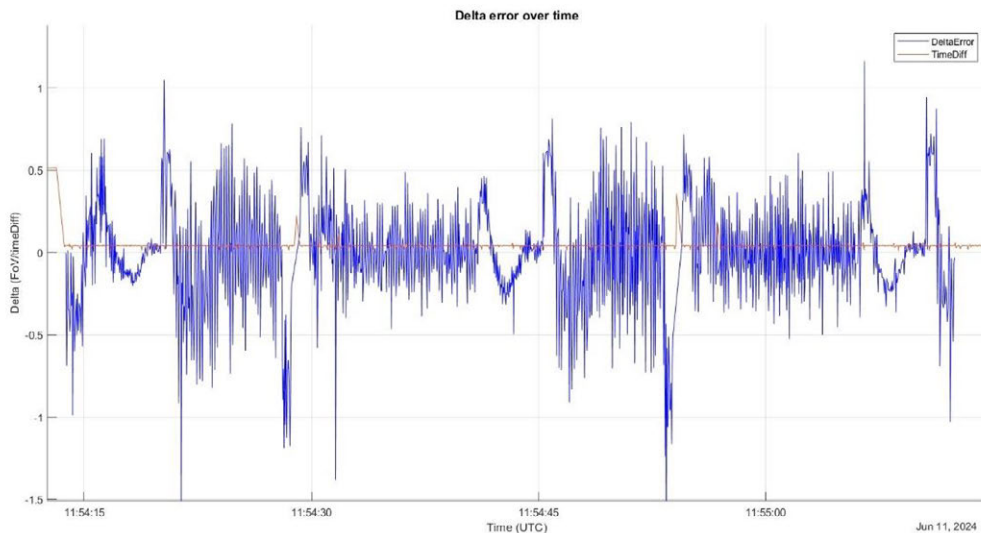


Figura 4.18. Tasa de cambio del error a lo largo del tiempo

*Descripción de la solución propuesta*

De los resultados prácticos se deduce que un buen valor para el universo del delta del error puede ser 0.4 FoV.

Los valores de los universos que se emplearan en el control quedan por tanto definidos según la siguiente tabla.

Universo del error	[-0.2, 0.2]
Universo del delta del error	[-0.4, 0.4]

**Tabla 4.3. Parámetros difusos de entrada**

Los universos de las ganancias para los parámetros de PID se establecen con un umbral en torno a cada una de las constantes. Se emplea un umbral del 35% basado en la experiencia para cada uno de los valores de ganancia.

Universo proporcional:

$$U_p = K_p * 0.35 \tag{Ec. 35}$$

Universo integral:

$$U_i = K_i * 0.35 \tag{Ec. 36}$$

Universo derivativo:

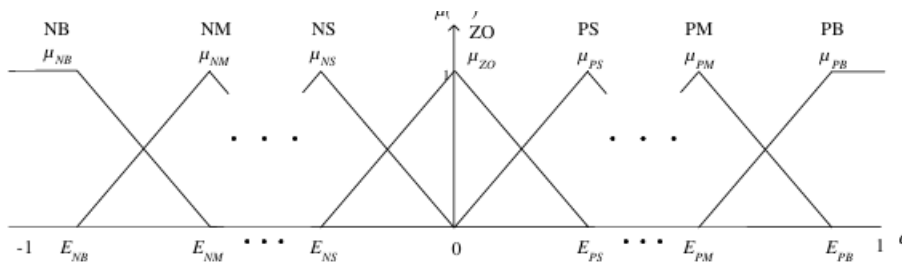
$$U_d = K_d * 0.35 \tag{Ec. 37}$$

Se definen de esta manera los siguientes valores para los universos que se aplicaran en el proceso de defuzzificación para las salidas:

Universo proporcional	[-0.62, 0.62]
Universo integral	[-0.532, 0.532]
Universo derivativo	[-0.183, 0.183]

**Tabla 4.4. Parámetros difusos de salida**

La función de pertenencia utilizada es una función triangular, en donde cada conjunto ocupa el mismo espacio dentro del universo.



**Figura 4.19. Funciones de pertenencia triangular [39]**

Para optimizar el funcionamiento del algoritmo se puede modificar la función de pertenencia modificando la asignación de grados de pertenencia a los conjuntos como se propone en [39].

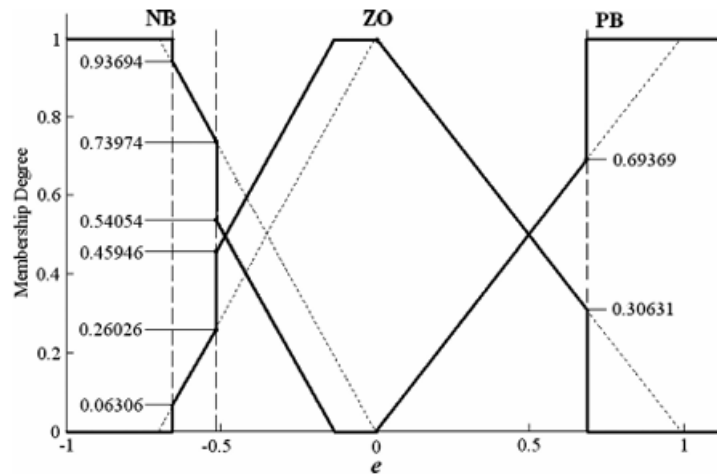


Figura 4.20. Función de pertenencia triangular modificada [39]

Para el diseño del controlador se modifican las funciones de pertenencia, reduciendo a la mitad los picos de NM, NS, PS y PM. La modificación propuesta se realiza con el objetivo de incrementar el espacio de los conjuntos NB y PB, suavizando la respuesta cuando el error se encuentra más próximo a ZO (cero) que con una función triangular sin modificar.

Esta modificación se basa en la experiencia, ya que la evaluación de este tipo de modificaciones para obtener un ajuste fino requeriría de gran cantidad de pruebas con un tiempo ampliamente superior a la duración del proyecto. Sin embargo, se define como una línea de investigación futura que podría mejorar significativamente el rendimiento del controlador.

En el proceso de fuzzificación se aplica la función de pertenencia para asignar grados de pertenencia a los conjuntos difusos para los valores de entrada del sistema.

- Error:  $e[k]$
- Variación del error:  $\Delta e[k]$

Los estados resueltos de ambas variables se utilizan para determinar el estado difuso para la corrección de cada término del PID.

En [41], se hacen algunas consideraciones para la determinar esos estados:

- Cuando  $e$  es un valor muy grande,  $K_p$  debe ser también muy grande. Por otra parte,  $K_d$  debe ser muy pequeño para no generar oposición y  $K_i$  debe ponerse a 0 para evitar oscilaciones.
- Cuando  $e$  y  $\Delta e$  sean valores medios,  $K_i$  debería ser pequeño, mientras que  $K_p$  y  $K_d$  deberían tener valores moderados para una respuesta rápida.
- Cuando  $e$  sea un valor pequeño,  $K_p$  y  $K_d$  deberían aumentar para una respuesta rápida, mientras que  $K_i$  debería tener valores moderados para evitar oscilar entorno a la posición de SP.

*Descripción de la solución propuesta*

Basándose en estas consideraciones se construyen las siguientes tablas de estado para las variables.

$e \setminus \Delta e$	NB	NM	NS	ZO	PS	PM	PB
NB	PB	PB	PM	PM	PS	ZO	ZO
NM	PB	PB	PM	PM	PS	ZO	ZO
NS	PM	PM	ZO	NS	NM	NB	NB
ZO	PB	PB	PM	ZO	NM	NB	NB
PS	PM	PM	ZO	NS	NM	NB	NB
PM	ZO	ZO	NS	NM	NM	NB	NB
PB	ZO	ZO	NS	NM	NM	NB	NB

Tabla 4.5. Regla difusa para Kp [41]

$e \setminus \Delta e$	NB	NM	NS	ZO	PS	PM	PB
NB	NB	NB	NM	NM	NS	ZO	ZO
NM	NB	NB	NM	NS	NS	ZO	ZO
NS	NB	NM	NS	NS	ZO	PS	PS
ZO	NM	NM	NS	ZO	PS	PM	PM
PS	NM	NS	ZO	PS	PS	PM	PB
PM	ZO	ZO	PS	PS	PM	PB	PB
PB	ZO	ZO	PS	PM	PM	PB	PB

Tabla 4.6. Regla difusa para Ki [41]

$e \setminus \Delta e$	NB	NM	NS	ZO	PS	PM	PB
NB	PS	NS	NB	NB	NB	NM	PS
NM	PS	NS	NB	NM	NM	NS	ZO
NS	ZO	NS	NM	NM	NS	NS	ZO
ZO	ZO	NS	NS	NS	NS	NS	ZO
PS	ZO	ZO	ZO	ZO	ZO	ZO	ZO
PM	PB	PS	PS	PS	PS	PS	PB
PB	PB	PM	PM	PM	PS	PS	PB

Tabla 4.7. Regla difusa para Kd [41]

A partir de los estados del error y la tasa de cambio del error, se resuelven los estados de salida para las ganancias de los coeficientes. Se aplica la desfuzzificación a cada uno y se obtienen los valores reales de ganancias proporcional, integral y derivativa que modificarán los parámetros del PID.

## 5. Resultados

En este capítulo se describen las pruebas realizadas y se presentan los resultados obtenidos para la evaluación del algoritmo propuesto. Se distinguen dos bloques de pruebas:

- **Simulaciones:** Se elaboran una serie de escenarios con dinámicas simuladas para evaluar el rendimiento del algoritmo propuesto, el controlador Fuzzy PID, con respecto al algoritmo empleado actualmente en la solución, un controlador PID. Para ambos controladores se activa la acción antiwindup diseñada. Todas las simulaciones se llevan a cabo con la prueba desarrollada y ejecutando distintos vídeos preparados para simular varias dinámicas.
- **Pruebas en campo:** Se evalúan las prestaciones del algoritmo propuesto en un escenario real.

Las prestaciones se evalúan en cualquier caso para el eje de Pan por ser el que presenta mayor complejidad de movimiento y requiere de una mayor optimización para la aplicación estudiada.

Para todos los escenarios se emplean valores de PID fijos obtenidos en el proceso de sintonización y valores para los universos difusos calculados en el diseño del controlador. En la siguiente tabla se establecen estos valores que se emplearan en las simulaciones.

Parámetro	Valor
Constante proporcional	1.77
Constante integral	1.52
Constante derivativa	0.522
Universo del error	0.2
Universo de la tasa de cambio del error	0.4
Universo proporcional	0.62
Universo integral	0.532
Universo derivativo	0.183

Tabla 5.1. Parámetros de control

### 5.1 Simulación A: Respuesta al impulso

En este escenario se evalúa la respuesta al impulso del controlador Fuzzy PID propuesto en comparación con la respuesta al impulso del controlador PID.

Para realizar esta simulación, se ejecuta la prueba con un vídeo con la detección estática y se apunta la cámara, con un FoV de 4.3 °, de tal forma que la detección quede desplazada del centro de la imagen aproximadamente 0.5 FoV, es decir, quedando la detección en el límite derecho de la imagen obtenida por la cámara. Estos parámetros quedan definidos igual para la prueba con el controlador PID y la prueba con el controlador Fuzzy PID.

Se ejecuta la prueba para el controlador PID y se recogen los resultados exportados del proceso de video seguimiento.

Finalmente, se activa el controlador Fuzzy PID y se repite la prueba.

Se ejecutan los scripts de análisis de resultados en modo “step” para analizar el rendimiento de ambos controladores ante una respuesta al impulso.

En esta simulación se evalúan los siguientes parámetros.

- **Rise time:** En este caso se sincronizan las respuestas al mismo instante de tiempo y se entiende como rise time el tiempo que transcurre desde que el error pasa del 100% del valor del escalón al 10%.
- **Overshoot:** Porcentaje en que la variable de proceso excede el valor final del SP.
- **Settling time:** Tiempo que transcurre desde el inicio de la respuesta hasta que la variable de proceso se estabiliza dentro de un umbral del 10% del escalón ( $\approx 0.05$  FoV) en torno al SP.

A continuación, se muestran representadas las respuestas del controlador PID, el controlador Fuzzy PID, y la respuesta de ambos controladores comparada una respuesta ideal.

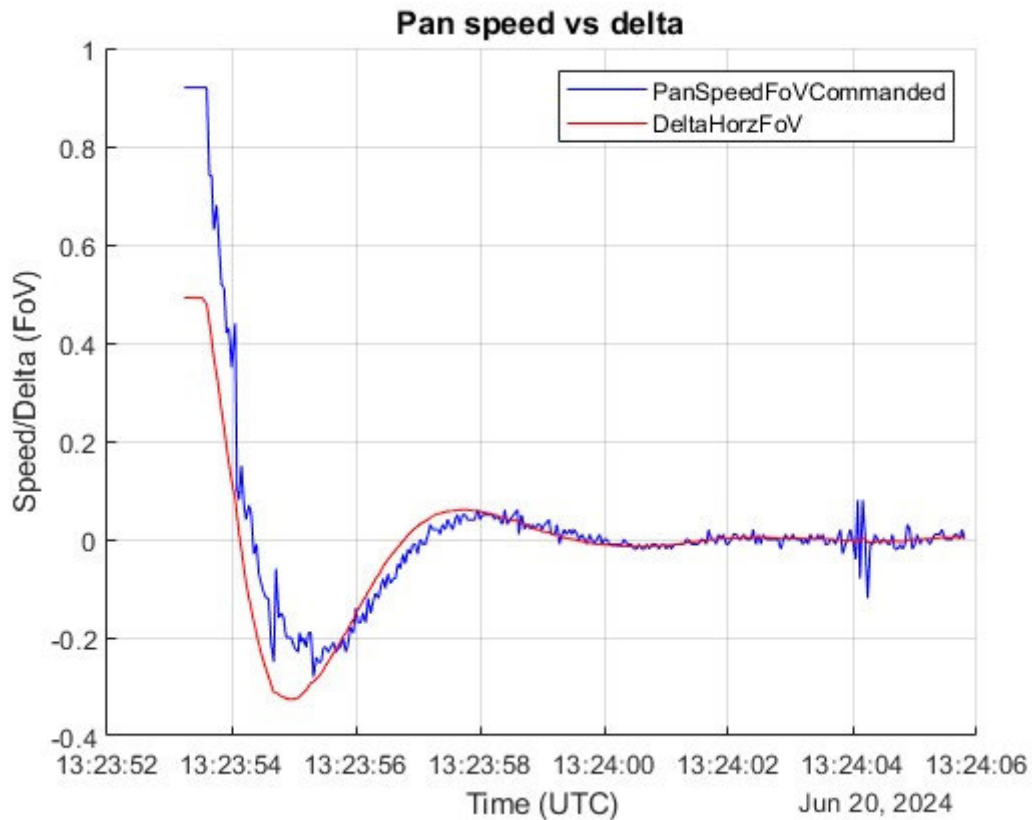


Figura 5.1. Respuesta del controlador PID

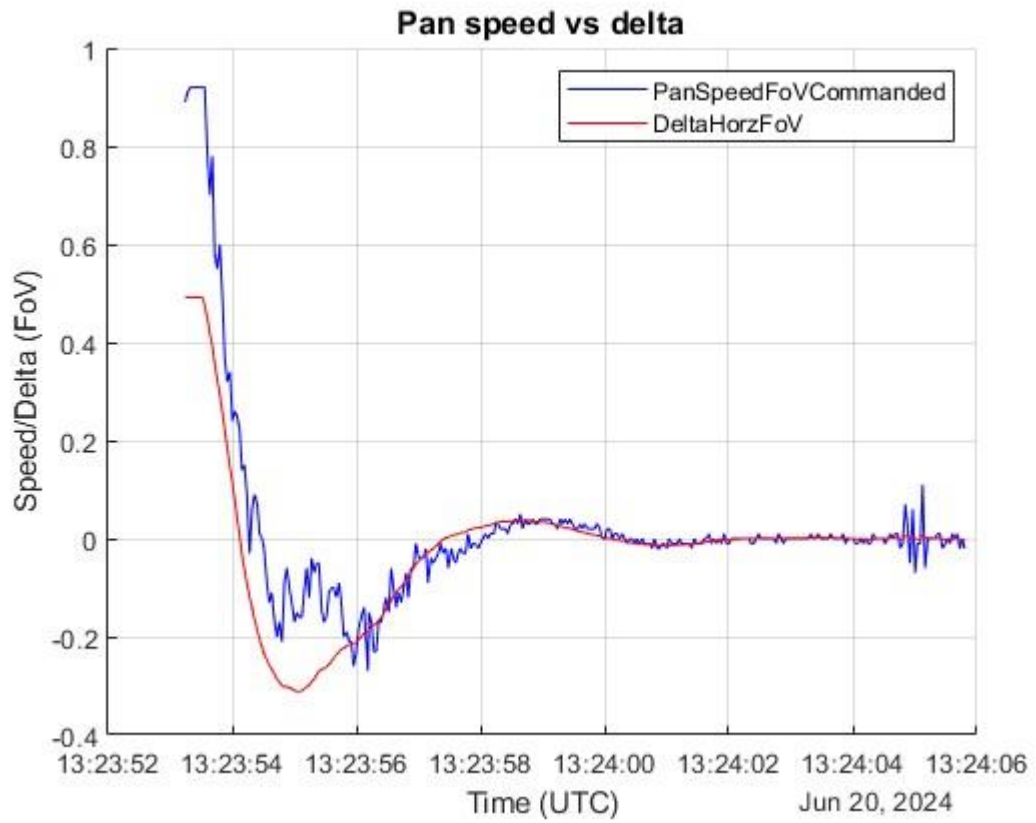


Figura 5.2. Respuesta del controlador Fuzzy PID

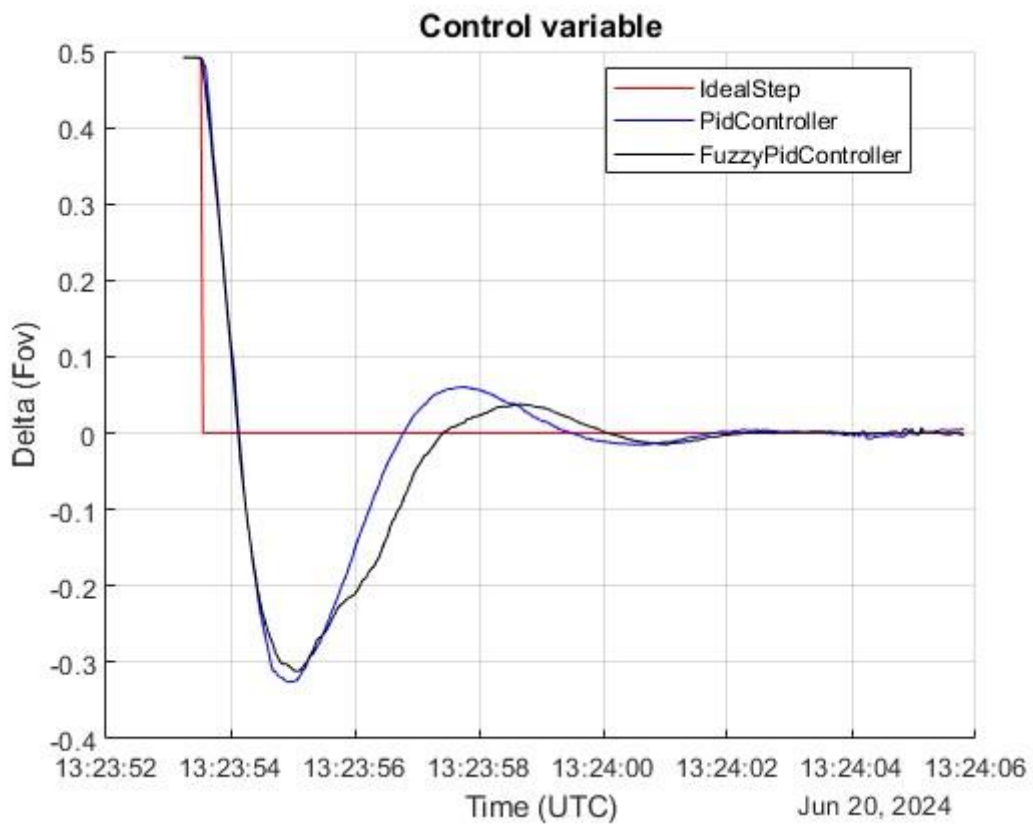


Figura 5.3. Comparativa de respuestas

Se obtienen los siguientes resultados para las respuestas de los controladores.

Controller	Rise time (s)	Percentage overshoot	Settling time (s)
PID	0.568	-0.662	4.720
Fuzzy PID	0.550	-0.634	3.439

Tabla 5.2. Resultados simulación A

Estudiando los resultados obtenidos para la respuesta al impulso, se aprecia una ligera mejora en el tiempo de respuesta y el porcentaje de oscilación para el controlador propuesto con respecto al controlador PID. Se observa también una estabilización más suave para el controlador propuesto, con oscilaciones de asentamiento menores que permiten alcanzar el punto de estabilidad (definido por un umbral del 10% del escalón) en menor tiempo.

Se puede concluir, por tanto, para esta simulación, que el controlador Fuzzy PID mejora ligeramente los tiempos de reacción, y realiza una estabilización más suave de la variable de control cuando el error se reduce, frente al controlador PID.

## 5.2 Simulación B: Desplazamiento y maniobras

En este escenario se evalúan las prestaciones del algoritmo propuesto ante cambios de maniobra bruscos del objetivo, y se compara su respuesta contra un controlador PID. El objetivo realiza maniobras bruscas y se desplaza a distintas velocidades cuando va en un sentido o en otro.

Para realizar esta simulación, se ejecuta la prueba partiendo de una posición en la que el objetivo se encuentra centrado en la imagen y se desplaza a lo largo de toda la pantalla de forma continua. El objetivo parte de la izquierda y se desplaza hasta la derecha en un tiempo total de 15 s, donde realiza un cambio de sentido de 180° y regresa en un tiempo de 10 s. Se establece un FoV para la cámara de 4.3 ° y se realiza la prueba de forma idéntica para ambos controladores.

Se recogen los resultados exportados de ambas pruebas y se ejecutan los scripts de análisis en modo "movement". Se recortan temporalmente los datos para eliminar los instantes iniciales hasta que se recibe la video detección y los instantes finales desde que finaliza el videoseguimiento.

Para este caso se evalúan matemáticamente los parámetros de error integral y se analizan los tiempos de respuesta en las maniobras, así como los picos en las oscilaciones producidas por estos cambios de maniobra y la estabilidad del seguimiento.

Los parámetros evaluados son:

- **IAE:** Error acumulado total a lo largo del tiempo.
- **ITAE:** Error acumulado multiplicado por el tiempo. Este criterio penaliza errores que ocurran más tarde.
- **ISE:** Error cuadrático a lo largo del tiempo. Este criterio penaliza errores más grandes.
- **ITSE:** Error cuadrático multiplicado por el tiempo. Este criterio penaliza errores más grandes y errores que ocurran más tarde.
- **ISTE:** Error cuadrático multiplicado por el tiempo al cuadrado. Este criterio penaliza errores más grandes y penaliza, en mayor medida, errores que ocurran más tarde.

A continuación, se muestran representadas las respuestas del controlador PID, el controlador FuzzyPID, y la respuesta de ambos controladores comparada una respuesta ideal.

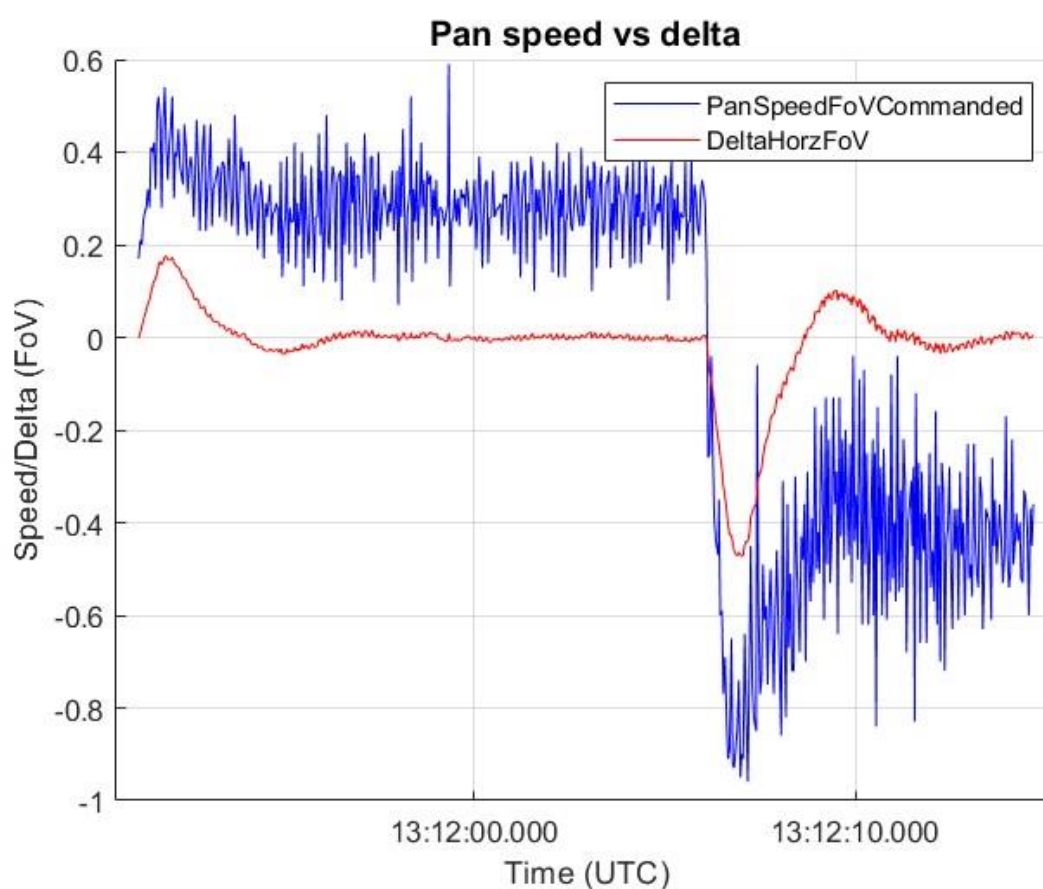


Figura 5.4. Respuesta del controlador PID

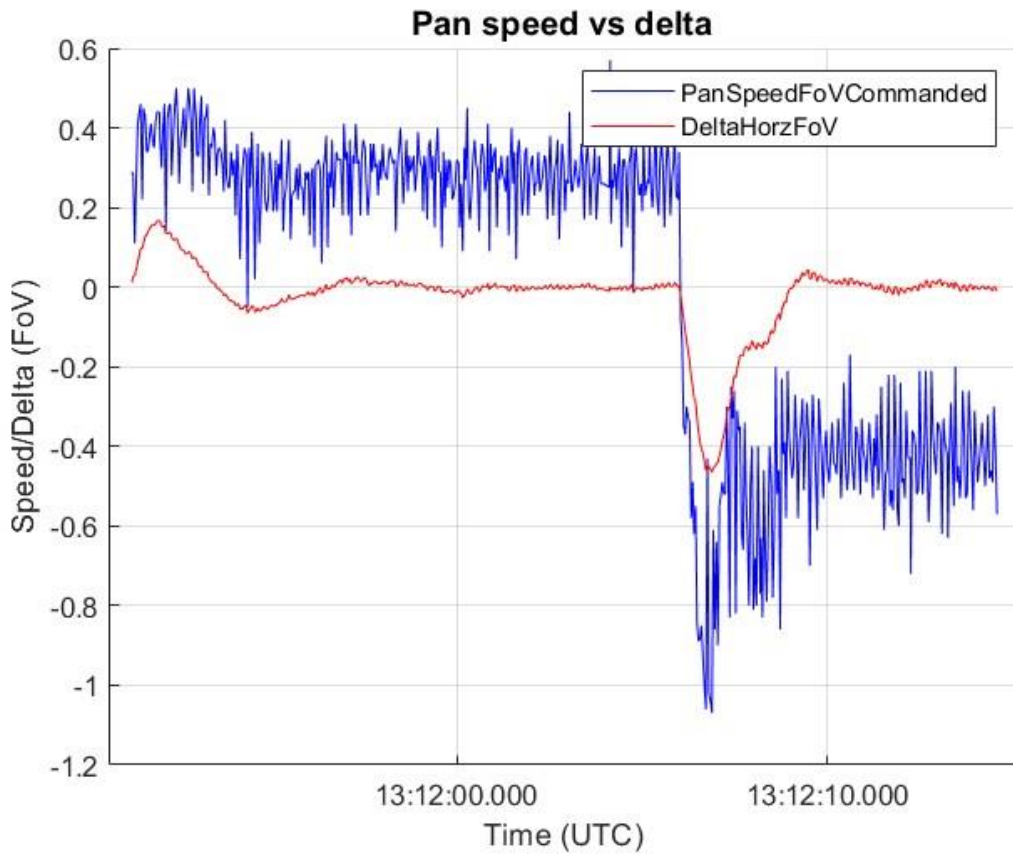


Figura 5.5. Respuesta del controlador Fuzzy PID

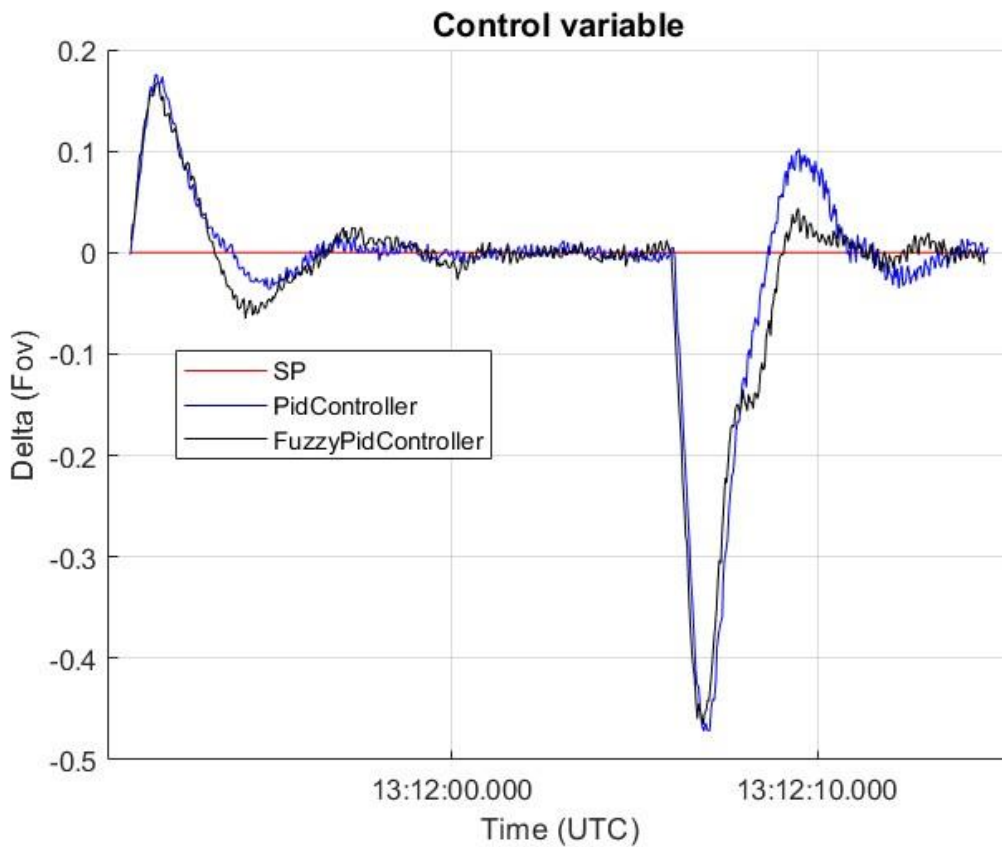


Figura 5.6. Comparativa de respuestas

Se obtienen los siguientes resultados para las respuestas de los controladores.

Controller	IAE	ITAE	ISE	ITSE	ISTE
PID	1.094	13.872	0.244	3.473	55.143
Fuzzy PID	1.094	12.813	0.233	3.249	50.842

Tabla 5.3. Resultados simulación B

Se observa en las respuestas obtenidas como ambos controladores logran un seguimiento muy estable cuando el objetivo se desplaza linealmente.

La diferencia principal entre los controladores es la gestión de los cambios de maniobra.

En el pico inicial que se produce cuando el objetivo comienza a moverse, ambos controladores logran la estabilidad en el mismo periodo de tiempo, con una oscilación ligeramente mayor del controlador Fuzzy PID en la dirección del movimiento.

En el segundo pico se produce el cambio de maniobra de 180 °, donde ambos controladores logran mantener el objetivo dentro de la imagen (con un error menor a 0.5 FoV). En la respuesta de ambos controladores se observa claramente el cambio en el estado del error para el controlador Fuzzy PID, cuando el error cae por debajo de  $\approx 0.12$  FoV, pasando de un error grande a un error medio y controlando la acción para minimizar la oscilación, mientras que el controlador PID presenta una oscilación bastante mayor, lo que le dificulta estabilizar el sistema posteriormente.

En los parámetros de error integral se observa este comportamiento, con un error integral total idéntico, pero con menor valor para el Fuzzy PID en los términos que penalizan errores en instantes de tiempo mayores.

Se puede concluir, por tanto, para esta simulación, que el controlador Fuzzy PID realiza un mejor control ante cambios de maniobra bruscos del objetivo frente al controlador PID, sin perder prestaciones en condiciones nominales de movimiento.

### 5.3 Simulación C: Cambios de maniobra con velocidad máxima

En este escenario se evalúan las prestaciones del algoritmo propuesto ante cambios de maniobra bruscos y a la velocidad máxima definida en el escenario de aplicación, y se compara su respuesta contra un controlador PID. El objetivo realiza maniobras bruscas y se desplaza a velocidad máxima (equivalente a recorrer la pantalla en 8 s) en ambos sentidos.

Para realizar esta simulación, se ejecuta la prueba partiendo de una posición en la que el objetivo se encuentra centrado en la imagen y se desplaza a lo largo de toda la pantalla de forma continua. El objetivo parte de la izquierda y se desplaza hasta la derecha en un tiempo total de 8 s, donde realiza un cambio de sentido de 180° y regresa en un tiempo de 8 s. Se establece un FoV para la cámara de 4.3 ° y se realiza la prueba de forma idéntica para ambos controladores.

## Resultados

Se recogen los resultados exportados de ambas pruebas y se ejecutan los scripts de análisis en modo "movement". Se recortan temporalmente los datos para eliminar los instantes iniciales hasta que se recibe la video detección y los instantes finales desde que finaliza el videoseguimiento.

Para este caso se evalúan matemáticamente los parámetros de error integral y se analizan los tiempos de respuesta en las maniobras, así como los picos en las oscilaciones producidas por estos cambios de maniobra y la estabilidad del seguimiento.

A continuación, se muestran representadas las respuestas del controlador PID, el controlador FuzzyPID, y la respuesta de ambos controladores comparada una respuesta ideal.

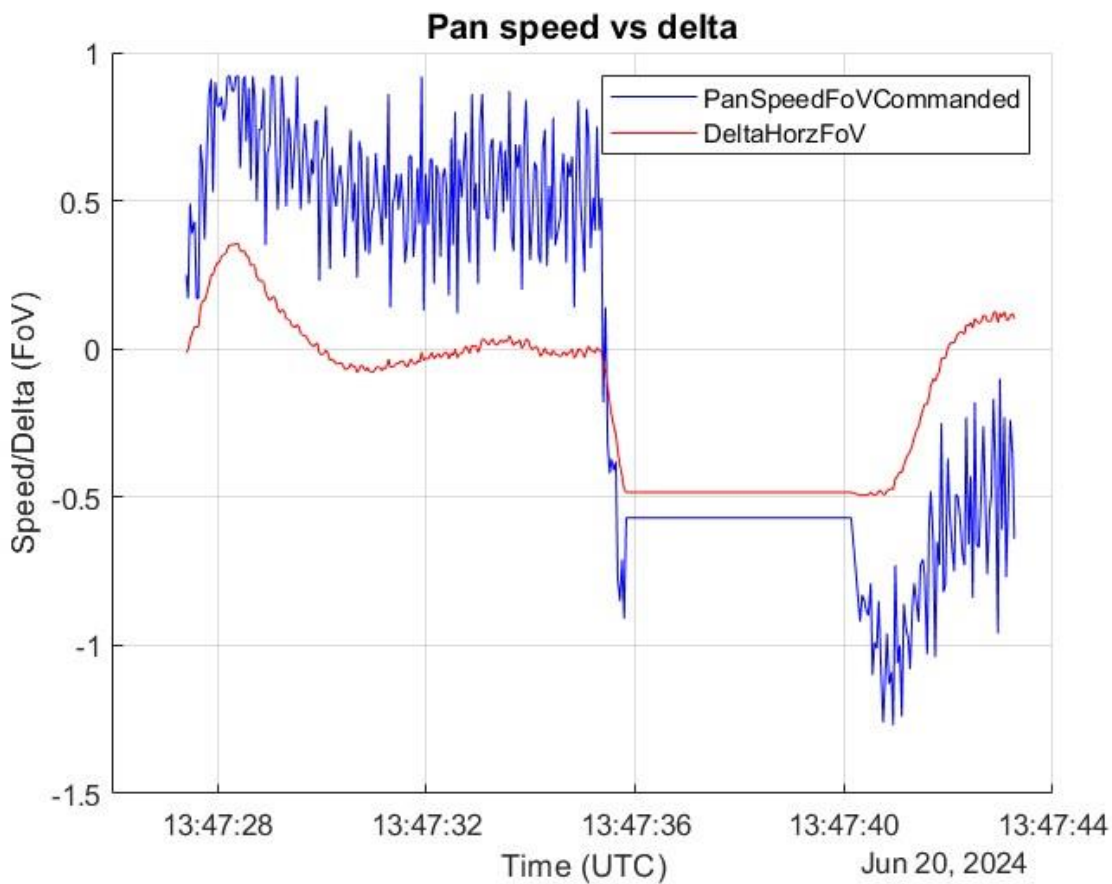


Figura 5.7. Respuesta del controlador PID

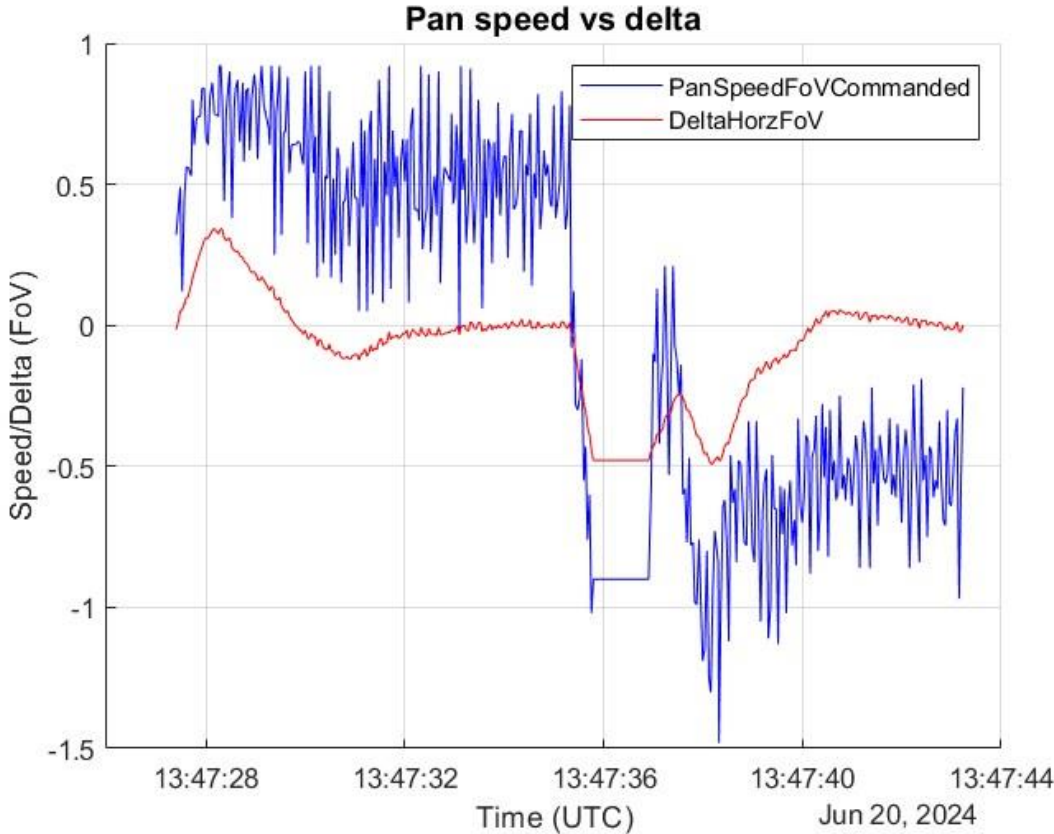


Figura 5.8. Respuesta del controlador Fuzzy PID

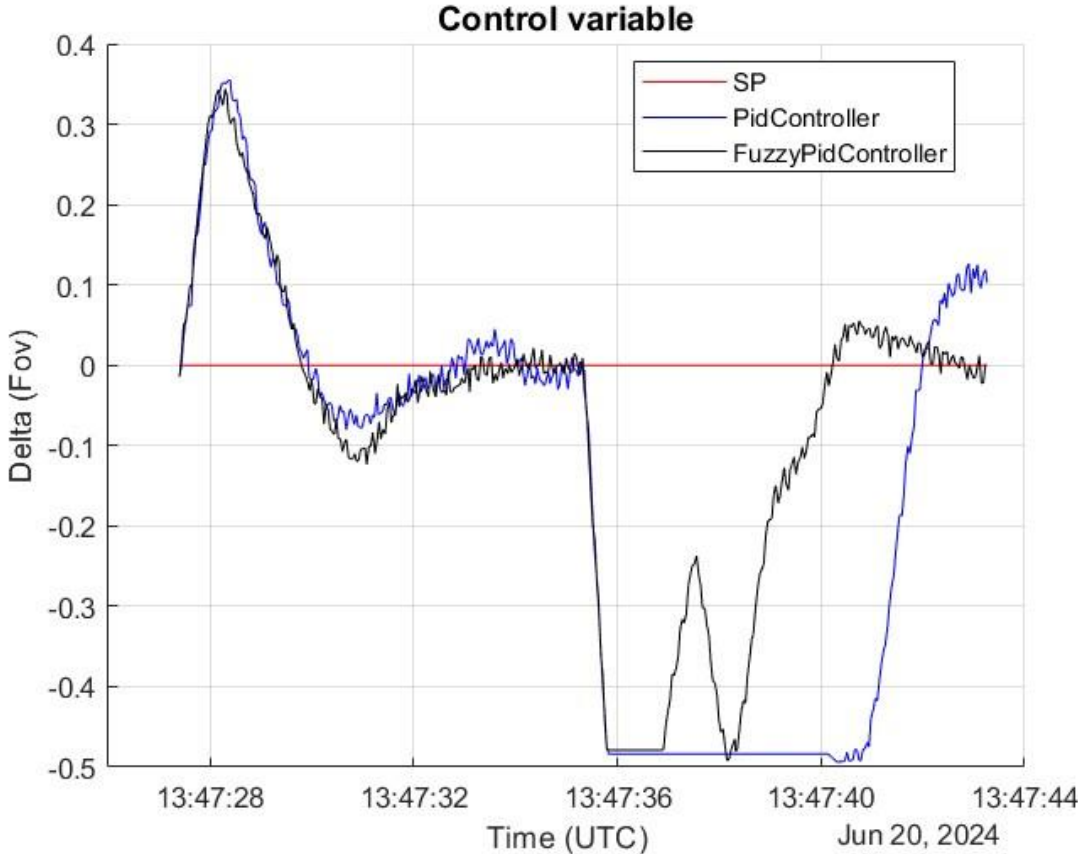


Figura 5.9. Comparativa de respuestas

Se obtienen los siguientes resultados para las respuestas de los controladores.

Controller	IAE	ITAE	ISE	ITSE	ISTE
PID	3.571	34.526	1.453	14.926	168.250
Fuzzy PID	2.244	17.449	0.717	5.963	57.724

Tabla 5.4. Resultados simulación C

En los resultados se observa como ambos controladores tienen un rendimiento similar hasta que se produce la maniobra, en donde la latencia del sistema ante el cambio de maniobra de 180° de un objetivo a gran velocidad provoca una respuesta retardada por parte del controlador, y mientras que el controlador Fuzzy PID es capaz de recuperar el objetivo en un corto periodo de tiempo (menos de 1 segundo), el controlador PID tiene una respuesta bastante más lenta. Esto se debe a que, en el momento de pérdida de la detección, el valor de velocidad comandada del Fuzzy PID es mayor, y su valor comandado en el tiempo en que no hay detección, correspondiente a la última velocidad comandada, es también mayor.

En los parámetros de error integral se observa como el controlador propuesto mantiene un menor error que el controlador PID a lo largo del tiempo, y un menor error cuando se incluye la penalización temporal, indicando que presenta una respuesta predictiva mejor.

Se puede concluir para esta prueba, que el controlador Fuzzy PID realiza una acción de control mayor ante errores grandes frente al controlador PID.

## 5.4 Pruebas de campo

En este apartado se evalúa el rendimiento del controlador propuesto y su viabilidad en condiciones reales de operación en un escenario de seguimiento de aves.

Las pruebas se realizan en un parque eólico ubicado en Tudela, en el que se encuentra implantado un sistema de protección de avifauna compuesto por un radar y un conjunto de cámaras. El sistema se compone de los mismos elementos, sustituyendo el modelo de visión artificial empleado en las simulaciones para la detección del cuadrado por un modelo de visión artificial entrenado para detectar aves, y un radar 3D con 5 km de alcance instrumental para realizar el apuntamiento inicial.

El software de mando y control recibe detecciones proporcionadas por el radar y comanda el apuntamiento de la cámara más cercana. El software controla el movimiento de la cámara con las detecciones del radar y se realiza un escaneo en espiral con la cámara sobre la zona hasta que se recibe una video detección. Una vez obtenida la video detección del objetivo se cede el control de la cámara al algoritmo de control y se inicia el video seguimiento.

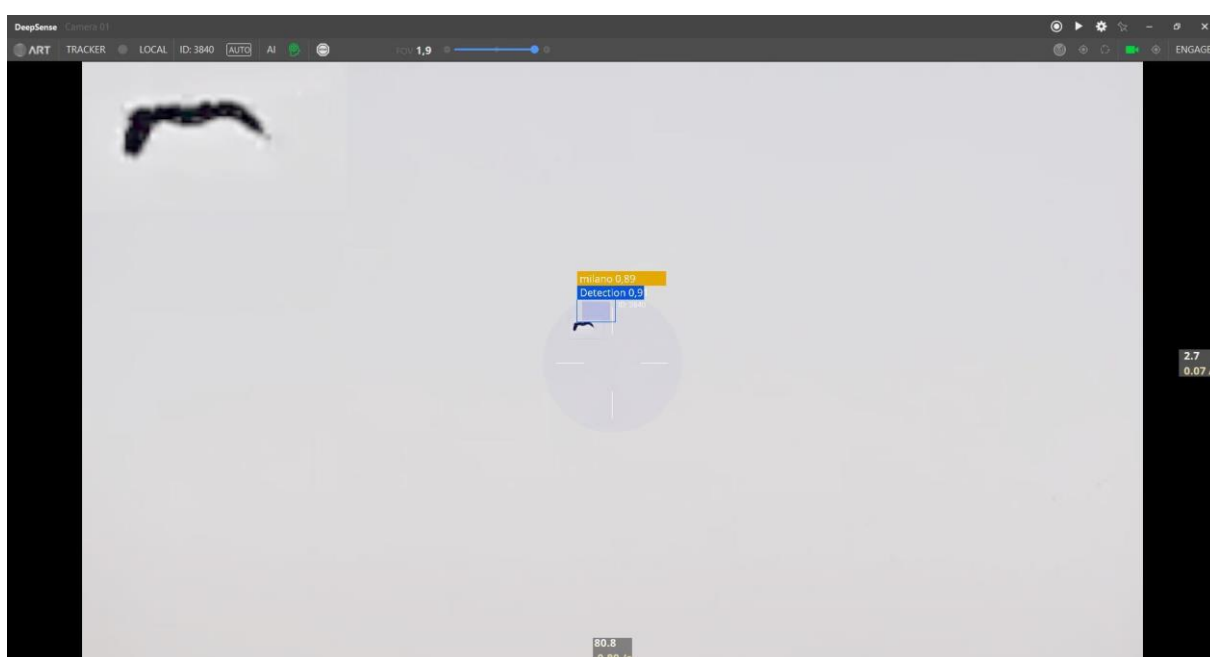


Figura 5.10. Captura del software de vídeo

El sistema está preparado para realizar apuntamientos automáticos sobre las detecciones que se encuentren en un área de 1 km. Si se obtiene una video detección durante el apuntamiento, se inicia la grabación de un vídeo, y continúa el proceso de video seguimiento durante un tiempo configurable (habitualmente 30 o 60 segundos) o hasta que se dejen de recibir video detecciones.

Los parámetros del controlador son los mismos que los definidos para el algoritmo y empleados en las simulaciones, estudiando únicamente la posibilidad de modificar el universo de la tasa de cambio del error, por ser dependiente del tiempo entre acciones de control. Se realizan varios apuntamientos y se estudia este parámetro.

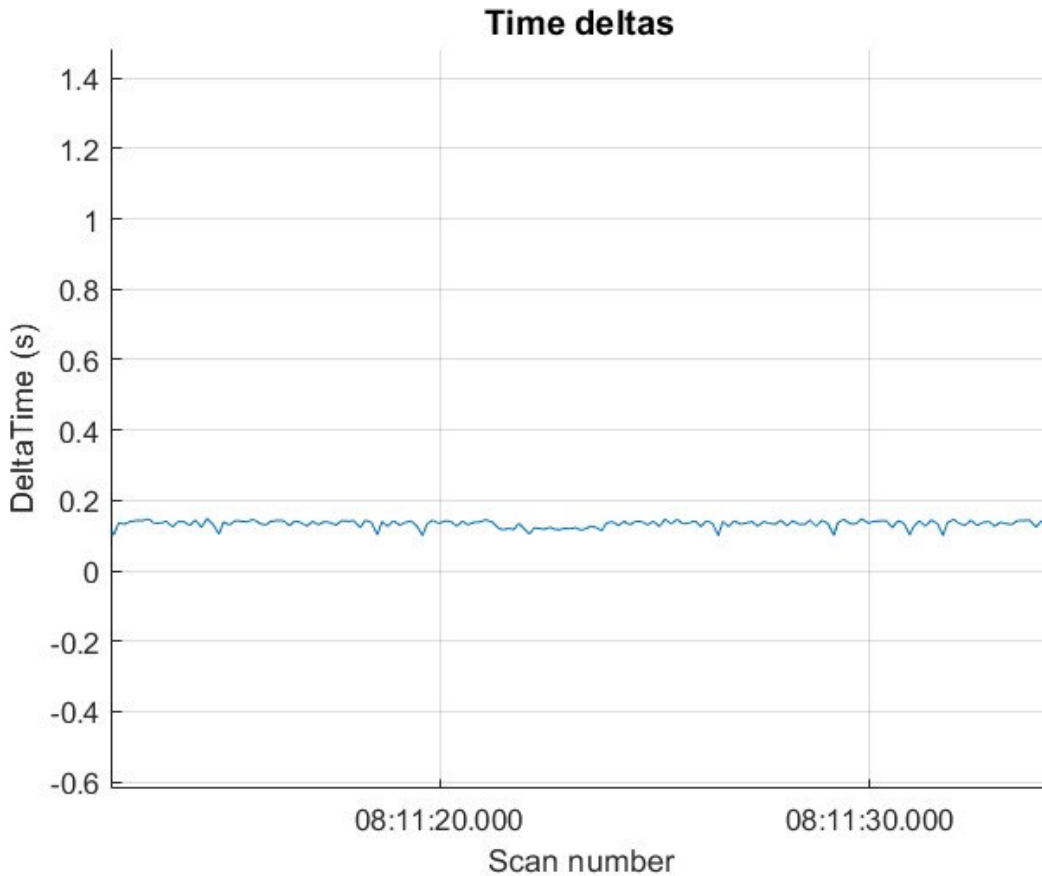


Figura 5.11. Diferencia temporal

Un valor nominal entre acciones de control es de 0.12 s. En el diseño se establecía que el valor del universo de la tasa de cambio del error podía ser de 0.4 FoV/s con diferencias temporales de 0.04 s. Trasladado al nuevo valor de diferencia temporal, se obtiene un valor para este universo de 0.12 FoV/s.

Se configuran los parámetros del controlador y se deja el sistema realizando apuntamientos de forma automática durante unas horas y con el almacenamiento de resultados de video seguimiento activo.

A continuación, se analizan los resultados de dos procesos de video seguimiento de aves, en los que se observa un desplazamiento considerable sobre el eje de Pan.

En el primer caso, el ave se desplaza 25 ° en 52 s durante el proceso de video seguimiento, llegando a visualizarse con un FoV mínimo de 2.15 °.

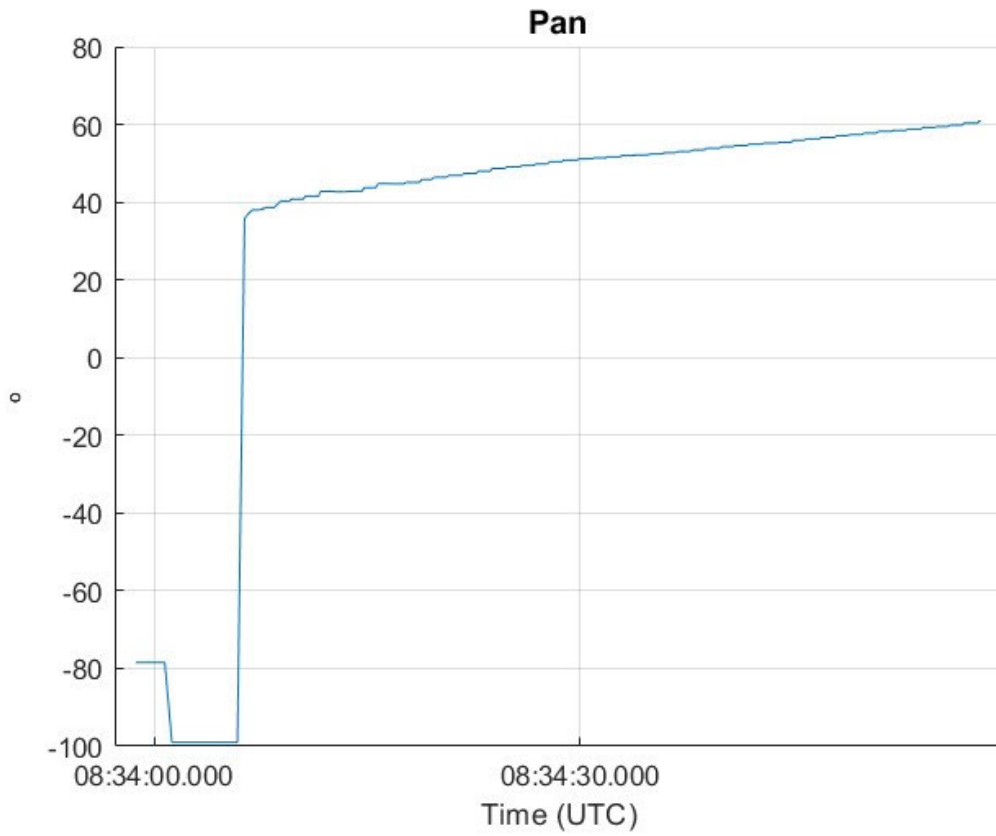


Figura 5.12. Movimiento sobre el eje de Pan

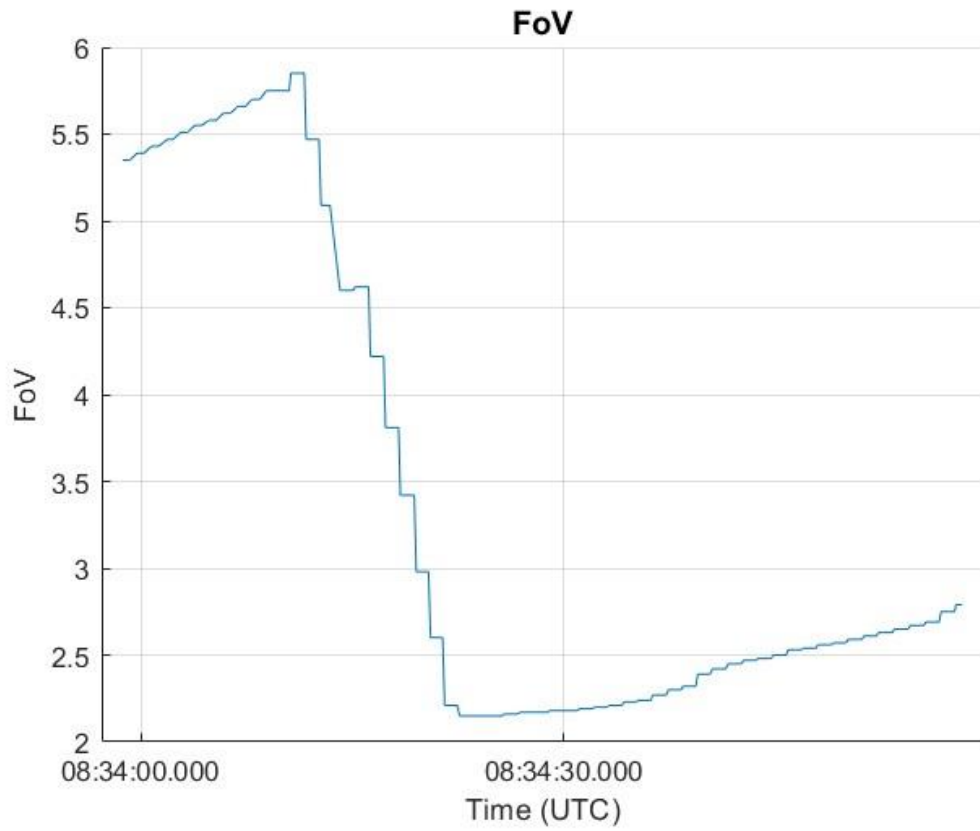
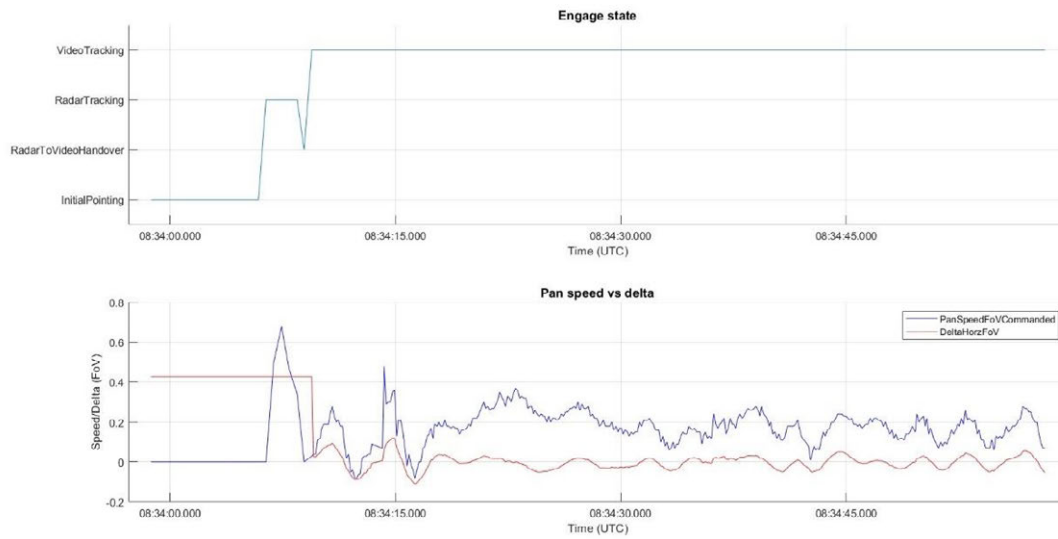


Figura 5.13. Estado del FoV

## Resultados

Se visualiza el estado del seguimiento junto al error al centro y la velocidad comandada en cada instante.



**Figura 5.14. Estado del seguimiento y error de seguimiento**

Se observa como el controlador estabiliza el objetivo en la imagen de forma controlada tras obtener la primera video detección e iniciar el proceso de video seguimiento, manteniendo el error por debajo de 0.2 FoV y alcanzando la estabilidad (error por debajo del 10%) en 8 s.

Se extraen los parámetros de error integral durante el proceso de video seguimiento.

Controller	IAE	ITAE	ISE	ITSE	ISTE
Fuzzy PID	1.353	27.754	0.072	0.984	30.220

**Tabla 5.5. Resultados pruebas de campo**

En el segundo caso, el ave se desplaza  $112.5^\circ$  en 112 s durante el proceso de video seguimiento, llegando a visualizarse con un FoV mínimo de  $2.75^\circ$ .

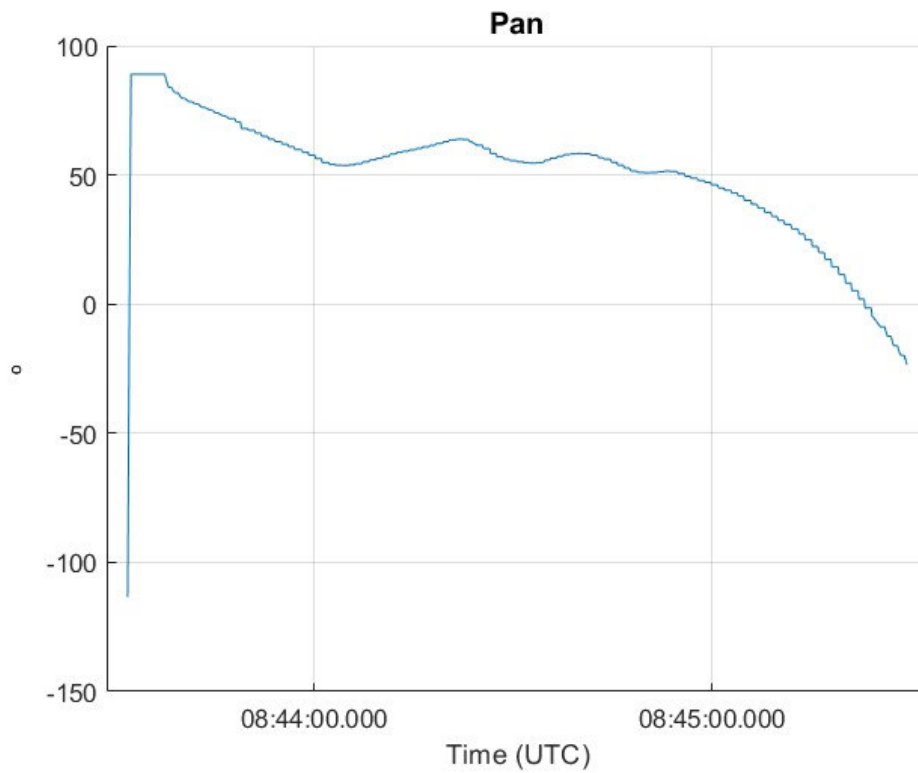


Figura 5.15. Movimiento sobre el eje de Pan

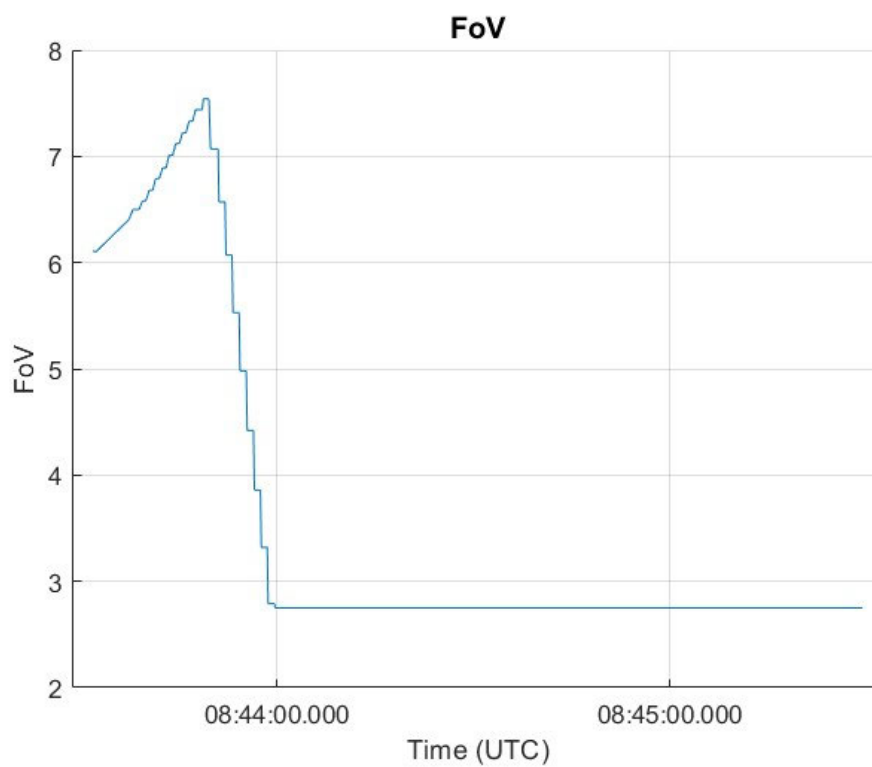


Figura 5.16. Estado del FoV

## Resultados

Se visualiza el estado del seguimiento junto al error al centro y la velocidad comandada en cada instante.

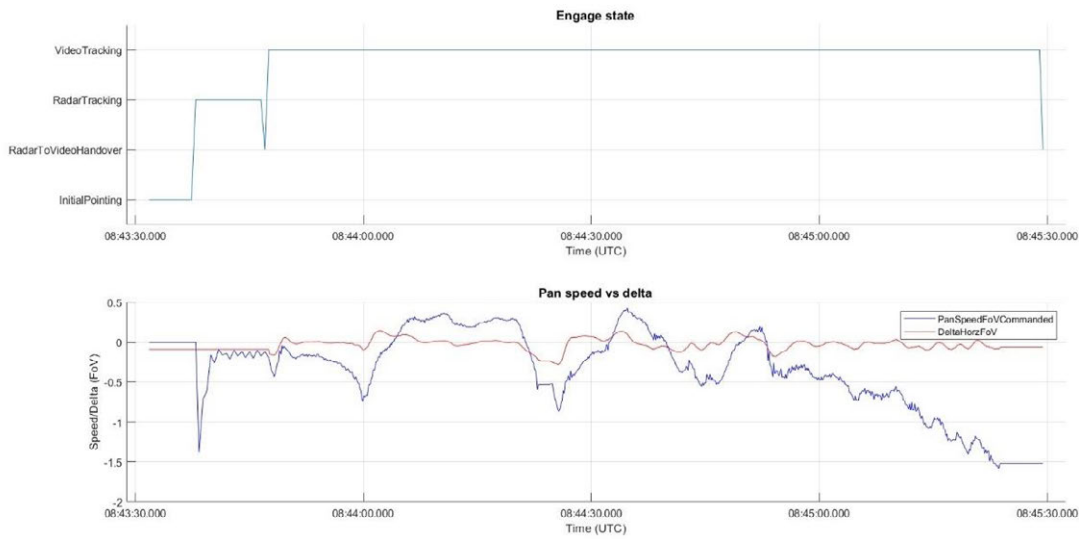


Figura 5.17. Estado del seguimiento y error de seguimiento

Se observa como el controlador estabiliza el objetivo en la imagen de forma controlada tras obtener la primera video detección e iniciar el proceso de video seguimiento, manteniendo el error por debajo de 0.2 FoV y alcanzando la estabilidad (error por debajo del 10%) en 2.5 s.

El controlador es capaz de adaptarse a cambios de velocidad del objetivo (generalmente causados por la geometría del movimiento, que no se produce puramente sobre el eje de Pan), controlando el error de seguimiento por debajo de 0.2 FoV.

Se extraen los parámetros de error integral durante el proceso de video seguimiento.

Controller	IAE	ITAE	ISE	ITSE	ISTE
Fuzzy PID	5.773	292.32	0.614	28.007	1611.9

Tabla 5.6. Resultados pruebas de campo

Con los datos recogidos y los resultados estudiados se comprueba la viabilidad del algoritmo sobre el sistema en operación para el video seguimiento de aves.

## 6. Presupuesto

En este capítulo se detallan los costos del proyecto. Se dividen en tres tipos según su naturaleza: Recursos humanos, recursos hardware y recursos software.

### 6.1 Recursos humanos

Aquí se tienen en cuenta las horas dedicadas para la realización del proyecto.

Dado que la dedicación final al proyecto ha sido de 320 horas y el salario por hora se establece en 12 €/h, el presupuesto asociado a los recursos humanos sería de **3840,00 €**.

### 6.2 Recursos hardware

En la siguiente tabla se detallan los equipos hardware utilizados junto a su coste y su propietario, ya que algunos elementos del proyecto han sido cedidos.

Recurso	Propietario	Coste
Ordenador portátil Asus TUF gaming A15, AMD Ryzen 7 7735HS with Radeon Graphics, 16.0 GB RAM	Advanced Radar Technologies S.A.	1.138,00 €
Ordenador portátil Asus Tp300l, Intel core i3-4030u	Advanced Radar Technologies S.A.	373,00 €
Cámara AXIS Q6225-LE PTZ	Advanced Radar Technologies S.A.	3.449,00 €
Inyector PoE Gigabit Ethernet	Advanced Radar Technologies S.A.	27,69 €
Switch Gigabit Ethernet	Advanced Radar Technologies S.A.	20,50 €
Televisión LG 55UR78006LK 55", 4K UHD	Advanced Radar Technologies S.A.	419,99 €
Guías para soporte de cámara	Advanced Radar Technologies S.A.	65,00 €
Cables HDMI, Ethernet	Advanced Radar Technologies S.A.	57,00 €
<b>TOTAL</b>		<b>5550,18 €</b>

Tabla 6.1. Recursos hardware

### 6.3 Recursos software

En la siguiente tabla se detallan los programas utilizados junto a su coste de licencia y su propietario.

Recurso	Propietario	Coste
Matlab (licencia académica proporcionada por la UPM)	Personal	0,00 €
Da Vinci Resolve	Personal	0,00 €
Visual Studio Community	Advanced Radar Technologies S.A.	0,00 €
Windows x64	Advanced Radar Technologies S.A.	148,18 €
<b>TOTAL</b>		<b>148,18 €</b>

Tabla 6.2. Recursos software

### 6.4 Presupuesto final

En la siguiente tabla se refleja el coste total del proyecto agrupando todos los costes descritos anteriormente.

Recurso	Coste
Recursos humanos	3840,00 €
Recursos hardware	5550,18 €
Recursos software	148,18 €
<b>TOTAL</b>	<b>9538,18 €</b>

Tabla 6.3. Presupuesto final

## 7. Impacto del proyecto

En este capítulo se analizan las implicaciones sociales y de seguridad, de salud, ambientales, económicas, tecnológicas e industriales del proyecto, y su posible aportación a los objetivos de desarrollo sostenible.

Las implicaciones sociales del proyecto están principalmente relacionadas con la seguridad y la privacidad. La mejora en los algoritmos de seguimiento mediante cámaras puede contribuir a la seguridad ciudadana en cuanto a prevención de actividades delictivas, protección de eventos ante ataques terroristas, protección del espacio aéreo...

Estos sistemas de vigilancia pueden vulnerar en la misma medida la privacidad de las personas y se debe por tanto garantizar un uso responsable y controlado de estas aplicaciones.

A nivel de salud, el mayor impacto se produce en la salud mental debido a las preocupaciones de privacidad de los ciudadanos.

En el ámbito ambiental, una mejora en los algoritmos de control de sistemas Pan-Tilt puede traducirse en un menor consumo energético por parte de estos y en alargar su vida útil reduciendo así la huella de carbono. Adicionalmente, este tipo de sistemas se emplean en vigilancia de avifauna en parques eólicos con el objetivo de evitar colisiones contra aerogeneradores.

Las implicaciones económicas están ligadas a los avances tecnológicos y de la industria, ya que mejoras en los sistemas de control pueden reducir costos aumentando así los beneficios. Del mismo modo, en el sector público se puede reducir el gasto en seguridad.

El desarrollo de un algoritmo de control sencillo, económico y adaptable supone una mejora a nivel tecnológico con respecto a los algoritmos de control más comunes, ofreciendo mayores prestaciones sin añadir demasiada complejidad a los sistemas y manteniendo su adaptabilidad. La industria puede beneficiarse de ello consiguiendo mejoras en los procesos de producción, reduciendo así los costos de producción y siendo más competitiva.

A continuación, se expone la relación del proyecto con algunos de los objetivos de desarrollo sostenible (ODS). [43]

- **ODS 8. Trabajo decente y crecimiento económico:** Al tener un impacto económico positivo sobre la industria, el desarrollo y la innovación tecnológicas contribuyen a frenar la desaceleración del PIB mundial.
- **ODS 9. Industria, innovación e infraestructura:** El proyecto se alinea con la rama de fomento de innovación tecnológica.
- **ODS 11. Ciudades y comunidades sostenibles:** La mejora sobre los sistemas de vigilancia contribuye a mejorar la seguridad ciudadana en las ciudades.
- **ODS 16. Paz, justicia e instituciones sólidas:** Del mismo modo, una mejora en los sistemas de vigilancia puede ayudar a prevenir actividades delictivas y ayudar a las autoridades y la justicia en su labor.



## 8. Conclusiones

### 8.1 Conclusiones

El objetivo del proyecto era el planteamiento y la implementación de un algoritmo que suponga una mejora en el video seguimiento de objetivos móviles frente a los algoritmos actuales, especialmente orientado a sistemas de seguimiento de objetivos pequeños, veloces y a gran distancia, así como establecer una metodología para comparar el rendimiento entre algoritmos.

Para abordar este problema, se propuso implementar un controlador Fuzzy PID y compararlo con el tradicional controlador PID, empleado actualmente en soluciones de seguimiento de aves, drones y objetivos marítimos. Adicionalmente, se emplearon técnicas de control de Windup para la protección del controlador ante situaciones en las que la aportación integral se oponga notablemente a la acción necesaria para minimizar el error durante el proceso de seguimiento. El controlador propuesto se ha implementado en un software de mando y control conectado a un software de video encargado de proveer las detecciones de los objetivos en los fotogramas.

Se ha desarrollado un escenario de pruebas automatizado para comparar las prestaciones de los algoritmos ante situaciones idénticas de movimiento de objetivos, y se han definido los parámetros de error a evaluar. Para el proceso de evaluación, se ha dotado al software de mando y control con la capacidad de exportar registros de operación y se han implementado herramientas de análisis para analizar las prestaciones.

De los resultados obtenidos en la experimentación se concluye que el algoritmo propuesto mejora las prestaciones del algoritmo empleado actualmente para esta aplicación, mejorando en tiempo de respuesta (especialmente ante errores muy grandes en el seguimiento del objetivo), minimización de oscilaciones, estabilidad y un seguimiento más suave. Todo esto se traduce en una mejora del video seguimiento especialmente ante cambios de maniobra y movimientos rápidos de objetivos.

Las pruebas realizadas en sistemas reales muestran la viabilidad del algoritmo en estas aplicaciones, logrando mantener en todo momento el objetivo dentro del campo de visión de la cámara con una imagen estable.

Se concluye también que la latencia presentada por el sistema en el que se implementa el controlador supone una degradación de sus prestaciones, incluso en mayor grado que sobre el controlador PID, ya que, dispone de más parámetros dependientes del tiempo, y una desincronización temporal grande puede llevar a acciones desfasadas en el tiempo y contrarias al seguimiento requerido.

## **8.2 Trabajos futuros**

Una línea de trabajo futura podría ser la caracterización del algoritmo para los ejes de Tilt y Zoom en la aplicación estudiada. Relativo al escenario también se podría trabajar sobre el software de vídeo para minimizar la latencia. Actualmente hay equipos trabajando en esta tarea con buenos resultados en fases experimentales.

Como mejora en la caracterización del algoritmo se podría estudiar la posibilidad de iterar sobre los parámetros del PID y los parámetros difusos con el objetivo de encontrar un conjunto de parámetros óptimo para un determinado caso de uso, y estudiar su adaptación a distintos tipos de sistemas.

La línea de trabajo más importante y que podría aportar una mejora importante en las prestaciones del algoritmo propuesto es el estudio de las funciones triangulares de membresía, y evaluar la posibilidad de modificarlas con un ajuste fino para maximizar el rendimiento del controlador con una caracterización de los conjuntos difusos más ajustada a la realidad.

## 9. Referencias

- [1] Aulafacil.com, "Movimientos de Cámara", [En línea]. Disponible en: <https://www.aulafacil.com/cursos/cine-y-teatro/cine/movimientos-de-camara-140028#:~:text=Tilt%20es%20el%20movimiento%20vertical%20que%20hace%20la,la%20c%C3%A1mara%20de%20izquierda%20a%20derecha%20y%20viceversa>. [Último acceso: 10-jun-2024].
- [2] PMD Corp., "Pan Tilt Camera Control Case Study", [En línea]. Disponible en: <https://www.pmdcorp.com/resources/type/case-studies/get/pan-tilt-camera-control-case-study>. [Último acceso: 10-jun-2024].
- [3] Laser Focus World, "Motion Control Accuracy and Precision Rule Pan Tilt Positioners in Critical Camera Applications", [En línea]. Disponible en: <https://www.laserfocusworld.com/software-accessories/positioning-support-accessories/article/16548257/motion-control-accuracy-and-precision-rule-pan-tilt-positioners-in-critical-camera-applications>. [Último acceso: 10-jun-2024].
- [4] Wikipedia, "Controlador PID", [En línea]. Disponible en: [https://es.wikipedia.org/wiki/Controlador\\_PID](https://es.wikipedia.org/wiki/Controlador_PID). [Último acceso: 10-jun-2024].
- [5] National Instruments, "PID Theory Explained", [En línea]. Disponible en: <https://www.ni.com/es/shop/labview/pid-theory-explained.html>. [Último acceso: 10-jun-2024].
- [6] J. B. Hibbeler, "PID Control", [En línea]. Disponible en: <https://www.eng.newcastle.edu.au/~jhb519/teaching/caut1/Apuntes/PID.pdf>. [Último acceso: 10-jun-2024].
- [7] Medium, "PID: P as in Proportional", [En línea]. Disponible en: <https://medium.com/luos/pid-p-as-in-proportional-39cd1e1e8567>. [Último acceso: 10-jun-2024].
- [8] Medium, "PID: The I as in Integral", [En línea]. Disponible en: <https://medium.com/luos/pid-the-i-as-in-integral-4390c71db12e>. [Último acceso: 10-jun-2024].
- [9] L. H. Keel and S. P. Bhattacharyya, "Robust, fragile, or optimal? It depends", Engineering Applications of Artificial Intelligence, vol. 22, no. 3, pp. 351-362, 2009. [En línea]. Disponible en: <https://www.sciencedirect.com/science/article/pii/S095915240900167X>. [Último acceso: 10-jun-2024].
- [10] M. A. Johnson and M. H. Moradi, "Performance Assessment of PID Controllers", [En línea]. Disponible en: [https://www.researchgate.net/publication/228857995\\_Performance\\_Assessment\\_of\\_PID\\_Controllers](https://www.researchgate.net/publication/228857995_Performance_Assessment_of_PID_Controllers). [Último acceso: 10-jun-2024].

## Referencias

---

- [11] Picuino, "Control Ziegler-Nichols", [En línea]. Disponible en: <https://www.picuino.com/es/control-ziegler-nichols.html>. [Último acceso: 10-jun-2024].
- [12] YouTube, "Vídeo: Introducción al Control PID", [En línea]. Disponible en: <https://www.youtube.com/watch?v=ps8KLVg5uiM>. [Último acceso: 10-jun-2024].
- [13] YouTube, "Vídeo: Implementación de un Controlador PID en Arduino", [En línea]. Disponible en: [https://www.youtube.com/watch?v=epMJ2N4IT\\_g](https://www.youtube.com/watch?v=epMJ2N4IT_g). [Último acceso: 10-jun-2024].
- [14] EnergéticaXXI, "Título del Documento", [En línea]. Disponible en: [https://programacionsiemens.com/wp-content/uploads/2013/03/Energ%C3%A9ticaXXI\\_Octubre\\_2005.pdf](https://programacionsiemens.com/wp-content/uploads/2013/03/Energ%C3%A9ticaXXI_Octubre_2005.pdf). [Último acceso: 10-jun-2024].
- [15] ResearchGate, "Vision Navigation Based PID Control for Line Tracking Robot", [En línea]. Disponible en: [https://www.researchgate.net/publication/366760136\\_Vision\\_Navigation\\_Based\\_PID\\_Control\\_for\\_Line\\_Tracking\\_Robot](https://www.researchgate.net/publication/366760136_Vision_Navigation_Based_PID_Control_for_Line_Tracking_Robot). [Último acceso: 10-jun-2024].
- [16] Prototipando, "Título del Documento", [En línea]. Disponible en: [https://view.officeapps.live.com/op/view.aspx?src=http%3A%2F%2Fprototipando.es%2Fphocadownload%2Fpfc%2Fhermes%2Fbackup%2FPROGRAMACION\\_C%2FPID\\_Discreto%2FDiscretised%2520PID%2520Controllers.doc&wdOrigin=BROWSELINK](https://view.officeapps.live.com/op/view.aspx?src=http%3A%2F%2Fprototipando.es%2Fphocadownload%2Fpfc%2Fhermes%2Fbackup%2FPROGRAMACION_C%2FPID_Discreto%2FDiscretised%2520PID%2520Controllers.doc&wdOrigin=BROWSELINK). [Último acceso: 10-jun-2024].
- [17] A. H. Marzouki, Y. Elhillali and M. El Kamili, "Design of fuzzy-PI controller for a SCARA robot in a pick and place task," *Procedia Computer Science*, vol. 113, pp. 136-143, 2017. [En línea]. Disponible en: <https://www.sciencedirect.com/science/article/pii/S1474667017518650>. [Último acceso: 10-jun-2024].
- [18] J. G. Ziegler and N. B. Nichols, "Optimum settings for automatic controllers," *Transactions of the ASME*, vol. 64, no. 11, pp. 759-768, 1942. [En línea]. Disponible en: <https://ieeexplore.ieee.org/document/375281>. [Último acceso: 10-jun-2024].
- [19] Y. Wang, Y. Zhang, L. Lin, H. Zhu and X. Zhang, "The Application of PID-DMC Algorithm with Feedforward Compensation in the PTZ Platform," [En línea]. Disponible en: [https://www.researchgate.net/publication/328005607\\_The\\_Application\\_of\\_PID-DMC\\_Algorithm\\_with\\_Feedforward\\_Compensation\\_in\\_the\\_PTZ\\_Platform](https://www.researchgate.net/publication/328005607_The_Application_of_PID-DMC_Algorithm_with_Feedforward_Compensation_in_the_PTZ_Platform). [Último acceso: 10-jun-2024].
- [20] Instrumentación Digital, "4 Ventajas de los Reguladores PID", [En línea]. Disponible en: <https://www.instrumentaciondigital.es/4-ventajas-de-los-reguladores-pid/>. [Último acceso: 10-jun-2024].

- [21] Integra Sources, "Basics of PID Controllers: Design & Applications", [En línea]. Disponible en: <https://www.integrasources.com/blog/basics-of-pid-controllers-design-applications/#:~:text=Here%20are%20some%20of%20the%20key%20advantages%20of,intuitively%2C%20by%20trial%20and%20error.%20...%20M%C3%A1s%20elementos>. [Último acceso: 10-jun-2024].
- [22] Wikipedia, "Control predictivo por modelo", [En línea]. Disponible en: [https://es.wikipedia.org/wiki/Control\\_predictivo\\_por\\_modelo](https://es.wikipedia.org/wiki/Control_predictivo_por_modelo). [Último acceso: 10-jun-2024].
- [23] do-mpc, "Theory - Model Predictive Control (MPC)", [En línea]. Disponible en: [https://www.do-mpc.com/en/latest/theory\\_mpc.html](https://www.do-mpc.com/en/latest/theory_mpc.html). [Último acceso: 10-jun-2024].
- [24] R. Scattolini, "Multi-Area Predictive Control for Combined Electricity and Natural Gas Systems", [En línea]. Disponible en: [https://www.researchgate.net/publication/242397596\\_Multi-Area\\_Predictive\\_Control\\_for\\_Combined\\_Electricity\\_and\\_Natural\\_Gas\\_Systems](https://www.researchgate.net/publication/242397596_Multi-Area_Predictive_Control_for_Combined_Electricity_and_Natural_Gas_Systems). [Último acceso: 10-jun-2024].
- [25] M. Haddara, S. V. Rakovic, H. Nounou and A. El-Ferik, "An overview of predictive control applications in power electronics", [En línea]. Disponible en: <https://www.sciencedirect.com/science/article/abs/pii/S1751570X20300297>. [Último acceso: 10-jun-2024].
- [26] C. Cai, Y. Wang, Y. Zhang, M. Chen and J. Yu, "Data-driven MPC for nonlinear systems with neural network dynamics model", [En línea]. Disponible en: <https://arxiv.org/pdf/2203.07747>. [Último acceso: 10-jun-2024].
- [27] Universidad de Sevilla, "Contenido del Trabajo Fin de Master/2. Control Predictivo Basado en Modelo (MPC) aplicado a sist", [En línea]. Disponible en: <https://biblus.us.es/bibing/proyectos/abreproy/70242/fichero/Contenido+del+Trabajo+Fin+de+Master%252F2.+Control+Predictivo+Basado+en+Modelo+%28MPC%29+aplicado+a+sist.pdf>. [Último acceso: 10-jun-2024].
- [28] LibreTexts, "MIMO using model predictive control", [En línea]. Disponible en: [https://eng.libretexts.org/Bookshelves/Industrial\\_and\\_Systems\\_Engineering/Chemical\\_Process\\_Dynamics\\_and\\_Controls\\_\(Woolf\)/12%3AMultiple\\_Input\\_Multiple\\_Output\\_\(MIMO\)\\_Control/12.03%3AMIMO\\_using\\_model\\_predictive\\_control#:~:text=Advantages%20of%20MPC%20can%20be%20used,7%20MPC%20is%20able%20to%20handle%20structural%20changes](https://eng.libretexts.org/Bookshelves/Industrial_and_Systems_Engineering/Chemical_Process_Dynamics_and_Controls_(Woolf)/12%3AMultiple_Input_Multiple_Output_(MIMO)_Control/12.03%3AMIMO_using_model_predictive_control#:~:text=Advantages%20of%20MPC%20can%20be%20used,7%20MPC%20is%20able%20to%20handle%20structural%20changes). [Último acceso: 10-jun-2024].
- [29] Cayetano Guerra, "Lógica difusa y sistemas de control", [En línea]. Disponible en: <https://cayetanoguerra.github.io/ia/logicadifusa/Logica%20difusa%20y%20sistemas%20de%20control.pdf>. [Último acceso: 10-jun-2024].
- [30] Wikipedia, "Lógica difusa", [En línea]. Disponible en: [https://es.wikipedia.org/wiki/L%C3%B3gica\\_difusa](https://es.wikipedia.org/wiki/L%C3%B3gica_difusa). [Último acceso: 10-jun-2024].

## Referencias

---

- [31] MathWorks, "Foundations of Fuzzy Logic", [En línea]. Disponible en: <https://www.mathworks.com/help/fuzzy/foundations-of-fuzzy-logic.html>. [Último acceso: 10-jun-2024].
- [32] Philadelphia University, "Lecture 18: Different Types of Membership Functions", [En línea]. Disponible en: [https://www.philadelphia.edu.jo/academics/qhamarsheh/uploads/Lecture%2018 Different %20Types%20of%20Membership%20Functions%201.pdf](https://www.philadelphia.edu.jo/academics/qhamarsheh/uploads/Lecture%2018%20Different%20Types%20of%20Membership%20Functions%201.pdf). [Último acceso: 10-jun-2024].
- [33] IIS, "Fuzzy Logic Based Multifunctional Mobile Robot Navigation System", [En línea]. Disponible en: <https://www.iis.org/CDs2009/CD2009CSC/CISCI2009/PapersPdf/C875XP.pdf>. [Último acceso: 10-jun-2024].
- [34] G. Hernández and D. Berrones, "Lógica Difusa Aplicada al Control Local del Péndulo Invertido," [En línea]. Disponible en: <https://www.semanticscholar.org/paper/L%C3%B3gica-Difusa-Aplicada-al-Control-Local-del-P%C3%A9ndulo-Guillermo-Hern%C3%A1ndez/257d4ac3fc867c0a4b4f96b3a31f2204f5c1e11b>. [Último acceso: 10-jun-2024].
- [35] Instrumentation Tools, "What is Fuzzy Logic? Advantages of Fuzzy Logic", [En línea]. Disponible en: <https://instrumentationtools.com/what-is-fuzzy-logic/#advantages-of-fuzzy-logic>. [Último acceso: 10-jun-2024].
- [36] Axis Communications, "Datasheet Axis Q6225-LE PTZ Camera", [En línea]. Disponible en: <https://www.axis.com/dam/public/66/8e/75/datasheet-axis-q6225-le-ptz-camera-es-ES-441197.pdf>. [Último acceso: 10-jun-2024].
- [37] Axis Communications, "Axis Q6225-LE | Axis Communications", [En línea]. Disponible en: <https://www.axis.com/es-es/products/axis-q6225-le>. [Último acceso: 10-jun-2024].
- [38] DataCamp, "YOLO: Object Detection Explained", [En línea]. Disponible en: <https://www.datacamp.com/es/blog/yolo-object-detection-explained>. [Último acceso: 10-jun-2024].
- [39] S. K. Pandey and M. M. Gupta, "Design of fuzzy PID controllers using modified triangular membership functions," [En línea]. Disponible en: [https://www.researchgate.net/publication/220311061\\_Design\\_of\\_fuzzy\\_PID\\_controllers\\_using\\_modified\\_triangular\\_membership\\_functions](https://www.researchgate.net/publication/220311061_Design_of_fuzzy_PID_controllers_using_modified_triangular_membership_functions). [Último acceso: 10-jun-2024].
- [40] C. C. Lee, "Fuzzy logic in control systems: Fuzzy logic controller. Part I & II," IEEE Transactions on Systems, Man, and Cybernetics, vol. 20, no. 2, pp. 404-435, 1990. [En línea]. Disponible en: <https://ieeexplore.ieee.org/document/556213>. [Último acceso: 10-jun-2024].

- [41] G. Schuster and M. Winkler, "Object-oriented identification of predictive control models," IEEE Transactions on Control Systems Technology, vol. 26, no. 3, pp. 1162-1170, 2018. [En línea]. Disponible en: <https://ieeexplore.ieee.org/document/7979027>. [Último acceso: 10-jun-2024].
- [42] A. Alcocer, A. El Kholly, E. F. Camacho, M. M. M. de Quevedo, F. Sandoval and J. A. Mérida, "Control of multi-area irrigation canals using networked predictive control," IEEE Transactions on Control Systems Technology, vol. 24, no. 4, pp. 1204-1215, 2016. [En línea]. Disponible en: <https://ieeexplore.ieee.org/document/7287949>. [Último acceso: 10-jun-2024].
- [43] Naciones Unidas, "Objetivos de Desarrollo Sostenible", [En línea]. Disponible en: <https://www.un.org/sustainabledevelopment/es/objetivos-de-desarrollo-sostenible/>. [Último acceso: 10-jun-2024].

## Bibliografía

- T. Thanh-Hai, N. Duy-Thinh, V. Duc-Cuong y N. H. Nam, "Application of parameter self-adaptive fuzzy-PI controller to rotary position control", 2016. Disponible en: [https://www.researchgate.net/publication/289951432\\_Application\\_of\\_parameter\\_self-adaptive\\_fuzzy-PI\\_controller\\_to\\_rotary\\_position\\_control](https://www.researchgate.net/publication/289951432_Application_of_parameter_self-adaptive_fuzzy-PI_controller_to_rotary_position_control). [Último acceso: 29-jun-2024].
- Åström, K. J., "Advanced PID Control", 2006. Disponible en: <https://portal.research.lu.se/en/publications/advanced-pid-control>. [Último acceso: 29-jun-2024].
- Rundqwist, Lars., "Anti-Reset Windup for PID Controllers". 1991. Disponible en: <https://www.sciencedirect.com/science/article/pii/S1474667017518650>. [Último acceso: 29-jun-2024].
- MathWorks. (2018). "Understanding Model Predictive Control, Part 1: Why Use MPC?" [Vídeo]. Disponible en: <https://es.mathworks.com/videos/understanding-model-predictive-control-part-1-why-use-mpc--1526484715269.html>. [Último acceso: 10-jun-2024].
- Navarrete, D. R. (2019). "Análisis comparativo de controladores clásicos versus controladores avanzados aplicados a una planta real de control de nivel de dos tanques no interactivos bajo la plataforma de LabVIEW". Disponible en: <https://www.dspace.espol.edu.ec/handle/123456789/52559>. [Último acceso: 29-jun-2024]



## Anexo

### A.1 Datos generados

Datos generados por el software de mando y control al iniciar un proceso de seguimiento.

#### A.1.1. Parámetros de la cámara

Fichero en formato XML que contiene los parámetros de configuración de la cámara, con nombre *EngageLog\_<año>-<mes>-<día>\_<hora>-<minuto>-<segundo>\_TrackId\_<ID>.xml*. A continuación, se presenta un ejemplo de parte de este documento.

```
<?xml version="1.0" encoding="utf-8"?>
<parameters>
<!--Category 01. General-->
<parameter name="name" type="string">Camera 01</parameter>
<parameter name="model" type="string">AxisQ6215</parameter>

...

<!--Category 6.3 Engage Controller: Octopus - Video tracking-->
<parameter name="octopus_pTermPidPan" type="double">1.77</parameter>
<parameter name="octopus_iTermPidPan" type="double">1.52</parameter>
<parameter name="octopus_dTermPidPan" type="double">0.522</parameter>
<parameter name="octopus_pTermPidTilt" type="double">0.7</parameter>
<parameter name="octopus_iTermPidTilt" type="double">0.1</parameter>
<parameter name="octopus_dTermPidTilt" type="double">0.2</parameter>
<parameter name="octopus_alpha" type="double">1</parameter>
<parameter name="octopus_videoTrackLockedThreshold"
type="double">0.1</parameter>
<parameter name="octopus_dynamicZoomAdjustment"
type="boolean">True</parameter>
<parameter name="octopus_minimumTargetFov" type="double">35</parameter>
<parameter name="octopus_zoomAdjustmentSpeed" type="double">0.1</parameter>

<!--Category 6.3 Engage Controller: Octopus - Video tracking - Fuzzy PID Controller-->
<parameter name="fuzzyControllerEnabled" type="boolean">True</parameter>
<parameter name="octopus_errorUniverse" type="double">0.2</parameter>
<parameter name="octopus_deltaErrorUniverse" type="double">0.4</parameter>
<parameter name="octopus_proportionalUniverse" type="double">0.62</parameter>
<parameter name="octopus_integralUniverse" type="double">0.532</parameter>
<parameter name="octopus_derivativeUniverse" type="double">0.183</parameter>
<parameter name="outputControl" type="boolean">False</parameter>
<parameter name="octopus_outputCorrectionUniverse" type="double">0.2</parameter>

<!--Category 6.3 Engage Controller: Octopus - Video tracking - Maneouver detector-->
<parameter name="maneouverDetectorEnabled" type="boolean">True</parameter>
</parameters>
```

### A.1.2. Datos del seguimiento

Fichero en formato CSV que registra los datos de un proceso de seguimiento, con nombre *EngageLog\_<año>-<mes>-<día>\_<hora>-<minuto>-<segundo>\_TrackId\_<ID>.csv*.

A continuación, se presentan las cabeceras de este documento.

Timestamp,EngageState,RadarSourceActive,RadarLocked,ScanActive,VideoTrackingActive,VideoTrackingLocked,PtrTimestamp,PtrScenarioCoordinate1,PtrScenarioCoordinate2,PtrScenarioCoordinate3,Pan,Tilt,PtrRange,Fov,FovVertical,PanSpeedCommanded,TiltSpeedCommanded,ZoomSpeedCommanded,PanSpeedFovCommanded,TiltSpeedFoVCommanded,PanSpeedMeasured,TiltSpeedMeasured,ZoomSpeedMeasured,DeltaPan,DeltaTilt,DeltaRange,DeltaPanRaw,DeltaTiltRaw,DeltaRangeRaw,UserDeltaPan,UserDeltaTilt,UserDeltaRange,ScanDeltaPan,ScanDeltaTilt,ScanDeltaRange,TrackTimeStamp,TrackScenarioTimeStamp,DeviceId,DeviceScenarioCoordinate1,DeviceScenarioCoordinate2,DeviceScenarioCoordinate3,TrackCoordinate1,TrackCoordinate2,TrackCoordinate3,TrackScenarioCoordinate1,TrackScenarioCoordinate2,TrackScenarioCoordinate3,Azimuth,RadarRange,HeightFromSensor,Elevation,VelAxis1,VelAxis2,VelAxis3,VideoTrackTimeStamp,ScanNumber,VideoTrackId,UpdateType,VideoDetectionX,VideoDetectionY,VideoDetectionWidth,VideoDetectionHeight,DeltaHorizontalFov,DeltaVerticalFov,ManeuverDetected
--

### A.1.3. Datos del controlador difuso

Fichero en formato CSV que registra los datos del controlador difuso durante un seguimiento, con nombre *EngageLog\_<año>-<mes>-<día>\_<hora>-<minuto>-<segundo>\_logFuzzy.csv*.

A continuación, se presentan las cabeceras de este documento.

Timestamp,Error,DeltaError,SubsetE,SubsetEc,Up,Ui,Ud,FuzzyP,FuzzyI,FuzzyD,OutputCorrection
--