

**A. A. Rodriguez**

**Modelado neuro-difuso y control lateral de un tractor robot**

**2026**



**Universidad Autónoma del  
Estado de Hidalgo**

**Instituto de Ciencias Básicas e  
Ingeniería**

**Modelado neuro-difuso de intervalo tipo 2 y  
control lateral para el seguimiento de  
trayectorias de un tractor robot**

**Tesis**

Que como parte de los requisitos para obtener el Grado  
de

**Doctor en Ciencias Computacionales**

Presenta

**M. en M. Adrián Alberto Rodríguez**

**Dirigido por:**

**Dr. Virgilio López Morales**

**Dr. Julio César Ramos Fernández**

**Pachuca de Soto, Hgo. a 24 de marzo de 2026.**



**Universidad Autónoma del Estado de  
Hidalgo**

**Instituto de Ciencias Básicas e Ingeniería**

---

**Doctorado en Ciencias Computacionales**

**Modelado neuro-difuso de intervalo tipo 2 y control lateral  
para el seguimiento de trayectorias de un tractor robot**

**Tesis**

Que como parte de los requisitos para obtener el Grado de

**Doctor en Ciencias Computacionales**

Presenta

**M. en M. Adrián Alberto Rodríguez**

Dirigido por

**Dr. Virgilio López Morales**

**Dr. Julio César Ramos Fernández**

Dr. Víctor Estrada Manzo  
Presidente

Dr. Manuel Alejandro Ojeda Misses  
Secretario

Dr. Virgilio López Morales  
Vocal

Dr. Julio César Ramos Fernández  
Suplente

Dra. Anilú Franco Árcega  
Suplente

Área Académica de Computación y Electrónica, Pachuca de Soto, Hgo. México  
Fecha de aprobación 24 de marzo de 2026



# Universidad Autónoma del Estado de Hidalgo

Instituto de Ciencias Básicas e Ingeniería

School of Engineering and Basic Sciences

Área Académica de Computación y Electrónica

Mineral de la Reforma, Hgo., 24 de marzo del 2026

Número de control: ICBI-AACyE/234/2026

Asunto: Autorización de impresión de tema de tesis.

**MTRA. OJUKY DEL ROCÍO ISLAS MALDONADO**  
**DIRECTORA DE ADMINISTRACIÓN ESCOLAR DE LA UAEH**

El Comité Tutorial de la **TESIS** del Programa Educativo de Posgrado titulado **"Modelado neuro-difuso de intervalo tipo 2 y control lateral para el seguimiento de trayectorias de un tractor robot"**, realizado por el sustentante **Adrián Alberto Rodríguez**, con número de cuenta 296001, perteneciente al programa de **Doctorado en Ciencias Computacionales**, una vez que se ha revisado, analizado y evaluado el documento recepcional de acuerdo a lo estipulado en el Artículo 110 del Reglamento de Estudios de Posgrado, tiene a bien extender la presente: **AUTORIZACIÓN DE IMPRESIÓN**

Por lo que el sustentante deberá cumplir con los requisitos del Reglamento de Estudios de Posgrado y con lo establecido en el proceso de grado vigente.



Atentamente  
"Amor, Orden y Progreso"

**Dr. Virgilio López Morales**  
**Director de Tesis**

ÁREA ACADÉMICA DE  
COMPUTACIÓN Y ELECTRÓNICA

	Comité Tutorial	
Dr. Víctor Estrada Manzo	Presidente	UAEH
Dr. Manuel Alejandro Ojeda Misses	Secretario	UAEH
Dr. Virgilio López Morales	Vocal	UAEH
Dr. Julio César Ramos Fernández	Primer Suplente	UAEH
Dra. Anilu Franco Arcega	Segunda Suplente	UAEH

APL/KAA

Ciudad del Conocimiento, Carretera Pachuca-Tulancingo  
Km. 4.5 Colonia Carboneras, Mineral de la Reforma, Hidalgo,  
México. C.P 42184  
Teléfono: 52 (771) 71 720 00 Ext. 40052, 40053  
aacye\_icbi@uaeh.edu.mx

"Amor, Orden y Progreso"



uaeh.edu.mx

Esta tesis está dedicada a quienes me han acompañado en cada paso de este desafiante y maravilloso recorrido. A mis padres, por su apoyo incondicional, por enseñarme el valor del esfuerzo y la perseverancia, y por creer en mis sueños incluso cuando parecían inalcanzables. Gracias por darme las herramientas para ser quien soy hoy. A mis profesores y mentores, por inspirarme a buscar siempre el conocimiento y nunca conformarme con lo fácil. Su guía y sus enseñanzas han dejado una huella imborrable en mi vida académica y personal. A mis amigos y colegas, por las largas conversaciones, el intercambio de ideas, y el apoyo inquebrantable durante los momentos difíciles. Y a todos aquellos que creen en un futuro donde la tecnología y la agricultura se encuentran para mejorar la calidad de vida de las personas, por ser mi fuente de inspiración constante y recordarme la importancia del trabajo que realizo.

# Agradecimientos

Quisiera expresar mi más sincero agradecimiento a todas las personas e instituciones que hicieron posible la realización de esta tesis:

A **CONAHCYT** por la beca de doctorado otorgada durante el periodo julio 2021 - junio 2024, con número de becario y registro No. 887768.

A la Universidad Politécnica de Pachuca y al **Laboratorio Nacional en Vehículos Autónomos y Exoesqueletos (LANAVEX)** por proporcionar los recursos y el entorno para el desarrollo de esta investigación. Su apoyo fue fundamental para llevar a cabo este proyecto.

A mis asesores, el Dr. Virgilio López Morales y el Dr. Julio César Ramos Fernández, por su guía experta y constante motivación. Sus valiosas ideas y sugerencias me permitieron enfrentar los desafíos de este trabajo con confianza y determinación.

A mi familia y amigos, quienes siempre han estado a mi lado brindándome ánimo y apoyo emocional. Gracias por su paciencia y comprensión durante las largas jornadas de trabajo, y por siempre estar ahí para darme fuerzas cuando más lo necesitaba.

Finalmente, agradezco a todas aquellas personas que, de una u otra forma, contribuyeron a la realización de esta tesis. Su apoyo, ya sea directo o indirecto, ha sido invaluable y siempre estaré agradecido por ello.

## Abreviaturas y siglas

Siglas	Descripción
ANFIS	Sistema adaptable de inferencia neuro-difuso
BP	Retropropagación
CAN	Red de área del controlador
EBC	Controlador basado en el error
EIASC	Algoritmo mejorado iterativo con condición de parada
EKMA	Algoritmo mejorado de Karnik-Mendel
E/S	Entrada/salida
FLC	Controlador de lógica difusa
FLS	Sistema de lógica difusa
FOU	Huella de incertidumbre
FS	Conjunto difuso
GBSMF	Función de membresía en forma de campana generalizada
GD	Descenso del gradiente
GLC	Controlador geométrico lateral
GPS	Sistema de posicionamiento global
IASC	Algoritmo iterativo con condición de parada
IMU	Unidad de medición inercial
IT2	Intervalo tipo 2
IT2-ANFIS	Sistema adaptable de inferencia neuro-difuso de intervalo tipo 2
IT2-ANFIS OSTR	Sistema adaptable de inferencia neuro-difuso de intervalo tipo 2 con el algoritmo de reducción de tipo de un solo paso
IT2FLC	Controlador de lógica difusa de intervalo tipo 2
IT2FLS	Sistema de lógica difusa de intervalo tipo 2
IT2FS	Conjunto difuso de intervalo tipo 2
KF	Filtro de Kalman
KMA	Algoritmo de Karnik-Mendel
LANAVEX	Laboratorio nacional en vehículos autónomos y exoesqueletos
LMF	Función de membresía inferior
LMI	Desigualdades matriciales lineales
LSE	Estimación de mínimos cuadrados
MF	Función de membresía
M-KMA	Algoritmo de Karnik-Mendel modificado
NED	Norte, Este y Abajo
OSTRA	algoritmo de reducción de tipo de un solo paso
PC	Computadora personal
PD	Proporcional, derivativo

PI	Proporcional, integral
PID	Proporcional, integral y derivativo.
PIDC	Control de distancia proporcional, integral
PIVC	Control de velocidad proporcional, integral
PPA	Algoritmo de persecución pura
RC	Radio control
RMSE	Raíz del error cuadrático medio
RTK-GNSS	Sistema global de navegación por satélite cinemático en tiempo real
T1	Tipo 1
T1-ANFIS	Sistema adaptable de inferencia neuro-difuso de tipo 1
T1FLC	Controlador de lógica difusa de tipo 1
T1FLS	Sistema de lógica difusa de tipo 1
T1FS	Conjunto difuso de tipo 1
T2	Tipo 2
T2-ANFIS	Sistema adaptable de inferencia neuro-difuso de tipo 2
T2FLS	Sistema de lógica difusa de tipo 2
T2FS	Conjunto difuso de tipo 2
TM	Tiempo de muestreo
TR	Tractor robot
TRA	Algoritmo de reducción de tipo
T-S	Takagi-Sugeno
UMF	Función de membresía superior

## Resumen

Esta tesis aborda el control lateral para seguimiento de trayectorias en vehículos a escala para agricultura de precisión. El objetivo es mejorar el control direccional ante incertidumbres y perturbaciones, sin modelado dinámico complejo. Para ello, se generaron controladores difusos automáticamente mediante un sistema adaptable de inferencia neuro-difuso de intervalo tipo 2 (IT2-ANFIS, por sus siglas en inglés) y el algoritmo de reducción de tipo de un solo paso (OSTRA, por sus siglas en inglés). La metodología requirió adecuar un Tractor Robot (TR) experimental para aislar el desempeño direccional, integrando datos de un sistema global de navegación por satélite cinemático en tiempo real (RTK-GNSS, por sus siglas en inglés) mediante arquitectura distribuida. El algoritmo OSTRA reduce los costos computacionales frente a métodos iterativos clásicos y mejora la precisión respecto al algoritmo de Karnik-Mendel modificado (M-KMA, por sus siglas en inglés). Asimismo, se diseñó un simulador cinemático para sortear las limitaciones de actuación longitudinal del TR, explorando el modelado IT2-ANFIS OSTRA en controladores de velocidad acoplados a la dinámica lateral. Pruebas experimentales a velocidad constante demostraron que este enfoque supera a estrategias clásicas como el sistema adaptable de inferencia neuro-difuso de tipo 1 (T1-ANFIS, por sus siglas en inglés) y el controlador geométrico lateral (GLC, por sus siglas en inglés), obteniendo una menor raíz del error cuadrático medio (RMSE, por sus siglas en inglés) en trayectorias lineales y curvas.

**Palabras clave:** Control difuso, IT2-ANFIS OSTRA, Tractor Robot experimental, seguimiento de trayectorias, reducción de tipo.

## Abstract

This thesis addresses lateral control for automatic trajectory tracking in experimental-scale vehicles for precision agriculture. The objective is to improve directional control against uncertainties and disturbances, avoiding complex dynamic modeling. To achieve this, fuzzy controllers were automatically generated using an Interval Type-2 Adaptive Neuro-Fuzzy Inference System (IT2-ANFIS) and the One-Step Type Reduction Algorithm (OSTRA). The methodology involved adapting an experimental Robot Tractor (RT) to isolate steering performance, integrating data from a real-time kinematic global navigation satellite system (RTK-GNSS) via distributed architecture. The proposed OSTRA algorithm reduces computational costs compared to classical iterative methods and improves accuracy over the modified Karnik-Mendel algorithm (M-KMA). Additionally, a kinematic simulator was designed to bypass the RT's longitudinal actuation limits, exploring IT2-ANFIS OSTRA modeling in speed controllers coupled with lateral dynamics. Constant-speed experimental tests demonstrated that this approach outperforms classical strategies such as the Type-1 Adaptive Neuro-Fuzzy Inference System (T1-ANFIS) and the Geometric Lateral Controller (GLC), achieving a lower root mean squared error (RMSE) in linear trajectories and curved routes.

**Keywords:** Fuzzy control, IT2-ANFIS OSTRA, experimental Robot Tractor, trajectory tracking, type reduction.

# Índice

<b>Índice de Tablas</b>	<b>3</b>	
<b>Índice de Figuras</b>	<b>5</b>	
<b>1</b>	<b>Introducción</b>	<b>9</b>
1.1	Planteamiento del problema	10
1.2	Justificación	11
1.3	Antecedentes	12
1.4	Hipótesis	16
1.5	Objetivos	16
1.5.1	Objetivo general	16
1.5.2	Objetivos específicos	16
1.6	Alcances y delimitaciones	17
1.7	Aportaciones	17
1.8	Estructura de la tesis	18
<b>2</b>	<b>Marco Teórico</b>	<b>19</b>
2.1	Sistemas de lógica difusa	19
2.1.1	Modelo Mamdani	19
2.1.2	Modelo Takagi-Sugeno	19
2.2	Conjuntos difusos y funciones de membresía tipo 1	20
2.3	T1-ANFIS	21
2.3.1	Estructura T1-ANFIS	22
2.3.2	Algoritmos de optimización para T1-ANFIS	24
2.4	Conjuntos difusos y funciones de membresía Tipo 2	24
2.4.1	Conjuntos Difusos de Intervalo Tipo 2	26
2.4.2	Algoritmos de reducción de tipo	27
2.5	T2-ANFIS	31
2.5.1	Estructura IT2-ANFIS	32
2.6	Controladores de lógica difusa	35
2.7	Modelo matemático Ackermann	35
2.8	Algoritmo de persecución pura	37
2.9	Raíz del error cuadrático medio (RMSE)	39
<b>3</b>	<b>Metodología</b>	<b>41</b>
3.1	Introducción	41
3.2	Desarrollo de una nueva estructura IT2-ANFIS	41
3.2.1	Desarrollo del algoritmo de reducción de tipo de un solo paso (OSTRA)	42
3.2.2	Validación de la estructura IT2-ANFIS OSTRA	43
3.3	Implementación del algoritmo de persecución pura en el TR	43

3.4	Desarrollo de un ambiente de simulación . . . . .	44
<b>4</b>	<b>Resultados y discusión</b>	<b>45</b>
4.1	Nueva estructura IT2-ANFIS OSTR A . . . . .	45
4.1.1	Pruebas iniciales con T1-ANFIS . . . . .	46
4.1.2	Desarrollo del OSTR A . . . . .	53
4.1.3	Experimentos preliminares con IT2-ANFIS . . . . .	54
4.1.4	Validación de IT2-ANFIS OSTR A . . . . .	66
4.2	Seguimiento de trayectorias . . . . .	76
4.3	Prueba de estabilidad . . . . .	89
4.3.1	Prueba de estabilidad ante variaciones de velocidad . . . . .	97
4.3.2	No linealidad del actuador e invarianza dimensional . . . . .	98
4.4	Ambiente de simulación . . . . .	99
<b>5</b>	<b>Conclusiones</b>	<b>111</b>
	<b>Referencias bibliográficas</b>	<b>113</b>
<b>A</b>	<b>Modificaciones al TR</b>	<b>121</b>
A.1	Justificación de las modificaciones . . . . .	121
A.1.1	Actualización del sistema de control . . . . .	122
A.1.2	Implementación de una arquitectura modular . . . . .	126
A.1.3	Integración de sistemas de comunicación avanzados . . . . .	131
A.1.4	Incorporación de sensores de alta precisión . . . . .	134
A.1.5	Implementación del IT2FLS modelado utilizando IT2-ANFIS OSTR A . . . . .	137
<b>B</b>	<b>Contribuciones al estado del arte</b>	<b>141</b>

## Índice de Tablas

2	Comparación del modelo Mamdani con el modelo T-S . . . . .	20
3	Parámetros $a, b, c$ y $h$ de las distintas MF de la Figura 5. . . . .	28
4	Tabla de verdad para diferentes números de entradas y de MF. . . . .	33
5	Plataformas T1-ANFIS comparadas y sus algoritmos de optimización. . . . .	47
6	Parámetros premisa y consecuentes de los modelos obtenidos con las plataformas T1-ANFIS para el Ejemplo 1. . . . .	48
7	RMSE de entrenamiento y tiempo de cómputo obtenidos con las plataformas T1-ANFIS para el Ejemplo 1. . . . .	48
8	Parámetros premisa y consecuentes de los modelos obtenidos con las plataformas T1-ANFIS para el Ejemplo 2. . . . .	50
9	RMSE de entrenamiento y tiempo de cómputo obtenidos con las plataformas T1-ANFIS para el Ejemplo 2. . . . .	51
10	Parámetros premisa y consecuentes de los modelos obtenidos con las plataformas T1-ANFIS para el Ejemplo 3. . . . .	52
11	RMSE de entrenamiento y tiempo de cómputo obtenidos con las plataformas T1-ANFIS para el Ejemplo 3. . . . .	52
12	Resultados para un IT2FS: gaussiana con media incierta. . . . .	55
13	Resultados para un IT2FS: gaussiana con desviación típica incierta. . . . .	55
14	Resultados para un IT2FS: gaussiana con media incierta (lado derecho). . . . .	55
15	RMSE de entrenamiento de las plataformas T1-ANFIS, IT2-ANFIS M-KMA e IT2-ANFIS para el Ejemplo 4, considerando diferentes valores manuales del parámetro $h$ . . . . .	57
16	RMSE de entrenamiento de las plataformas T1-ANFIS, IT2-ANFIS M-KMA e IT2-ANFIS para el Ejemplo 5, considerando diferentes valores manuales del parámetro $h$ . . . . .	59
17	RMSE de entrenamiento de las plataformas T1-ANFIS, IT2-ANFIS M-KMA e IT2-ANFIS para el Ejemplo 6, considerando diferentes valores manuales del parámetro $h$ . . . . .	60
18	RMSE de entrenamiento de las plataformas T1-ANFIS, IT2-ANFIS M-KMA e IT2-ANFIS para el Ejemplo 7, considerando diferentes valores manuales del parámetro $h$ . . . . .	62
19	RMSE de entrenamiento de las plataformas T1-ANFIS, IT2-ANFIS M-KMA e IT2-ANFIS para el Ejemplo 8, considerando diferentes valores manuales del parámetro $h$ . . . . .	63
20	RMSE de entrenamiento de las plataformas T1-ANFIS, IT2-ANFIS M-KMA e IT2-ANFIS para el Ejemplo 9, considerando diferentes valores manuales del parámetro $h$ . . . . .	64
21	RMSE de entrenamiento de las plataformas T1-ANFIS, IT2-ANFIS M-KMA e IT2-ANFIS para el Ejemplo 10, considerando diferentes valores manuales del parámetro $h$ . . . . .	66
22	Comparación de estructuras ANFIS y algoritmos de optimización utilizados. . . . .	68

23	Resultados RMSE de los modelos T1-ANFIS vs IT2-ANFIS para el Ejemplo 11. . . . .	69
24	Resultados RMSE de los modelos T1-ANFIS vs IT2-ANFIS para el Ejemplo 12. . . . .	70
25	Resultados RMSE de los modelos T1-ANFIS vs IT2-ANFIS para el Ejemplo 13. . . . .	72
26	Parámetros premisa y consecuentes del modelo obtenido con IT2-ANFIS OS-TRA para el Ejemplo 13. . . . .	74
27	Comparación del RMSE y tiempo de cálculo de los TRA del Ejemplo 13. . .	75
28	Resultados RMSE del modelado con T1-ANFIS vs IT2-ANFIS para los datos de la Figura 51. . . . .	80
29	Comparativa de RMSE del error lateral y de cabeceo de diferentes controladores para el seguimiento de una línea recta. . . . .	81
30	Comparativa de RMSE del error lateral y de cabeceo de diferentes controladores para el seguimiento de tres surcos con radio de 2 m. . . . .	82
31	Comparativa de RMSE del error lateral y de cabeceo de diferentes controladores para el seguimiento de tres surcos con radio de 3 m. . . . .	86
32	Comparativa de RMSE del error lateral y de cabeceo de diferentes controladores para el seguimiento de una lemniscata. . . . .	87
33	Parámetros óptimos del IT2FLC empleado en la experimentación . . . . .	94
34	Resultados de la prueba LMI para estabilidad global variando la velocidad $v$ .	97
35	RMSE del error lateral y de cabeceo en la Ruta 1 bajo condiciones de ruido .	108
36	RMSE del error lateral y de cabeceo en la Ruta 2 bajo condiciones de ruido .	108
37	RMSE del error lateral y de cabeceo en la Ruta 3 bajo condiciones de ruido .	109
38	RMSE del error lateral y de cabeceo en la Ruta 4 bajo condiciones de ruido .	109
39	Comparativa de microcontroladores utilizados en el TR. . . . .	126
40	IDs asignados a los diferentes módulos del TR. . . . .	131

## Índice de Figuras

1	Funciones de membresía más habituales: a) gaussiana, b) triangular, c) campana generalizada, d) sigmoideal y e) trapezoidal. . . . .	21
2	Estructura de un T1-ANFIS en particular con dos entradas y una salida. . . .	22
3	Parámetros premisa y consecuentes de un T1-ANFIS particular. . . . .	24
4	FOU y funciones UMF y LMF de una GBSMF de IT2 con representación (a) 3D y (b) 2D, con un parámetro $h$ igual a 0.5. . . . .	26
5	Efectos de variar los parámetros de una GBSMF de IT2: (i) cambiando $a$ ; (ii) cambiando $b$ ; (iii) cambiando $c$ ; (iv) cambiando $h$ . . . . .	27
6	Estructura general de un T2FLS. . . . .	28
7	Estructura de un IT2-ANFIS particular con dos entradas y una salida. . . . .	32
8	Estructura de retroalimentación de un T1FLC con una sola variable de entrada (error). . . . .	35
9	Modelo cinemático de bicicleta para un vehículo con dirección Ackermann. . .	37
10	Comparación de métodos de suavizado de trayectorias. . . . .	38
11	Efecto de la distancia de anticipación en el seguimiento de trayectorias. . . .	39
12	Geometría del PPA para el seguimiento de trayectorias. . . . .	40
13	Aproximación de los modelos de las plataformas T1-ANFIS a la salida real del Ejemplo 1. . . . .	47
14	Error $(y_r - y_m)$ de los modelos de las plataformas T1-ANFIS para el Ejemplo 1.	49
15	Aproximación de los modelos de las plataformas T1-ANFIS a la salida real del Ejemplo 2. . . . .	49
16	Error $(y_r - y_m)$ de los modelos de las plataformas T1-ANFIS para el Ejemplo 2.	51
17	Aproximación de los modelos de las plataformas T1-ANFIS a la salida real del Ejemplo 3. . . . .	52
18	Error $(y_r - y_m)$ de los modelos de las plataformas T1-ANFIS para el Ejemplo 3.	53
19	FOU utilizadas para la comparación computacional entre los algoritmos OSTRAS, EIASC, KM y EKM. (a) Gaussiana con media incierta, (b) Gaussiana con desviación típica incierta y (c) Gaussiana con media incierta (lado derecho).	54
20	Aproximación obtenida con las plataformas T1-ANFIS MatLab, IT2-ANFIS M-KMA e IT2-ANFIS OSTRAS a la salida real del Ejemplo 4. . . . .	57
21	Error $(y_r - y_m)$ de las plataformas T1-ANFIS MatLab, IT2-ANFIS M-KMA e IT2-ANFIS OSTRAS para el Ejemplo 4. . . . .	58
22	Aproximación obtenida con las plataformas T1-ANFIS MatLab, IT2-ANFIS M-KMA e IT2-ANFIS OSTRAS a la salida real del Ejemplo 5. . . . .	59
23	Error $(y_r - y_m)$ de las plataformas T1-ANFIS MatLab, IT2-ANFIS M-KMA e IT2-ANFIS OSTRAS para el Ejemplo 5. . . . .	60
24	Aproximación obtenida con T1-ANFIS MatLab, IT2-ANFIS M-KMA e IT2-ANFIS OSTRAS para el Ejemplo 6, en comparación con la referencia. . . . .	61
25	Error de entrenamiento de T1-ANFIS MatLab, IT2-ANFIS M-KMA e IT2-ANFIS OSTRAS para el Ejemplo 6. . . . .	61

26	Aproximación obtenida con T1-ANFIS MatLab, IT2-ANFIS M-KMA e IT2-ANFIS OSTRÁ para el Ejemplo 7, en comparación con la referencia. . . . .	62
27	Error de entrenamiento de T1-ANFIS MatLab, IT2-ANFIS M-KMA e IT2-ANFIS OSTRÁ para el Ejemplo 7. . . . .	63
28	Aproximación obtenida con T1-ANFIS MatLab, IT2-ANFIS M-KMA e IT2-ANFIS OSTRÁ para el Ejemplo 8, en comparación con la referencia. . . . .	64
29	Error de entrenamiento de T1-ANFIS MatLab, IT2-ANFIS M-KMA e IT2-ANFIS OSTRÁ para el Ejemplo 8. . . . .	65
30	Aproximación obtenida con T1-ANFIS MatLab, IT2-ANFIS M-KMA e IT2-ANFIS OSTRÁ para el Ejemplo 9, en comparación con la referencia. . . . .	65
31	Error de entrenamiento de T1-ANFIS MatLab, IT2-ANFIS M-KMA e IT2-ANFIS OSTRÁ para el Ejemplo 9. . . . .	66
32	Aproximación obtenida con T1-ANFIS MatLab, IT2-ANFIS M-KMA e IT2-ANFIS OSTRÁ para el Ejemplo 10, en comparación con la referencia. . . . .	67
33	Error de entrenamiento de T1-ANFIS MatLab, IT2-ANFIS M-KMA e IT2-ANFIS OSTRÁ para el Ejemplo 10. . . . .	67
34	Aproximación del modelo T1-ANFIS vs IT2-ANFIS OSTRÁ a la salida real del Ejemplo 11. . . . .	69
35	Error ( $y_r - y_m$ ) del modelo T1-ANFIS vs IT2-ANFIS OSTRÁ para el Ejemplo 11. . . . .	69
36	Aproximación del modelo T1-ANFIS vs IT2-ANFIS OSTRÁ a la salida real del Ejemplo 12. . . . .	70
37	Error ( $y_r - y_m$ ) del modelo T1-ANFIS vs IT2-ANFIS OSTRÁ para el Ejemplo 12. . . . .	71
38	Estructura del sistema de volante del TR, con 2 entradas y 1 salida. . . . .	72
39	Señal de entrada $u(k)$ para entrenamiento. . . . .	72
40	Aproximación del modelo T1-ANFIS vs IT2-ANFIS OSTRÁ a la salida real del Ejemplo 13. . . . .	73
41	Error ( $y_r - y_m$ ) del modelo T1-ANFIS vs IT2-ANFIS OSTRÁ para el Ejemplo 13. . . . .	73
42	Señal de entrada $u(k)$ para validación. . . . .	74
43	Prueba No.1 el TR en un terreno plano, resultados del modelo frente al sistema real con RMSE=0.3084. . . . .	74
44	Prueba No.2 el TR en un camino de tierra, resultados del modelo frente al sistema real con RMSE=0.3088. . . . .	75
45	Señal de entrada $u(k)$ con ruido blanco gaussiano. . . . .	76
46	Aproximación de IT2-ANFIS OSTRÁ a la señal de salida del Ejemplo 13 con ruido blanco gaussiano en $u(k)$ . . . . .	76
47	Seguimiento de trayectorias en línea recta utilizando el GLC; (a) con distancia de anticipación de 0.5 m, (b) de 2 m y (c) de 3 m. . . . .	77
48	Seguimiento de trayectorias en tres surcos con radio 2 m; (a) utilizando el GLC con distancia de anticipación de 2 m y (b) utilizando el controlador BOE con distancia de anticipación de 2 m. . . . .	77

49	Variables cinemáticas registradas durante una trayectoria manual para el entrenamiento de los modelos difusos. (a) Trayectoria con identificadores de muestra $k$ para el inicio y fin de las curvas. (b) Ángulo de dirección del volante $\delta(k)$ . (c) Velocidad longitudinal del vehículo $v(k)$ . (d) Orientación del TR $\theta(k)$ . Las regiones sombreadas establecen la correspondencia exacta con las curvas delimitadas en (a). . . . .	78
50	Estructura de los modelos difusos y prueba de validación; (a) estructura propuesta para los IT2FLS utilizados para predecir $x(k+1)$ e $y(k+1)$ a partir de las variables de entrada y (b) comparación entre la ruta deseada y la ruta obtenida con el modelo propuesto durante la prueba de validación. . . . .	80
51	Conjunto de datos del seguimiento de trayectoria mediante el GLC para el modelado con ANFIS. (a) Referencia y trayectoria con identificadores de muestra $k$ para el inicio y fin de las curvas. (b) Señal de control del ángulo del volante $\delta(k)$ . (c) Evolución temporal del error de cabeceo $e_\theta$ . (d) Evolución de la derivada del error de cabeceo $\dot{e}_\theta$ . Las regiones sombreadas establecen la correspondencia exacta con las curvas delimitadas en (a). . . . .	81
52	Comparación del desempeño de diferentes controladores para el seguimiento de una línea recta. . . . .	82
53	Desempeño del seguimiento de trayectorias en una línea recta: comparativa entre el controlador T1FLC (columna izquierda) y el IT2FLC (columna derecha). (a) y (f) Seguimiento de la ruta de referencia en el plano Este-Norte; (b) y (g) Evolución de la orientación $\theta(k)$ ; (c) y (h) Evolución del error lateral $e_l(k)$ ; (d) y (i) Evolución del error de cabeceo $e_\theta(k)$ ; (e) y (j) Señal de control de dirección (posición del volante) $\delta(k)$ aplicada al actuador. . . . .	83
54	Comparación del desempeño de diferentes controladores para el seguimiento de tres surcos con radio de 2 m. . . . .	84
55	Desempeño del seguimiento de trayectorias en tres surcos con radio de 2 m: comparativa entre el controlador T1FLC (columna izquierda) y el IT2FLC (columna derecha). (a) y (f) Seguimiento de la ruta de referencia en el plano Este-Norte; (b) y (g) Evolución de la orientación $\theta(k)$ ; (c) y (h) Evolución del error lateral $e_l(k)$ ; (d) y (i) Evolución del error de cabeceo $e_\theta(k)$ ; (e) y (j) Señal de control de dirección (posición del volante) $\delta(k)$ aplicada al actuador. . . . .	85
56	Métodos para calcular el error lateral en el seguimiento de una trayectoria con curvas. . . . .	86
57	Comparación del desempeño de diferentes controladores para el seguimiento de tres surcos con radio de 3 m. . . . .	87
58	Desempeño del seguimiento de trayectorias en tres surcos con radio de 3m: comparativa entre el controlador T1FLC (columna izquierda) y el IT2FLC (columna derecha). (a) y (f) Seguimiento de la ruta de referencia en el plano Este-Norte; (b) y (g) Evolución de la orientación $\theta(k)$ ; (c) y (h) Evolución del error lateral $e_l(k)$ ; (d) y (i) Evolución del error de cabeceo $e_\theta(k)$ ; (e) y (j) Señal de control de dirección (posición del volante) $\delta(k)$ aplicada al actuador. . . . .	88

59	Comparación del desempeño de diferentes controladores para el seguimiento de una lemniscata. . . . .	89
60	Desempeño del seguimiento de trayectorias en una lemniscata: comparativa entre el controlador T1FLC (columna izquierda) y el IT2FLC (columna derecha). (a) y (f) Seguimiento de la ruta de referencia en el plano Este-Norte; (b) y (g) Evolución de la orientación $\theta(k)$ ; (c) y (h) Evolución del error lateral $e_l(k)$ ; (d) y (i) Evolución del error de cabeceo $e_\theta(k)$ ; (e) y (j) Señal de control de dirección (posición del volante) $\delta(k)$ aplicada al actuador. . . . .	90
61	Representación matemática en diagrama a bloques del sistema de control lateral en lazo cerrado. . . . .	91
62	Interfaz del simulador desarrollado para el TR, mostrando los componentes que la integran. . . . .	101
63	Configuración del sistema de seguimiento de trayectorias para un TR: integración de sensores y actuadores . . . . .	123
64	Configuración del microcontrolador LaunchPad TM4C1294 de la serie Tiva C. . . . .	124
65	Configuración de los microcontroladores Arduino Uno y Mega. . . . .	124
66	Configuración de los microcontroladores Arduino Uno y Mega. . . . .	125
67	Arquitectura modular física del sistema de control del TR. . . . .	127
68	Arquitectura física del sistema de control del TR . . . . .	130
69	Nube de puntos del RTK-GNSS durante la prueba de inmovilidad del TR . . . . .	138

# 1. Introducción

El diseño de controladores para sistemas dinámicos que operan en entornos físicos reales exige arquitecturas capaces de lidiar con altos niveles de incertidumbre, ruido en las mediciones y dinámicas no modeladas. En este contexto, los sistemas de lógica difusa (FLS, por sus siglas en inglés) han demostrado ser herramientas eficaces al incorporar conocimiento experto. Sin embargo, mientras que los sistemas de lógica difusa de tipo 1 (T1FLS, por sus siglas en inglés) poseen limitaciones frente a incertidumbres de segundo orden, los sistemas de lógica difusa de intervalo tipo 2 (IT2FLS, por sus siglas en inglés) ofrecen una mayor capacidad de manejo de la variabilidad gracias a su huella de incertidumbre (FOU, por sus siglas en inglés) [1, 2]. A pesar de esta superioridad teórica, la adopción práctica de los IT2FLS se ha visto frenada por tres factores principales: la ausencia de herramientas de modelado automático a partir de datos experimentales, el costo computacional de sus algoritmos de reducción de tipo (TRA, por sus siglas en inglés) y la necesidad de establecer garantías formales de estabilidad.

Esta tesis se enfoca en superar dichas barreras metodológicas y computacionales mediante el desarrollo de una arquitectura integral de modelado neuro-difuso. La propuesta central radica en la estructura algorítmica denominada IT2-ANFIS OSTRÁ, la cual genera de manera automática modelos difusos a partir de señales empíricas de entrada/salida (E/S). Para asegurar la viabilidad de su implementación en sistemas embebidos de tiempo real, la propuesta soluciona el problema computacional resolviendo la reducción de tipo de manera directa en un solo paso. Adicionalmente, para dotar al controlador de rigor analítico, se presenta una demostración formal de su estabilidad y convergencia global asintótica utilizando desigualdades matriciales lineales (LMI, por sus siglas en inglés).

Para validar la eficacia de esta metodología, las aportaciones teóricas se evalúan en un escenario de aplicación exigente: el control direccional de un TR experimental. En la práctica agrícola comercial, la orientación de estos vehículos depende frecuentemente de la fusión de sensores, combinando unidades RTK-GNSS con sistemas de medición inercial (IMU, por sus siglas en inglés) [3, 4]. No obstante, la literatura indica que las vibraciones continuas y los movimientos bruscos en terrenos irregulares introducen derivas en las mediciones inerciales, afectando la confiabilidad direccional [5, 6]. Frente a este desafío, el caso de estudio de esta investigación aborda el control lateral del vehículo prescindiendo totalmente de la IMU, utilizando en su lugar exclusivamente las coordenadas sucesivas del RTK-GNSS para alimentar el sistema. De esta forma, el algoritmo se somete a un escenario de ruido cinemático para evaluar experimentalmente el control difuso frente a otras técnicas de control.

El TR utilizado concentra de manera simultánea incertidumbres de medición, no linealidades cinemáticas y restricciones de hardware. Mediante experimentación bajo condiciones de oclusión satelital parcial (efecto multitrayecto), el controlador propuesto alcanzó un RMSE del error lateral de 0.094 m. Considerando que estudios de navegación agrícola recientes basados en esquemas que involucran tanto control lateral como longitudinal y emplean fusión sensorial con IMU reportan márgenes de error de entre 0.07 m y 0.12 m [7], el resultado

obtenido demuestra un desempeño competitivo frente al estado del arte. Este logro es destacable al operar con una estrategia exclusivamente lateral y prescindir de una IMU, lo cual confirma de forma empírica la viabilidad computacional y la solidez teórica del modelado con IT2-ANFIS OSTRÁ.

## 1.1. Planteamiento del problema

Los T1FLS son ampliamente utilizados para modelar y controlar sistemas con incertidumbre, gracias a su capacidad para capturar el conocimiento experto mediante funciones de membresía y reglas lingüísticas [8]. Sin embargo, sus funciones de membresía son deterministas: un valor de entrada produce un único grado de pertenencia, lo que limita su capacidad para representar la incertidumbre inherente a las mediciones ruidosas o a los parámetros del sistema que varían en el tiempo. Los T2FLS extienden esta representación mediante conjuntos difusos de tipo 2, cuya FOU permite modelar explícitamente la variabilidad en los propios grados de pertenencia [1], lo que permite su aplicación en sistemas con incertidumbre de segundo orden. No obstante, a diferencia de los T1FLS, para los cuales existen plataformas de modelado automático ampliamente disponibles como la herramienta ANFIS de MatLab [9], no existe una estructura equivalente para IT2FLS que permita generar modelos automáticamente a partir de datos de E/S con una calidad comparable. Esta ausencia representa un problema abierto que limita la adopción práctica de los IT2FLS en aplicaciones de control.

Un segundo problema abierto es el costo computacional de los TRA, etapa inevitable en cualquier sistema IT2FLS que convierte los conjuntos difusos de tipo 2 en valores accionables de tipo 1. Los algoritmos clásicos como el de Karnik-Mendel (KMA, por sus siglas en inglés) [10] y sus variantes mejoradas [11, 12] requieren un proceso iterativo de convergencia cuyo número de iteraciones es desconocido a priori en cada instancia de inferencia. Este comportamiento iterativo dificulta su implementación en sistemas embebidos de tiempo real con recursos computacionales limitados, representando una barrera práctica para el despliegue de controladores de lógica difusa de intervalo tipo 2 (IT2FLC, por sus siglas en inglés) en plataformas de bajo costo.

Un tercer problema emerge cuando se considera la síntesis de un controlador a partir de un modelo IT2FLS obtenido automáticamente: ¿es posible demostrar formalmente la estabilidad y convergencia de dicho controlador? La respuesta a esta pregunta es necesaria para que el modelo neuro-difuso generado a partir de datos experimentales trascienda el nivel de una herramienta heurística y adquiera respaldo teórico, condición indispensable en aplicaciones de control de sistemas físicos reales.

Por lo tanto, el problema central que aborda esta investigación puede formularse en tres niveles complementarios: (1) ¿Cómo desarrollar una estructura IT2-ANFIS que genere modelos IT2FLS de manera automática a partir de datos de E/S, con una precisión comparable o su-

perior al T1-ANFIS? (2) ¿Cómo reducir el costo computacional del proceso de reducción de tipo sin sacrificar la representación de la FOU característica de los IT2FLS? (3) ¿Cómo demostrar la estabilidad del IT2FLC sintetizado a partir de dicho modelo? La validación experimental de esta cadena metodológica completa se realiza sobre el actuador de dirección de un TR experimental, plataforma que concentra de manera simultánea incertidumbres de medición, dinámicas no modeladas y restricciones de cómputo en tiempo real, constituyendo un banco de pruebas representativo y exigente.

## 1.2. Justificación

Existe una correspondencia directa con los tres problemas identificados en el planteamiento. En primer lugar, la ausencia de una plataforma de modelado automático para IT2FLS equivalente a T1-ANFIS no es un problema menor: implica que el diseño de IT2FLC en la práctica continúa dependiendo de la sintonización manual de parámetros o de estructuras ad hoc con escasa reproducibilidad. Esto limita la transferencia tecnológica de los IT2FLS desde la investigación hacia aplicaciones de ingeniería reales, a pesar de que su ventaja teórica sobre los T1FLS en el manejo de la incertidumbre está ampliamente documentada [2, 13]. Desarrollar una estructura IT2-ANFIS que opere de forma automática a partir de datos de E/S responde, por tanto, a una necesidad concreta y tangible del área.

En segundo lugar, la viabilidad práctica de los IT2FLC en sistemas embebidos depende directamente de la eficiencia computacional del proceso de reducción de tipo. Los costos iterativos de los algoritmos clásicos no son solo un inconveniente académico: se traducen en retardos de cómputo que comprometen la respuesta en tiempo real de un controlador embebido, lo que puede conducir a inestabilidad o pérdida de precisión en sistemas físicos dinámicos. Un algoritmo de reducción de tipo no iterativo, capaz de completar el mapeo en un solo paso con precisión comparable, elimina esta barrera y abre la viabilidad de despliegue en plataformas de bajo costo, contribuyendo al objetivo más amplio de democratizar el acceso a controladores IT2FLS.

En tercer lugar, la generación automática de un modelo difuso a partir de datos experimentales, sin garantías formales de estabilidad, produciría únicamente una herramienta de ingeniería heurística. La demostración de que la ley de control interpolada por el IT2FLS es global y asintóticamente estable, mediante la formulación de LMI derivadas del modelo cinemático de Ackermann, eleva la propuesta al nivel de una contribución con fundamento teórico verificable, lo cual es un requisito de rigor científico ineludible en una investigación doctoral.

El TR experimental constituye el contexto de aplicación que motivó la identificación de estos tres problemas y la plataforma sobre la cual se valida experimentalmente la cadena metodológica completa, en condiciones reales de ruido de medición, dinámicas no modeladas y restricciones de hardware. Esta combinación de rigor teórico y validación empírica en un sistema físico real confiere a los resultados un nivel de solidez que trasciende el caso de estudio particular.

### 1.3. Antecedentes

Los FLS se han consolidado como una herramienta de uso generalizado para modelar sistemas complejos que implican incertidumbre e imprecisión [8]. Los T1FLS se han utilizado ampliamente en aplicaciones de modelización de sistemas sociales, económicos y dinámicos, entre otros, logrando aproximaciones funcionales a los sistemas [14, 15, 16]. Para incluir sistemas con mayor complejidad e incertidumbre, T2FLS [1] pueden utilizarse como una estrategia que modifica los resultados de la modelización con respecto a los sistemas T1FLS. Los T2FLS proporcionan una representación más detallada de la incertidumbre, ya que la modelización se realiza utilizando conjuntos difusos con una mayor capacidad para manejar la incertidumbre que los conjuntos difusos de tipo 1 (T1FS, por sus siglas en inglés) [17].

La limitación central de los T1FLS radica en que sus funciones de membresía son deterministas: dado un valor de entrada, el grado de pertenencia a cada conjunto difuso es un número fijo. Esta propiedad es aplicable cuando la incertidumbre es baja y las mediciones son confiables; sin embargo, en presencia de ruido de medición, parámetros variables en el tiempo o desacuerdo entre expertos al definir las reglas lingüísticas, los T1FLS limitan la representación de la variabilidad inherente a los propios grados de pertenencia [18]. Es decir, los T1FLS modelan la incertidumbre del sistema, pero no la incertidumbre sobre la propia descripción del sistema. Los IT2FLS extienden esta capacidad mediante una región de incertidumbre que engloba un intervalo de funciones de membresía posibles, ofreciendo una representación de la incertidumbre difusa, como se detalla en las secciones siguientes.

Un problema general de T2FLS es el alto costo computacional debido a un proceso de optimización iterativo para calcular los parámetros antecedentes y consecuentes del modelo final y el cálculo analítico implicado. Además, un IT2FLS delimita una región operativa formada por dos funciones de pertenencia, superior e inferior, constituyendo la FOU; mientras que el T1FLS sólo tiene en cuenta una línea determinista. Como consecuencia, en un T2FLS clásico se debe realizar iterativamente un proceso matemático adicional. En este sentido, los TRA funcionan como una etapa fundamental, ya que se encargan de mapear la salida inferida representada por conjuntos difusos de tipo 2 (T2FS, por sus siglas en inglés) voluminosos debido a su FOU hacia un T1FS unidimensional, posibilitando así su eventual defuzzificación hacia un valor accionable [19]. Existen diversos TRA [19], cuya elección depende netamente del compromiso entre precisión analítica y tiempo de convergencia iterativa.

En la herramienta MatLab Type-2 Fuzzy Inference Systems, se dispone de cuatro TRA: KMA [10], algoritmo de Karnik-Mendel mejorado (EKMA, por sus siglas en inglés) [11], algoritmo iterativo con condición de parada (IASC, por sus siglas en inglés) [20] y el algoritmo iterativo con condición de parada mejorado (EIASC, por sus siglas en inglés) [12]. El KMA es el método más común para calcular el centroide de un T2FS mediante la búsqueda iterativa de puntos de conmutación óptimos. Un punto de conmutación reside en el dominio donde la función de membresía (MF, por sus siglas en inglés) del T2FS cambia de un valor a otro. A continuación, este algoritmo itera hasta encontrar el centroide del T2FS. En contraste, en

la arquitectura propuesta en esta tesis, se introduce un nuevo algoritmo de reducción de tipo que elimina la necesidad de iterar para encontrar los puntos de conmutación. En su lugar, el algoritmo OSTRAL logra el mapeo en un solo paso implementando una abstracción analítica que calcula el promedio algebraico directo entre los límites superior e inferior de la FOU, lo que facilita y acelera sustancialmente el cálculo de los parámetros consecuentes mediante algoritmos de optimización.

La optimización del T2FLS sigue siendo una cuestión discutible, ya que muchas de las decisiones, como la elección de las funciones de pertenencia, el número de reglas, los operadores y los algoritmos de defuzificación, se siguen tomando de forma heurística. Sin embargo, pueden obtenerse mejoras mediante el aprendizaje automático, los algoritmos evolutivos y las técnicas de optimización bioinspiradas [2]. El sistema adaptable de inferencia neuro-difuso (ANFIS, por sus siglas en inglés) [21], es uno de los métodos más populares para el modelado de FLS, se basa en una combinación de redes neuronales y sistemas de lógica difusa para optimizar automáticamente los parámetros antecedentes y consecuentes, basados en las E/S del sistema a modelar. En [22] se mencionan hasta 52 algoritmos diferentes para la optimización de parámetros con T1-ANFIS, además de muchos de ellos con aplicaciones reales.

Entre los algoritmos más populares y precisos de optimización de parámetros de T1FLS se encuentran los siguientes: Algoritmo de Retropropagación (BP, por sus siglas en inglés) [23], Optimización por Enjambre de Partículas [24], Algoritmos Evolutivos [25] y Algoritmo de Optimización basado en Biogeografía [26]. En este trabajo se implementa BP para la optimización de los parámetros premisa del IT2-ANFIS, así como el parámetro que determina la amplitud de la FOU. El filtro de Kalman (KF, por sus siglas en inglés) se aplica para la optimización de los parámetros consecuentes de T1 Takagi-Sugeno (T-S) [27].

En [28] se propone un IT2-ANFIS y se denomina IT2 NS-1 ANFIS, haciendo referencia a que los parámetros antecedentes son de intervalo tipo 2 (IT2), posee fuzificación no puntual (donde los datos de entrada se modelan a sí mismos como conjuntos difusos para atenuar intrínsecamente el ruido de medición, en contraste a la fuzificación puntual clásica de valores numéricos nítidos) y sus reglas poseen un consecuente del tipo T-S. En [29] se presenta de igual forma un IT2-ANFIS con fuzificación no puntual y consecuentes lineales T-S, donde los algoritmos de optimización utilizados son, para los parámetros antecedentes el algoritmo de descenso de gradiente (GD, por sus siglas en inglés) y Levenberg-Marquardt, y para los consecuentes, mínimos cuadrados recursivos. En [30] se describen las ecuaciones de dos estructuras IT2-ANFIS denominadas ANFIS2. La primera utiliza conjuntos difusos de intervalo tipo 2 (IT2FS, por sus siglas en inglés) del tipo gaussianas en la parte antecedente y T1FS igual del tipo gaussianas en la parte consecuente. La segunda se basa en consecuentes lineales T-S. Ambas estructuras ANFIS2 utilizan el algoritmo de BP basado en la tasa de aprendizaje adaptable con convergencia garantizada para la optimización de parámetros. También, en [31] se utiliza la estructura ANFIS2 para modelar un brazo robótico con articulaciones flexibles y se prueba en un esquema de control, obteniendo una mejor respuesta con el modelo ANFIS2 frente a un T1-ANFIS.

Autores como Chen en [32, 33, 34], modifican la estructura tradicional T1-ANFIS y se prueba para entrenar IT2FLS. Sin embargo, afirman que los resultados no presentan una diferencia de magnitud relevante en comparación con los resultados de T1-ANFIS. En [35], se utiliza un IT2-ANFIS con un TRA que llaman M-KMA, en dicho trabajo sólo los parámetros del consecuente se optimizan para mostrar los efectos del TRA, mientras que los parámetros del antecedente permanecen sin cambios durante el entrenamiento en cada una de las estructuras ANFIS. Afirman que los resultados de IT2-ANFIS son iguales o presentan menor error que los de T1-ANFIS.

A partir de la revisión de literatura, se identifica que los métodos T2-ANFIS limitan la optimización a los parámetros consecuentes para evitar el costo computacional de la reducción de tipo iterativa [35], o bien reportan un desempeño de salida similar a los modelos de T1 [32]. Para solucionar estas restricciones, este trabajo unifica e implementa la arquitectura IT2-ANFIS OSTRÁ. A diferencia de las redes que mantienen antecedentes estáticos en el entrenamiento, el modelo propuesto vincula los métodos de aprendizaje (BP y KF) con una reducción de tipo procesada en un solo paso analítico. Esta integración permite optimizar de forma simultánea los parámetros que definen la función de membresía, la amplitud en la región de la FOU y las reglas del consecuente.

Para cuantificar operativamente esta arquitectura, se estructuró una validación del sistema en dos etapas. En un primer momento analítico y en condiciones aisladas, se evaluó y comparó el tiempo de respuesta de OSTRÁ contra los métodos iterativos estandarizados KMA, EKMA y EIASC. La segunda etapa evaluó la reducción del RMSE de entrenamiento de la red IT2-ANFIS OSTRÁ frente a los resultados de una arquitectura IT2-ANFIS M-KMA y las capas tradicionales de T1-ANFIS. La comparación se fundamentó utilizando trece conjuntos de datos de entrenamiento de diversa naturaleza, los cuales incluyeron funciones continuas, sistemas de segundo grado con varianza dinámica y registros experimentales. Para evaluar el desempeño de las arquitecturas ante perturbaciones, algunos de estos conjuntos fueron alterados intencionalmente induciendo ruido blanco gaussiano y emulando pérdidas de señal en los sensores.

El algoritmo diseñado genera IT2FLS con parámetros antecedentes y consecuentes óptimos, donde en los parámetros consecuentes mantienen la estructura T-S tradicional. Los IT2FLS obtenidos pueden ser utilizados para el diseño de controladores, estrategias de diagnóstico de fallos y predicción, entre otras aplicaciones. Esta estructura IT2-ANFIS OSTRÁ propuesta en este trabajo de tesis, será la responsable de modelar la dinámica lateral del TR, utilizando directamente como variables de entrada el error de cabeceo y su derivada, para estimar como variable de salida el ángulo del volante de dirección. De manera complementaria, para demostrar la flexibilidad del algoritmo en otras dinámicas del vehículo, esta estructura también se aplicó al modelado del control longitudinal (velocidad) en un entorno de simulación. Con ello, se obtiene un modelo difuso capaz de representar fielmente el comportamiento operativo de dichos subsistemas, contribuyendo así a los objetivos delineados en este trabajo.

Con la evolución constante y la implementación progresiva de la tecnología de conducción automática en maquinaria agrícola basada en sistemas de navegación por satélite, se ha presentado un avance en los niveles de automatización y la eficiencia operativa [36]. En el contexto agrícola moderno, la adopción de tecnologías como los tractores robot ha emergido como un catalizador para la eficiencia y la productividad. Estas innovaciones representan un avance en la automatización de tareas agrícolas, proporcionando beneficios en términos de tiempo, recursos y rendimiento. La conducción automática de tractores ha dejado de ser una visión futurista para convertirse en una realidad con el potencial de transformar la agricultura [37]. Empresas líderes en el comercio de tractores como John Deere [38], Yanmar [39], Case IH [40], entre otras, poseen ya avances y desarrollos en este campo.

En la comunidad de investigadores, existen diferentes aplicaciones de técnicas y metodologías que dotan a un tractor de la capacidad de navegar de manera automática. En [41], se desarrolló y validó un sistema de automatización de tractores capaz de realizar seguimiento de rutas, detectar obstáculos y determinar la necesidad de intervención humana, probando su viabilidad en condiciones reales de operación. Otro avance en la navegación automática de tractores, es el desarrollo de un sistema de guiado con un escáner láser para la navegación de un tractor agrícola entre hileras de árboles con aplicación en tiempo real [42]. En otra investigación, se modificó un tractor de tipo oruga comercial para convertirlo en un TR capaz de completar de forma automática trayectorias en línea recta con un error lateral de 0.01 a 0.03 m [43].

En el trabajo realizado en [44], derivado de una revisión exhaustiva de la literatura, identificaron que los sensores avanzados, los algoritmos inteligentes de alto rendimiento y el hardware de control eléctrico confiable son factores clave para promover el desarrollo de tractores robot confiables. En [45], se utiliza una configuración de dos módulos de sistema de posicionamiento global (GPS, por sus siglas en inglés) junto con tres sensores inerciales de bajo costo, para lograr la navegación a través de un viñedo con un error lateral de 0.2 m y  $0.2^\circ$  de error de cabeceo. En otro estudio, se desarrolló un esquema de control híbrido que combina controladores clásicos con redes neuronales difusas de tipo 2, logrando una navegación precisa y estable del tractor robot [46].

En esta tesis se propone una metodología de modelado difuso con el objetivo de generar automáticamente controladores IT2FLC a partir de datos de E/S del sistema real, capaces de gestionar las incertidumbres inherentes a la cinemática del vehículo. En este trabajo, dichas incertidumbres no se refieren a irregularidades topológicas del terreno (como baches o rocas), sino a perturbaciones presentes durante el desplazamiento, tales como el deslizamiento lateral de los neumáticos, las fluctuaciones de la señal satelital por oclusión transversal y las dinámicas no modeladas en el acoplamiento mecánico de la dirección. Para dotar de sustento analítico al enfoque basado en datos del modelado, se lleva a cabo una prueba de estabilidad y convergencia de la ley de control generada mediante la formulación de LMI derivadas del modelo cinemático de Ackermann, demostrando la estabilidad asintótica global del sistema

en lazo cerrado. Los controladores resultantes están orientados a minimizar el error lateral y de cabeceo durante el seguimiento de una trayectoria establecida, garantizando la precisión direccional del vehículo. Asimismo, se aborda el problema del sesgo en las unidades de medición inercial mediante el cálculo de la orientación a partir de las coordenadas posicionales del TR, presentando los resultados experimentales que se discuten a lo largo de este documento.

## 1.4. Hipótesis

La estructura propuesta IT2-ANFIS OSTRa es capaz de modelar automáticamente la dinámica lateral de un TR con actuador sobre el volante a partir de señales de entrada/salida, generando un IT2FLC cuya estabilidad en lazo cerrado se demuestra formalmente mediante LMIs. El IT2FLC resultante iguala o supera el desempeño de un controlador de lógica difusa de tipo 1 (T1FLC, por sus siglas en inglés) y el GLC en el seguimiento de trayectoria.

## 1.5. Objetivos

Los objetivos de este trabajo de investigación son los siguientes:

### 1.5.1. Objetivo general

Analizar y validar el modelado y control de un sistema TR a través de herramientas de FLS y de un control diseñado a partir de información extraída del modelo matemático, con el fin de ejecutar trayectorias agrícolas de forma precisa en áreas de trabajo.

### 1.5.2. Objetivos específicos

- a) Diseñar de forma automática un modelo difuso que represente el comportamiento del sistema TR, tomando en cuenta las variables de E/S que influyen directamente en la navegación: error de cabeceo ( $e_\theta$ ), su derivada ( $\dot{e}_\theta$ ) y el ángulo del volante ( $\delta$ ).
- b) Integrar el modelo difuso basado en la extracción de características del modelo matemático, como controlador de lógica difusa (FLC, por sus siglas en inglés) asegurando una representación coherente de la dinámica del sistema TR.
- c) Evaluar el desempeño del controlador en el TR para el seguimiento de trayectorias, analizando su capacidad para completar la ruta definida y adaptarse a las condiciones.
- d) Documentar los resultados obtenidos durante las pruebas, presentando conclusiones claras sobre la validez del modelado y control del TR para el seguimiento de trayectorias.

## 1.6. Alcances y delimitaciones

El enfoque de este trabajo de investigación se centra en el diseño e implementación de un sistema de control lateral para el TR. El objetivo principal es lograr el seguimiento de trayectorias mediante la regulación del ángulo de la dirección, mitigando las desviaciones causadas por las incertidumbres inherentes del TR mediante técnicas de lógica difusa.

En consecuencia, el control de la dinámica longitudinal del tractor, que involucra la modulación activa de la aceleración, el sistema de frenado y los cambios de marcha de la transmisión, no forma parte del alcance de esta tesis. Para los fines de las pruebas experimentales, estos actuadores se operaron a parámetros constantes. Esta decisión metodológica fue necesaria para establecer un entorno de experimentación controlado y evaluar de manera aislada el desempeño del controlador modelado utilizando la herramienta IT2-ANFIS OSTRÁ desarrollada en este trabajo de tesis. Asimismo, las tareas de esquivar obstáculos, la planificación de trayectorias dinámicas y las maniobras en reversa se consideran trabajos futuros ligados a la navegación autónoma y no se abordan en el presente documento.

## 1.7. Aportaciones

Las aportaciones como fruto de este trabajo de investigación se dividen en dos categorías:

a) Ciencias computacionales.

La presente investigación contribuye al estado del arte mediante el desarrollo de la plataforma IT2-ANFIS OSTRÁ, una arquitectura que integra un mecanismo directo de reducción de tipo. Esta aportación técnica elimina la naturaleza iterativa de los algoritmos clásicos de defuzificación, posibilitando una reducción en el costo computacional y en los tiempos de entrenamiento, lo cual resulta crítico para su viabilidad en aplicaciones de tiempo real. La plataforma permite la optimización automática de los parámetros del sistema, incluyendo la variable  $h$  encargada de ajustar la altura de la FOU.

Desde una perspectiva metodológica, se estableció un esquema de modelado automático para sistemas dinámicos, validado experimentalmente en un TR. Esta metodología facilita la generación de modelos IT2FLS capaces de asimilar dinámicas no lineales a partir de datos de E/S, logrando métricas de precisión que superan a los modelos correspondientes de T1.

En el ámbito de la aplicación práctica, se generó e implementó un IT2FLC derivado íntegramente del modelado automático. Este controlador fue evaluado en el TR bajo condiciones de perturbación externa y ruido de instrumentación. Las pruebas comparativas demostraron que el controlador propuesto obtiene un menor RMSE en el error lateral frente a técnicas clásicas (T1FLC) y controladores geométricos (GLC).

b) Publicación.

Como resultado directo de esta investigación, se ha generado producción científica que valida los hallazgos propuestos. Alberto-Rodriguez, A., López-Morales, V., & Ramos-Fernández, J. C. (2025). Novel Interval Type-2 ANFIS Modeling Based on One-Step Type Reducer Algorithm. *International Journal of Fuzzy Systems*, 27(1), 13-26.

## **1.8. Estructura de la tesis**

El presente trabajo se organiza en cinco capítulos distribuidos de la siguiente manera:

El Capítulo 1 presenta la introducción, que incluye el planteamiento del problema, la justificación, los antecedentes, la hipótesis, los objetivos, así como los alcances, delimitaciones y aportaciones del trabajo. Además, se proporciona un resumen de los resultados reportados en publicaciones científicas derivadas de la tesis. El Capítulo 2 describe el marco teórico y conceptual, abordando los fundamentos necesarios para comprender la metodología propuesta. El Capítulo 3 detalla la metodología empleada para el desarrollo del sistema de modelado y control. El Capítulo 4 presenta los resultados obtenidos y su discusión, evaluando el desempeño de la propuesta. El Capítulo 5 ofrece las conclusiones finales de la investigación y las perspectivas futuras. Finalmente, se incluyen las referencias bibliográficas y los apéndices correspondientes.

## 2. Marco Teórico

### 2.1. Sistemas de lógica difusa

#### 2.1.1. Modelo Mamdani

Este modelo lingüístico difuso propuesto en [47] como una forma de capturar el conocimiento semi-cualitativo disponible en forma de reglas SI-ENTONCES:

$$R_i: \text{SI } \tilde{x}_j \text{ está en } A_{j,k}. \text{ ENTONCES } \tilde{y}_i \text{ está en } B_{i,k}, \quad (1)$$

donde  $i = 1, 2, \dots, N$  es la  $i$ -ésima regla,  $j = 1, 2, \dots, J$  es la  $j$ -ésima entrada y  $k = 1, 2, \dots, K$  es la  $k$ -ésima MF de cada entrada.

Aquí  $\tilde{x}$  es la variable lingüística de entrada (premisa) y  $A_{i,k}$  son los términos lingüísticos de la premisa. De manera similar,  $\tilde{y}$  es la variable lingüística de salida (consecuente) y  $B_{i,k}$  son los términos lingüísticos del consecuente. Los valores de  $\tilde{x}$  e  $\tilde{y}$  y los términos lingüísticos  $A_{i,k}$ ,  $B_{i,k}$  son conjuntos difusos (FS, por sus siglas en inglés) definidos en los dominios de sus respectivas variables base:  $x \in X \subset \mathbb{R}^p$  e  $y \in Y \subset \mathbb{R}^q$ . Los grados de pertenencia de los FS de las premisas y consecuentes son entonces los mapeos:  $\mu(x) : X \rightarrow [0, 1]$ ,  $\mu(y) : Y \rightarrow [0, 1]$ . Los FS  $A_{i,k}$  definen regiones difusas en el espacio de las premisas, para las cuales se cumplen las proposiciones de los consecuentes respectivos. Los términos lingüísticos  $A_{i,k}$  y  $B_{i,k}$  generalmente se seleccionan de conjuntos de términos predefinidos, como *Pequeño*, *Mediano*, etc. Al denotar estos conjuntos por  $A$  y  $B$  respectivamente, se tiene  $A_{i,k} \subset A$  y  $B_{i,k} \subset B$ . La base de reglas  $R = \{R_i | i = 1, 2, \dots, N\}$  y los conjuntos  $A$  y  $B$  constituyen la base de conocimiento del modelo lingüístico.

#### 2.1.2. Modelo Takagi-Sugeno

El modelo lingüístico, introducido en la sección anterior, describe un sistema dado mediante reglas lingüísticas SI-ENTONCES con proposición difusa tanto en la premisa como en el consecuente. El modelo difuso de T-S [48] por otro lado, utiliza funciones numéricas en los consecuentes. Por tanto, puede verse como una combinación de modelos de regresión lingüística y matemática en el sentido de que las premisas describen regiones difusas en el espacio de entrada en las que las funciones consecuentes son válidas. Las reglas de T-S tienen la siguiente forma:

$$R_i: \text{SI } x_j \text{ está en } A_{j,k}. \text{ ENTONCES } y_i = f_i(x). \quad (2)$$

Las funciones  $f_i$  tienen la misma estructura para cada una de las reglas, variando solo los parámetros. Generalmente,  $f_i$  es una función con valores vectoriales, pero de igual forma puede ser un valor escalar nombrada como forma constante. Una parametrización simple y prácticamente útil es la forma lineal de los parámetros, que produce las reglas:

$$R_i: \text{SI } x_j \text{ está en } A_{j,k}. \text{ ENTONCES } y_i = C_{i,j}x_j + C_{i,J+1}, \quad (3)$$

donde  $C_{i,j}$  es un vector de parámetros consecuentes de dimensión igual a  $J$ . Este modelo se denomina modelo T-S. Este tipo de modelos, permite construir sistemas más precisos, robustos y aptos para ser optimizados automáticamente, razón por la cual es ampliamente utilizado en control automático, identificación de sistemas dinámicos y modelado de procesos no lineales. Su estructura también se adapta naturalmente a esquemas como ANFIS, donde los parámetros de las premisas y consecuentes se ajustan a partir de datos de entrenamiento. Por estas razones, en esta tesis se ha adoptado una estructura T-S en los consecuentes de los modelos difusos tanto T1 como T2, al permitir una inferencia más eficiente y compatible con algoritmos de optimización como BP o KF.

Tabla 2: Comparación del modelo Mamdani con el modelo T-S

<b>Característica</b>	<b>Mamdani</b>	<b>Takagi-Sugeno (T-S)</b>
Tipo de consecuente	Etiquetas lingüísticas	Funciones matemáticas
Facilidad de implementación	Intuitivo y fácil de interpretar.	Más preciso y computacionalmente eficiente.
Disponibilidad del modelo	Ideal para sistemas donde no se tiene un modelo matemático exacto y se depende de la experiencia humana.	Ideal para cuando sí se cuenta con un modelo matemático o datos de entrada/salida para identificarlo.
Adecuado para	Razonamiento heurístico y sistemas basados en reglas de conocimiento experto.	Control adaptativo, modelado automático y análisis analítico de estabilidad.
Uso en ANFIS	No estándar	Integrado (estructura T-S estándar)

## 2.2. Conjuntos difusos y funciones de membresía tipo 1

En la lógica difusa, a diferencia de la lógica clásica binaria, la transición de un estado a otro es gradual. El grado de pertenencia de un elemento a un conjunto está definido por la MF o función característica, la cual se asocia a un FS  $A$  y se define como  $\mu_A(x)$ . Para cada valor que tome la variable de entrada  $x$  en el universo de discurso  $X$ , la MF proporciona el grado de pertenencia al conjunto difuso. El FS  $A$  es definido formalmente como un par ordenado, como se muestra en la Ec. (4).

$$A = \{(x, \mu_A(x)), \forall x \in X\}. \quad (4)$$

Como se mencionó anteriormente, la MF proporciona una medida del grado de similitud de una variable de entrada en el universo de discurso  $X$  con el FS. La forma de la MF utilizada, depende del criterio aplicado por el experto, con la única condición de que ésta tome valores entre 0 y 1, con continuidad. Las funciones más comúnmente utilizadas por su simplicidad matemática y su manejabilidad son: triangular, trapezoidal, gaussiana, sigmoideal, gamma, pi y campana generalizada, etc. (Figura 1). Conceptualmente existen dos aproximaciones para determinar la MF asociada a un conjunto: la primera aproximación está basada en el conocimiento humano del experto, y la segunda es utilizar una colección de datos para diseñar la función. La segunda opción se discute en la Sección 2.3.

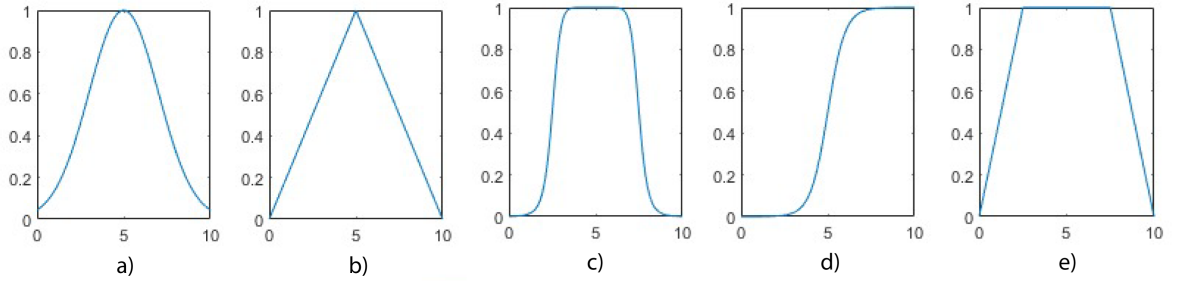


Figura 1: Funciones de membresía más habituales: a) gaussiana, b) triangular, c) campana generalizada, d) sigmoideal y e) trapezoidal.

En el presente trabajo de tesis se hace uso de GBSMF en los diferentes FLS. A continuación, se establece su definición.

$$\mu_{A_{j,k}}(x_j) = \mu_{A_{j,k}}(x_j) = f(x_j; a_{j,k}, b_{j,k}, c_{j,k}) = \frac{1}{1 + \left| \frac{x_j - c_{j,k}}{a_{j,k}} \right|^{2b_{j,k}}} \quad (5)$$

Para calcular el grado de pertenencia de una GBSMF, se utiliza la Ec. (5), donde  $a_{j,k}$  define la anchura de la MF. Un valor mayor crea una función de pertenencia más amplia. El parámetro  $b_{j,k}$  define la forma de la curva a cada lado de la meseta central, donde un valor mayor crea una transición más pronunciada, siendo  $c_{j,k}$  el centro de la GBSMF. El subíndice  $j = 1, 2, \dots, J$  es el número de entradas,  $k = 1, 2, \dots, K$  es el número de MF por cada entrada.

### 2.3. T1-ANFIS

La metodología de modelado llamada T1-ANFIS fue desarrollado por J. R. Jang en 1993 [21]. En esta metodología se combinan la capacidad de aprendizaje y la estructura relacional de las redes neuronales artificiales con el mecanismo de toma de decisiones de la lógica difusa. T1-ANFIS realiza el aprendizaje con conjuntos de datos E/S que se utilizan para el entrenamiento como en redes neuronales artificiales. De esta forma, se obtiene el T1FLS más idóneo para resolver el problema. La estructura obtenida posterior al entrenamiento se somete a un proceso de validación para ver su impacto en muestras de datos que nunca ha visto.

Obtener valores de error bajos en la prueba indican la efectividad del modelo T1-ANFIS. Una de las desventajas más importantes de las redes neuronales artificiales es que los valores de los pesos obtenidos no se pueden explicar. Esta desventaja se elimina mediante el sistema de inferencia difuso que se encuentra en la estructura de T1-ANFIS.

La estructura de T1-ANFIS consta de dos grupos de parámetros diferentes: premisas y consecuentes. Entrenar T1-ANFIS significa determinar estos parámetros mediante un algoritmo de optimización. En el modelo T1-ANFIS desarrollado por Jang [21], se propuso un enfoque de aprendizaje híbrido para la estimación de los parámetros. En este enfoque, mientras que los parámetros premisa se determinan mediante el GD, los parámetros consecuentes se encuentran con el método de estimación de mínimos cuadrados (LSE, por sus siglas en inglés). Desde que se desarrolló T1-ANFIS, se ha utilizado en el modelado e identificación de numerosos sistemas, logrando resultados exitosos [49, 50, 51, 52].

### 2.3.1. Estructura T1-ANFIS

La estructura de red de T1-ANFIS, como se mencionó anteriormente consta de dos partes denominadas premisas y consecuentes. Entrenar un T1-ANFIS significa determinar los parámetros pertenecientes a estas partes utilizando un algoritmo de optimización. T1-ANFIS utiliza datos de E/S existentes durante el entrenamiento obteniendo reglas difusas SI-ENTONCES. La estructura de una red T1-ANFIS consta de cinco capas, como se observa en la Figura 2. En esta figura en particular, la estructura tiene dos entradas y una salida, a su vez se compone de cuatro funciones de membresía y cuatro reglas difusas. A continuación se explica cada una de las capas de T1-ANFIS de acuerdo con la estructura que se muestra en la Figura 2.

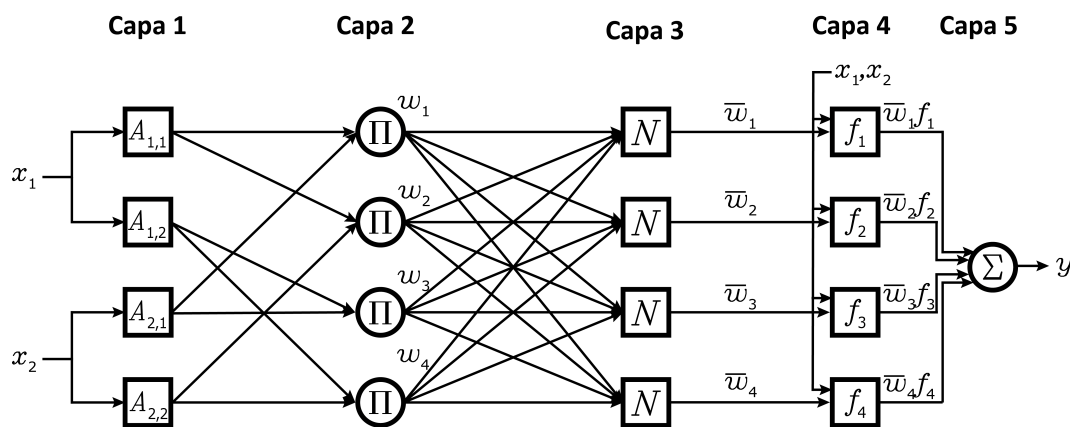


Figura 2: Estructura de un T1-ANFIS en particular con dos entradas y una salida.

#### Capa 1: capa de fuzificación

Cada nodo de esta capa es un fuzificador, se utilizan MF para obtener la pertenencia a un FS a partir de los valores de entrada. Los parámetros determinan la forma de las funciones de membresía y estos comúnmente se llaman parámetros premisa. Las variables  $a_{j,k}$ ,  $b_{j,k}$  y  $c_{j,k}$

son el conjunto de parámetros premisa de una GBSMF. Los grados de pertenencia de cada MF se calculan utilizando estos parámetros, como se indica en la Ec. (6). La forma correcta de interpretar los subíndices  $i, j, k$  es: el subíndice  $i = 1, 2, \dots, N$  representa el número de reglas;  $j$  y  $k$  como un bucle for desde  $j = 1$  hasta  $J$  entradas y anidar otro bucle for desde  $k = 1$  hasta  $K$  MF por cada entrada.

$$O_{(j-1)K+k}^1 = \mu_{A_{j,k}}(x_j) = f(x_j; a_{j,k}, b_{j,k}, c_{j,k}) = \frac{1}{1 + \left| \frac{x_j - c_{j,k}}{a_{j,k}} \right|^{2b_{j,k}}}. \quad (6)$$

### Capa 2: capa de reglas

Los niveles de activación  $w_i$  para las reglas se calculan utilizando los grados de pertenencia generados en la capa de fuzificación (Capa 1). Los valores de  $w_i$  se obtienen multiplicando estos grados de pertenencia de la siguiente manera:

$$O_i^2 = w_i = \mu_{A_{j,k}}(x_j) \mu_{A_{j+1,k}}(x_{j+1}) \dots \mu_{A_{J,k}}(x_J). \quad (7)$$

### Capa 3: capa de normalización

Aquí se calculan los niveles de activación normalizados que pertenecen a cada regla. El valor normalizado es la relación entre el nivel de activación de la regla  $i$ -ésima y la sumatoria de todos los niveles de activación como se indica en la Ec. (8).

$$O_i^3 = \tilde{w}_i = \frac{w_i}{\sum_{i=1}^N w_i}. \quad (8)$$

### Capa 4: capa de defuzificación

Los valores ponderados de las reglas se calculan en cada nodo de ésta capa como se indica en la Ec. (9). Este valor se determina utilizando un polinomio de primer orden,

$$O_i^4 = \tilde{w}_i f_i = \tilde{w}_i (C_{i,j} x_j + C_{i,j+1} x_{j+1} + \dots + C_{i,J} x_J + C_{i,J+1}), \quad (9)$$

donde  $\tilde{w}_i$  es la salida de la capa de normalización y  $C_{i,j}, C_{i,j+1}, \dots, C_{i,J}$  son los parámetros consecuentes y  $C_{i,J+1}$  es el parámetro independiente. El número de parámetros consecuentes de cada regla es uno más que el número de entradas.

### Capa 5: capa de sumatoria

La salida estimada por T1-ANFIS se obtiene sumando las salidas obtenidas para cada regla en la capa de defuzificación como en la Ec. (10).

$$O^5 = \sum_{i=1}^N O_i^4 = \sum_{i=1}^N \tilde{w}_i f_i, \quad (10)$$

donde  $N$  se calcula elevando el número de MF al número de entradas ( $N = K^J$ ).

### 2.3.2. Algoritmos de optimización para T1-ANFIS

Para utilizar T1-ANFIS, el entrenamiento es esencial; esto significa determinar los parámetros (premisa y consecuentes) en su estructura mediante un algoritmo de optimización. En la fase de entrenamiento, los parámetros premisa y consecuentes se utilizan como se muestra en la Figura 3. Desde el primer desarrollo de T1-ANFIS, se han sugerido tres diferentes enfoques de entrenamiento con este fin, estos son: algoritmos basados en derivadas, heurísticos e híbridos [22].

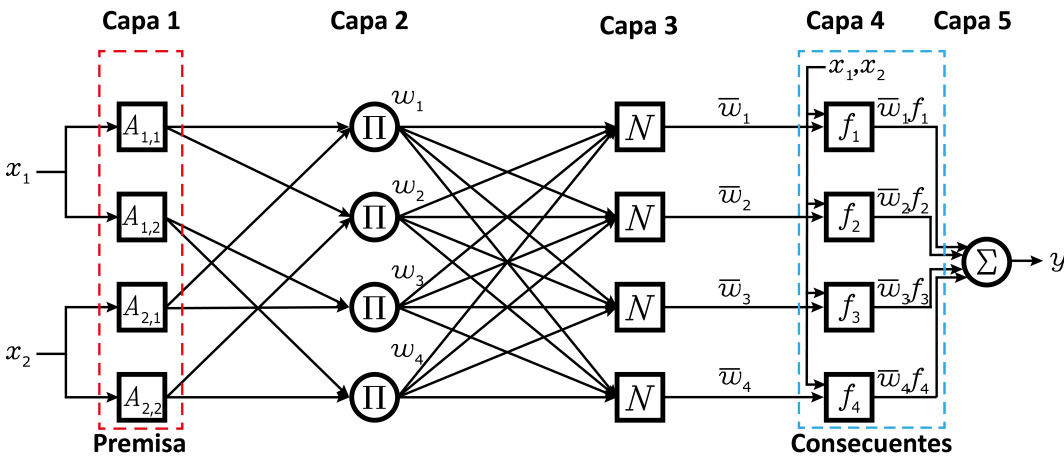


Figura 3: Parámetros premisa y consecuentes de un T1-ANFIS particular.

En la revisión bibliográfica realizada en [22], se establece que el algoritmo basado en derivadas GD/BP + LSE [21] generalmente proporciona un rendimiento mayor que otros enfoques basados en derivadas. Sin embargo, los algoritmos heurísticos de optimización por enjambre de partículas [53], genéticos [54] y colonia de abejas artificial [55], también han demostrado ser exitosos en el entrenamiento T1-ANFIS para resolver problemas del mundo real. En este trabajo de tesis se utilizan los algoritmos de BP para la optimización de los parámetros premisa y KF para los parámetros consecuentes.

## 2.4. Conjuntos difusos y funciones de membresía Tipo 2

En ocasiones, hay situaciones complejas en las que es extremadamente difícil encontrar el valor numérico preciso de la MF. La construcción de reglas utilizadas en FLS es una tarea difícil, debido a que el conocimiento requerido para este propósito es incierto. Por consiguiente, la formación de reglas difusas inadecuadas puede provocar incertidumbre en las funciones de pertenencia. Por lo tanto, independientemente de la utilidad de los T1FLS en ciertas aplicaciones, hay problemas en los que se desea lograr un nivel de precisión o rendimiento que con T1FLS no se obtiene [2]. Derivado de lo anterior, surge la necesidad de disponer de diferentes herramientas y técnicas que puedan modelar mayores niveles de incertidumbre.

La lógica difusa de T2 fue introducida por Zadeh en 1975 [56]. Durante los primeros años de su desarrollo, los investigadores se centraban en la lógica difusa de T1 y sus aplicaciones, pero a medida que enfrentaron problemas más desafiantes, el interés se desplazó hacia un enfoque que pudiera manejar situaciones más complejas. A menudo, el conocimiento que se utiliza para construir las reglas en un FLS es incierto. De acuerdo con [18] hay tres formas en las que pueden ocurrir incertidumbres en las reglas:

- Las palabras que se usan en las premisas y consecuentes de las reglas pueden significar cosas diferentes para diferentes personas.
- Los consecuentes obtenidos al sondear a un grupo de expertos a menudo serán diferentes para la misma regla, porque los expertos no necesariamente estarán de acuerdo.
- Para el entrenamiento en ocasiones se dispone de datos ruidosos. Esto se traduce en que las funciones de pertenencia de las premisas o consecuentes sean inciertas.

El abordaje de estas fuentes de incertidumbre, y en particular el modelado de datos de entrenamiento ruidosos señalado en el tercer punto, constituye una motivación teórica central para el desarrollo propuesto en esta tesis. Como se expone en los capítulos metodológicos, el diseño de la arquitectura IT2-ANFIS OSTRÁ se fundamenta en la capacidad de los T2FLS para modelar estas variabilidades inherentes de manera automática a partir de evidencia empírica, mitigando simultáneamente el alto costo computacional de reducción de tipo asociado a su implementación analítica.

En ciertos contextos es posible identificar incertidumbres asociadas al lenguaje natural, a su vez dicha incertidumbre puede ser ponderada con grados de pertenencia difusos, considerándolos de esta manera como incertidumbre difusa de T2. En general, un T2FS es una generalización de un T1FS asociado a una fuente secundaria de incertidumbre relacionada con la definición de un conjunto adicional  $\tilde{A}$ . Dicha fuente adicional de incertidumbre se representa como una MF secundaria asociada a cada valor del universo de discurso  $x \in X$ , en otras palabras:

$$\tilde{A} : X \rightarrow F([0, 1]), \quad (11)$$

donde  $F([0, 1])$  es la unión de todos los T1FS que pueden ser definidos dentro del universo de discurso  $x \in X$ . Un T2FS  $\tilde{A}$  se puede definir mediante una tupla de la siguiente manera:

$$\tilde{A} = \{(x, u), \mu_{\tilde{A}}(x, u) \mid \forall x \in X, \forall u \in J_x \subseteq [0, 1], \mu_{\tilde{A}}(x, u) \in [0, 1]\}, \quad (12)$$

donde,  $X$  es el dominio de la variable difusa, y  $u$  pertenece al intervalo llamado membresía primaria, es decir,  $u \in J_x \subseteq [0, 1]$ ,  $\mu_{\tilde{A}}(x, u)$  es una MF bidimensional, con valores  $0 \leq \mu_{\tilde{A}}(x, u) \leq 1$  para definir la pertenencia secundaria [1]. Si todos los  $\mu_{\tilde{A}}(x, u) = 1$ , entonces se denomina IT2FS. Los T2FS, cuya MF secundaria es un intervalo (es decir que la incertidumbre asociada al lenguaje es tratada de manera equitativa), son llamados IT2FS, ya que los grados de pertenencia asignados a los elementos del universo de discurso son un intervalo. A continuación en la Sección 2.4.1, se describen los IT2FS.

### 2.4.1. Conjuntos Difusos de Intervalo Tipo 2

Un IT2FS es expresado por dos funciones de pertenencia, donde una representa el grado de pertenencia en  $X$  y la otra da una ponderación a cada uno de los T1FS [8]. Así, un IT2FS es definido como:

$$\tilde{A} = \int_{x \in X} \int_{u \in J_x} \mu_{\tilde{A}}(x, u) / (x, u), J_x \subseteq [0, 1] = \int_{x \in X} \int_{u \in J_x} 1 / (x, u), \quad (13)$$

donde  $x$  es la variable principal,  $J_x$  tiene un intervalo entre  $[0, 1]$  y es la MF principal de  $x$ . La variable secundaria  $u$  y  $\int_{u \in J_x} 1/u$  es la MF secundaria en  $x$ . Teniendo en cuenta que la Ec. (13) significa que:  $\tilde{A} : X \rightarrow \{[a, b] : 0 \leq a \leq b \leq 1\}$ . La incertidumbre sobre  $\tilde{A}$  es transmitida por la unión de todas las membresías principales llamada FOU, de  $\tilde{A}$  [ $FOU(\tilde{A})$ ], es decir.

$$FOU(\tilde{A}) = \bigcup_{x \in X} J_x. \quad (14)$$

La FOU se muestra en la Figura 4 (región sombreada), la cual se encuentra delimitada por una MF superior (UMF, por sus siglas en inglés)  $\mu_{\tilde{A}}(x)$  y una MF inferior (LMF, por sus siglas en inglés)  $\mu_{\underline{A}}(x)$ , ambos son T1FS. En consecuencia, el grado de pertenencia de cada elemento de un IT2FS es un intervalo  $[\mu_{\tilde{A}}(x), \mu_{\underline{A}}(x)]$ . Así mismo, un IT2FS también se puede representar como se muestra en la Ec. (15).

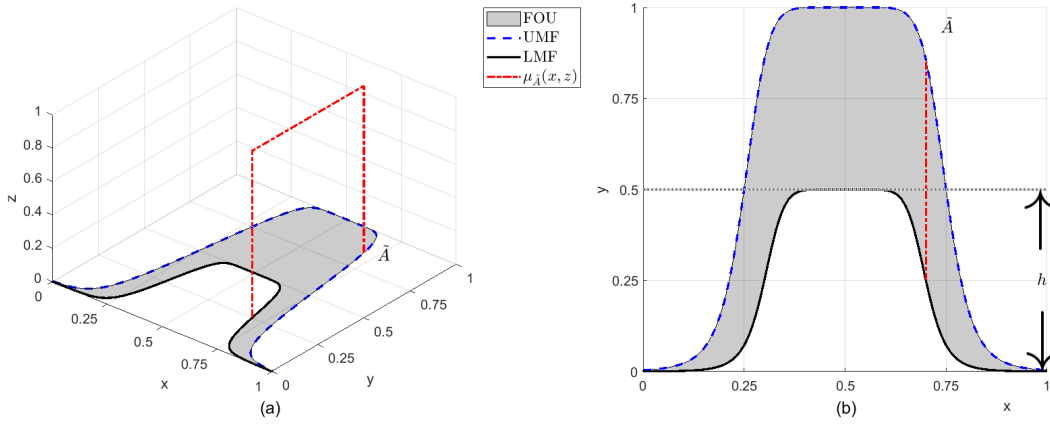


Figura 4: FOU y funciones UMF y LMF de una GBSMF de IT2 con representación (a) 3D y (b) 2D, con un parámetro  $h$  igual a 0.5.

$$\tilde{A} = 1/FOU(\tilde{A}) = 1/[\mu_{\tilde{A}}(x), \mu_{\underline{A}}(x)]. \quad (15)$$

Las UMF y LMF para el caso de una GBSMF se obtiene de igual forma que para los T1FS, esto recordando que un IT2FS es un intervalo entre dos T1FS, en las Ec. (16) y (17) se muestra la forma de calcular  $\mu_{\tilde{A}_{j,k}}(x_j)$  y  $\mu_{\underline{A}_{j,k}}(x_j)$  respectivamente.

$$\mu_{\tilde{A}_{j,k}}(x_j) = f(x_j; \bar{a}_{j,k}, b_{j,k}, c_{j,k}) = \frac{1}{1 + \left| \frac{x_j - c_{j,k}}{\bar{a}_{j,k}} \right| 2b_{j,k}}, \quad (16)$$

$$\mu_{\underline{A}_{j,k}}(x_j) = f(x_j; \underline{a}_{j,k}, b_{j,k}, c_{j,k}, h_{j,k}) = \frac{h_{j,k}}{1 + \left| \frac{x_j - c_{j,k}}{\underline{a}_{j,k}} \right| 2b_{j,k}}, \quad (17)$$

donde  $\bar{a}_{j,k}$ ,  $\underline{a}_{j,k}$ ,  $b_{j,k}$ ,  $c_{j,k}$  y  $h_{j,k}$  son los parámetros de la función,  $\bar{a}_{j,k}$  debe ser mayor que  $\underline{a}_{j,k}$ , de lo contrario la UMF se encontraría por debajo de la LMF. El parámetro  $h_{j,k} \in [0, 1]$  es la medida de las incertidumbres; cuanto más cerca de 0 mayor es el FOU, y viceversa. Este parámetro sólo define el límite de la LMF y puede utilizarse para los distintos tipos de MF. Una característica importante propuesta en este trabajo de tesis es la capacidad de la estructura IT2-ANFIS propuesta, para optimizar todos los parámetros antecedentes y el parámetro  $h_{j,k}$  descrito en la Ec. (17) usando el algoritmo de BP, definiendo la forma de la FOU para cada IT2FS durante el entrenamiento. En la Figura 5, se puede observar cómo la variación de los diferentes parámetros que influyen en la forma de la GBSMF de IT2.

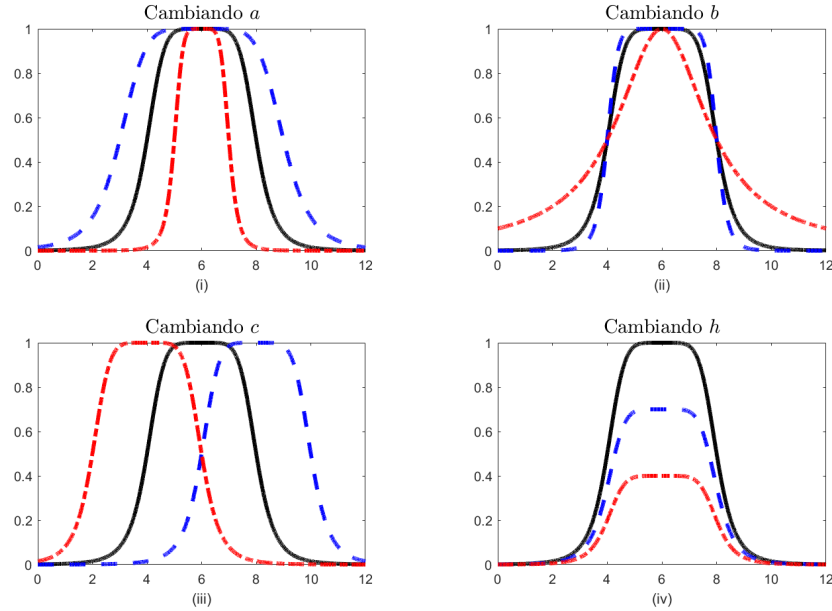


Figura 5: Efectos de variar los parámetros de una GBSMF de IT2: (i) cambiando  $a$ ; (ii) cambiando  $b$ ; (iii) cambiando  $c$ ; (iv) cambiando  $h$ .

La Tabla 3 muestra los diferentes parámetros para construir el GBSMF de la Figura 5 (i), (ii), (iii) y (iv), y cómo afecta el cambio de los parámetros de  $a, b, c$  y  $h$  a la forma de la MF.

#### 2.4.2. Algoritmos de reducción de tipo

Un TRA es utilizado para mapear la salida de un T2FLS (IT2FLS en ésta tesis) en un T1FS [57], que luego puede ser defuzificado a una salida nítida (Figura 6). Tal proceso se llama reducción de tipo y de acuerdo con [34] ha sido uno de los temas de investigación más activos

Tabla 3: Parámetros  $a, b, c$  y  $h$  de las distintas MF de la Figura 5.

Parámetro	Figura								
	(i,ii,iii,iv)---	(i)---	(i)---	(ii)---	(ii)---	(iii)---	(iii)---	(iv)---	(iv)---
$a$	2	1	3	2	2	2	2	2	2
$b$	3	3	3	5	1	3	3	3	3
$c$	6	6	6	6	6	4	8	6	6
$h$	1	1	1	1	1	1	1	0.4	0.7

para los T2FS. En la literatura, se han propuesto distintos TRA para IT2FS. Un TRA basado en centroides bien conocido y probablemente el más utilizado es el KMA [10], o su mejora el EKMA [11] que calcula los límites izquierdo y derecho de los centroides para todos los T1FS incrustados posibles. Existen también otros TRA como Nie-Tan [58], el IASC [20] y EIASC [12], entre otros. A continuación se describen los pasos de los algoritmos KMA y M-KMA:

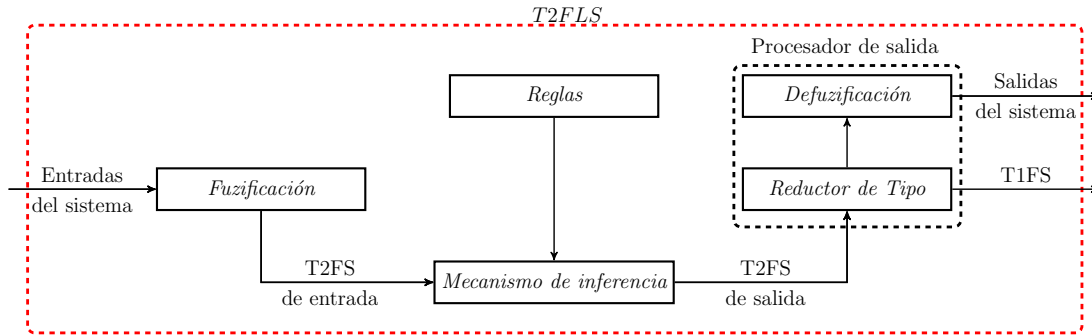


Figura 6: Estructura general de un T2FLS.

### Algoritmo de Karnik-Mendel

El KMA [10] se propuso para determinar los puntos de cambio  $l$  y  $r$  del centroide, y su costo computacional es alto dado que se trata de un método iterativo y además necesita de más de una iteración para converger. Para compensar el costo computacional del KMA, se propuso posteriormente el EKMA [11], el cual consiste en utilizar puntos óptimos de inicialización para el KMA que permitan reducir el número de iteraciones necesarias para convergencia, con un 97 % de probabilidad de converger en una iteración. A continuación se describe el KMA, donde  $\bar{w}_i$  y  $w_i$  son los pesos superior e inferior respectivamente (Ec. (18)),  $f_i$  son las salidas de la  $i$ -ésima regla (Ec. (19)).

$$\begin{aligned}
 \underline{w}_i &= \mu_{\underline{A}_{j,k}}(x_j) \mu_{\underline{A}_{j+1,k}}(x_{j+1}) \dots \mu_{\underline{A}_{J,k}}(x_J), \\
 \bar{w}_i &= \mu_{\bar{A}_{j,k}}(x_j) \mu_{\bar{A}_{j+1,k}}(x_{j+1}) \dots \mu_{\bar{A}_{J,k}}(x_J),
 \end{aligned} \tag{18}$$

$$f_i = C_{i,j}x_j + C_{i,j+1}x_{j+1} + \dots + C_{i,J}x_J + C_{i,J+1}, \quad (19)$$

1. Ordenar  $f_i$  en orden ascendente e igualar con los pesos  $\bar{w}_i$  y  $\underline{w}_i$ , obteniendo un nuevo índice  $n$ .
2. Inicializar  $w_n$  como  $w_n = \frac{w_n + \bar{w}_n}{2}$

**Calcular  $yl$**

3. Calcular:

$$yl = \frac{\sum_{n=1}^N f_n w_n}{\sum_{n=1}^N w_n}. \quad (20)$$

4. Encontrar el punto de cambio  $l \in [1, N - 1]$  para  $f_l \leq yl \leq f_{l+1}$ .
5. Calcular:

$$y' = \frac{\sum_{n=1}^l f_n \bar{w}_n + \sum_{n=l+1}^N f_n w_n}{\sum_{n=1}^l \bar{w}_n + \sum_{n=l+1}^N w_n}. \quad (21)$$

6. Si  $y' = yl$ , terminar; si no se cumple, definir  $yl = y'$  y repetir desde el Paso 4.

**Calcular  $yr$**

7. Calcular:

$$yr = \frac{\sum_{n=1}^N f_n w_n}{\sum_{n=1}^N w_n}. \quad (22)$$

8. Encontrar el punto de cambio  $r \in [1, N - 1]$  para  $f_r < yr \leq f_{r+1}$ .
9. Calcular:

$$y' = \frac{\sum_{n=1}^r f_n w_n + \sum_{n=r+1}^N f_n \bar{w}_n}{\sum_{n=1}^r w_n + \sum_{n=r+1}^N \bar{w}_n}. \quad (23)$$

10. Si  $y' = yr$ , terminar; si no se cumple, definir  $yr = y'$  y repetir desde el Paso 8.

11. Obtener la salida del IT2FLS de la siguiente manera:

$$y = \frac{yl + yr}{2} \quad (24)$$

### Algoritmo de Karnik-Mendel Modificado

El M-KMA es propuesto por M, Öztürk e I, Özkol en [35]. En dicho trabajo se modifica el KMA con la finalidad de obtener una matriz  $X$  que se utiliza para calcular los parámetros consecuentes mediante un algoritmo de optimización en un IT2-ANFIS. El algoritmo original da resultados para IT2FLS, pero no permite obtener las matriz  $X$  que se requieren para el entrenamiento de IT2-ANFIS, de ahí la modificación realizada que se describe a continuación:

1. Ordenar  $f_i$  en orden creciente e igualar con los pesos  $\bar{w}_i$  y  $\underline{w}_i$ , obteniendo un nuevo índice  $n$ .
2. Inicializar  $w_n$  como  $w_n = \frac{w_n + \bar{w}_n}{2}$

### Calcular $yl$

3. Calcular:

$$yl = \frac{\sum_{n=1}^N f_n w_n}{\sum_{n=1}^N w_n} \quad (25)$$

4. Encontrar el punto de cambio  $l$  para  $f_l < yl \leq f_{l+1}$ .
5. Calcular  $y' = X_l W$  donde:

$$X_l = \frac{[\bar{w}_1 x_1, \bar{w}_1 x_2, \bar{w}_1, \dots, \bar{w}_l x_1, \bar{w}_l x_2, \bar{w}_l, \dots, \underline{w}_{l+1} x_1, \underline{w}_{l+1} x_2, \underline{w}_{l+1}, \dots, \underline{w}_N x_1, \underline{w}_N x_2, \underline{w}_N]}{\sum_{n=1}^l \bar{w}_n + \sum_{n=l+1}^N \underline{w}_n}, \quad (26)$$

considere esta forma del vector para un caso de dos entradas, si el número de entradas es mayor aumentará la dimensión agregando la multiplicación de cada peso por cada entrada, por ejemplo:

$$\begin{aligned} & [\bar{w}_1 x_1, \bar{w}_1 x_2, \dots, \bar{w}_1 x_J, \bar{w}_1, \dots, \bar{w}_l x_1, \bar{w}_l x_2, \dots, \bar{w}_l x_J, \bar{w}_l, \dots \\ & \underline{w}_{l+1} x_1, \underline{w}_{l+1} x_2, \dots, \underline{w}_{l+1} x_J, \underline{w}_{l+1}, \dots, \underline{w}_N x_1, \underline{w}_N x_2, \dots, \underline{w}_N x_J, \underline{w}_N] \end{aligned} \quad (27)$$

6. Si  $y' = y_l$ , terminar; si no se cumple, definir  $y_l = y'$  y repetir desde el Paso 4.

**Calcular  $y_r$**

7. Calcular:

$$y_r = \frac{\sum_{n=1}^N f_n w_n}{\sum_{n=1}^N w_n}. \quad (28)$$

8. Encontrar el punto de cambio  $r$  para  $f_r < y_r \leq f_{r+1}$ .

9. Calcular  $y' = X_r W$  donde:

$$W_r = \frac{[\underline{w}_1 x_1, \underline{w}_1 x_2, \underline{w}_1, \dots, \underline{w}_r x_1, \underline{w}_r x_2, \underline{w}_r, \dots, \bar{w}_{r+1} x_1, \bar{w}_{r+1} x_2, \bar{w}_{r+1}, \dots, \bar{w}_n x_1, \bar{w}_n x_2, \bar{w}_n]}{\sum_{n=1}^r \underline{w}_n + \sum_{n=r+1}^N \bar{w}_n}. \quad (29)$$

10. Si  $y' = y_r$ , terminar; si no se cumple, definir  $y_r = y'$  y repetir desde el Paso 8.

**Calcular la salida  $y$ :**

$$y = \frac{y_l + y_r}{2} = \frac{X_l W + X_r W}{2} = \frac{X_l + X_r}{2} W = X W, \quad (30)$$

donde  $W$  es un vector columna con los parámetros consecuentes como se observa en la Ec. (31).

$$W = [C_{i,j} \ C_{i,j+1} \ \dots \ C_{i,J} \ C_{i,J+1} \ C_{i+1,j} \ C_{i+1,j+1} \ \dots \ C_{i+1,J} \ C_{i+1,J+1} \ \dots \ C_{N,j} \ C_{N,j+1} \ \dots \ C_{N,J}]. \quad (31)$$

## 2.5. T2-ANFIS

Los primeros estudios sobre T2-ANFIS se pueden encontrar en [59, 60], donde las entradas de la red son variables lingüísticas y las salidas son numéricas. En la literatura existen trabajos de investigación sobre T2-ANFIS como [61, 62, 63], donde las entradas de la red son numéricas así como las salidas. De acuerdo con [64] los métodos basados en mínimos cuadrados no pueden ser aplicados a un T2FLS, ya que la base de las funciones difusas para un T2FLS T-S, son el punto de partida para el método de LSE, por lo tanto no pueden obtenerse sin antes conocer los parámetros consecuentes.

Como se mencionó en la Sección 2.4.1, los IT2FLS tienen mayor capacidad para modelar sistemas inciertos en comparación con T1FLS. Sin embargo, la optimización de los parámetros de las premisas y consecuentes presenta mayor complejidad para este tipo de sistemas [65]. En la literatura existen métodos para determinar automáticamente los parámetros de un IT2FLS que se aplican para sistemas de una entrada, como la implementación del algoritmo de ajuste basado en el GD [66], un método de aprendizaje híbrido que utiliza LSE ortogonales recursivos para ajustar los parámetros consecuentes de T1 y BP para ajustar el intervalo de parámetros de las premisas de T2 [67]. Existen soluciones basadas en matemáticas puras como los controladores inversos IT2 [68, 69] que solo funcionan para una sola entrada. Mientras tanto T1-ANFIS proporciona una formación para entradas múltiples, existen trabajos como [32, 33, 34] donde se modifica la estructura T1-ANFIS y se prueba para entrenar IT2FLS.

### 2.5.1. Estructura IT2-ANFIS

Para brindar claridad, la estructura IT2-ANFIS propuesta en este trabajo de tesis (Figura 7), se muestra para  $J = 2$  entradas, una salida y  $K = 2$  MFs para cada entrada, con un número de reglas o nodos  $N = K^J = 4$ . La estructura IT2-ANFIS tiene 8 capas y utiliza OSTRa como TRA; se describen a continuación cada una de las capas:

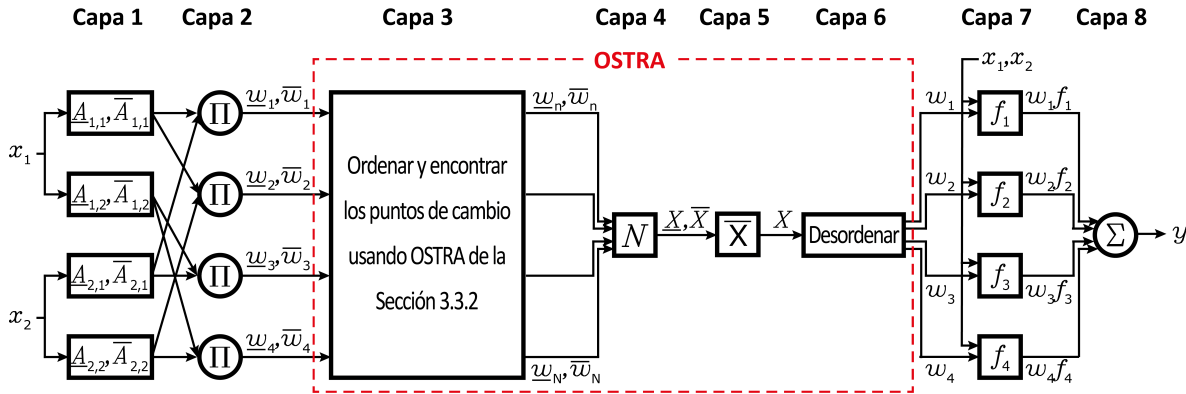


Figura 7: Estructura de un IT2-ANFIS particular con dos entradas y una salida.

**Capa 1:** De forma similar a T1-ANFIS, cada nodo de esta capa es un fuzzificador, por lo que la salida  $O_{(j-1)K+k}^1$  puede representarse como el intervalo de grados de pertenencia de dos T1FS  $\underline{A}_{j,k}$  y  $\bar{A}_{j,k}$ , que juntos representan un IT2FS. Así,  $a_{j,k}, \bar{a}_{j,k}, b_{j,k}, c_{j,k},$  y  $h_{j,k}$  son el conjunto de parámetros antecedentes de una GBSMF de IT2, donde  $(a_{j,k}, b_{j,k}, c_{j,k}, h_{j,k})$  pertenecen a la LMF y  $(\bar{a}_{j,k}, b_{j,k}, c_{j,k})$  a la UMF. Los grados de pertenencia de cada FS se calculan utilizando parámetros antecedentes, como se indica en las ecuaciones (16) y (17). La forma correcta de interpretar los índices  $j, k$  es como un bucle for desde  $j = 1$  hasta  $J$  y anidar otro bucle for desde  $k = 1$  hasta  $K$ .

$$O_{(j-1)K+k}^1 = \left[ \mu_{\underline{A}_{j,k}}(x_j), \mu_{\bar{A}_{j,k}}(x_j) \right]. \quad (32)$$

**Capa 2:** Se calcula el intervalo de los pesos inferior y superior  $[\underline{w}_i, \bar{w}_i]$  respectivamente. Los valores de  $\underline{w}_i$  y  $\bar{w}_i$  se obtienen multiplicando los grados de pertenencia obtenidos en la Capa 1 de la siguiente manera:

$$O_i^2 = [\underline{w}_i, \bar{w}_i] = \left[ \mu_{\underline{A}_{j,k}}(x_j) \mu_{\underline{A}_{j+1,k}}(x_{j+1}) \dots \mu_{\underline{A}_{J,k}}(x_J), \mu_{\bar{A}_{j,k}}(x_j) \mu_{\bar{A}_{j+1,k}}(x_{j+1}) \dots \mu_{\bar{A}_{J,k}}(x_J) \right]. \quad (33)$$

La interpretación el índice  $k$  de esta capa se debe tratar como una tabla de verdad pero con números que van ascendiendo. En la Tabla 4 se muestran tres ejemplos para distintos números de entradas y de MF, como se observa conforme el FS  $j = J$  llega a  $k = K$  aumenta en 1 el índice  $k$  de la  $j$  anterior y se repite el ciclo.

Tabla 4: Tabla de verdad para diferentes números de entradas y de MF.

$J = 2$ y $K = 2$	$J = 3$ y $K = 2$	$J = 2$ y $K = 3$
$A_{1,1} \quad A_{2,1}$	$A_{1,1} \quad A_{2,1} \quad A_{3,1}$	$A_{1,1} \quad A_{2,1}$
$A_{1,1} \quad A_{2,2}$	$A_{1,1} \quad A_{2,1} \quad A_{3,2}$	$A_{1,1} \quad A_{2,2}$
$A_{1,2} \quad A_{2,1}$	$A_{1,1} \quad A_{2,2} \quad A_{3,1}$	$A_{1,1} \quad A_{2,3}$
$A_{1,2} \quad A_{2,2}$	$A_{1,1} \quad A_{2,2} \quad A_{3,2}$	$A_{1,2} \quad A_{2,1}$
	$A_{1,2} \quad A_{2,1} \quad A_{3,1}$	$A_{1,2} \quad A_{2,2}$
	$A_{1,2} \quad A_{2,1} \quad A_{3,2}$	$A_{1,2} \quad A_{2,3}$
	$A_{1,2} \quad A_{2,2} \quad A_{3,1}$	$A_{1,2} \quad A_{2,1}$
	$A_{1,2} \quad A_{2,2} \quad A_{3,2}$	$A_{1,2} \quad A_{2,2}$
		$A_{1,2} \quad A_{2,3}$
		$A_{1,2} \quad A_{2,1}$
		$A_{1,2} \quad A_{2,2}$
		$A_{1,2} \quad A_{2,3}$
		$A_{1,3} \quad A_{2,1}$
		$A_{1,3} \quad A_{2,2}$
		$A_{1,3} \quad A_{2,3}$

**Capa 3:** En esta capa, los pesos obtenidos en la capa anterior se ordenan en orden ascendente para calcular los puntos de cambio  $l$  y  $r$  utilizando el OSTRa descrito en la Sección 3.2.1, hasta el paso número 3. Los pesos ordenados y los puntos de cambio calculados en el OSTRa se obtienen como salida de esta capa.

$$O_n^3 = [w_n, \bar{w}_n]. \quad (34)$$

**Capa 4:** Para calcular la salida de esta capa (capa de normalización), se utilizan los pesos ordenados y los puntos de cambio calculados en la Capa 3. El valor normalizado es la relación entre el peso de la  $i$ -ésima regla ordenada y el total de todos los pesos ordenados, generando dos vectores columna  $\underline{X}$  y  $\bar{X}$  como se muestra en la Ec. (35) y (36). El intervalo de estos dos vectores, forma la salida  $O^4$  como se muestra en la Ec. (37).

$$\underline{X} = \frac{[\bar{w}_n, \dots, \bar{w}_l, \underline{w}_{l+1}, \dots, \underline{w}_N]}{\sum_{n=1}^l \bar{w}_n + \sum_{n=l+1}^N \underline{w}_n}, \quad (35)$$

$$\bar{X} = \frac{[\underline{w}_n, \dots, \underline{w}_r, \bar{w}_{r+1}, \dots, \bar{w}_N]}{\sum_{n=1}^r \underline{w}_n + \sum_{n=r+1}^N \bar{w}_n}, \quad (36)$$

$$O^4 = [\underline{X}, \bar{X}]. \quad (37)$$

**Capa 5:** Sumar los vectores  $\underline{X}$  y  $\bar{X}$ , para obtener el promedio del intervalo como se representa en la Ec. (38) para generar los pesos normalizados.

$$O^5 = X = \frac{\underline{X} + \bar{X}}{2}. \quad (38)$$

**Capa 6:** Reordenar los pesos en la matriz  $X$  según el orden original en  $w_i$  antes de llegar a la Capa 3.

$$O^6 = Unsort(X) = [w_i, w_{i+1}, \dots, w_N]. \quad (39)$$

**Capa 7:** Se trata de una capa de defuzzificación. A través de un polinomio de primer orden como se observa en la Ec. (40), se calculan los valores ponderados de las reglas, utilizando los parámetros consecuentes, pesos y entradas.

$$O_i^7 = w_i f_i = w_i (C_{i,j} x_j + C_{i,j+1} x_{j+1} + \dots + C_{i,J} x_J + C_{i,J+1}). \quad (40)$$

**Capa 8:** Es la suma de las salidas de cada regla por el peso correspondiente, como se muestra a continuación:

$$O^8 = \sum_{i=1}^N O_i^7 = \sum_{i=1}^N w_i f_i. \quad (41)$$

## 2.6. Controladores de lógica difusa

Las primeras aplicaciones de los FLC fueron llevadas a cabo por Mamdani y Assilian en 1974 [70], implementando el controlador en una máquina de vapor de caldera pequeña especificando reglas de control difuso heurísticas. En años posteriores Kickert y Van Nauta Lemke [71], aplicaron controladores difusos a una planta de agua caliente, obteniendo menor error en comparación con el controlador proporcional, integral, derivativo (PID) clásico. Estos FLC son ampliamente utilizados para resolver problemas donde los sistemas no son lineales o es difícil modelar matemáticamente su comportamiento con precisión. En lugar de basarse en un modelo matemático preciso del sistema, los controladores de lógica difusa se basan en el conocimiento y la experiencia humana expresados en reglas de tipo SI-ENTONCES. Por este motivo, suelen ser robustos frente a cambios en las condiciones del sistema, ya que se basan en reglas heurísticas y no en un modelo matemático exacto.

En esencia, a nivel macroscópico de lazo cerrado, la topología de E/S de un FLC puede homologarse didácticamente a los conceptos de los controladores clásicos PID. Como se observa en la Figura 8, se muestra el ejemplo estructural de un T1FLC que, al recibir únicamente el error de trayectoria como entrada para dictaminar la señal de salida, emula funcionalmente la dinámica de un controlador de tipo Proporcional (P difuso). Si se buscara obtener el equivalente a un controlador proporcional derivativo (PD difuso), por ejemplo, se debe agregar la derivada del error como segunda entrada al FLC.

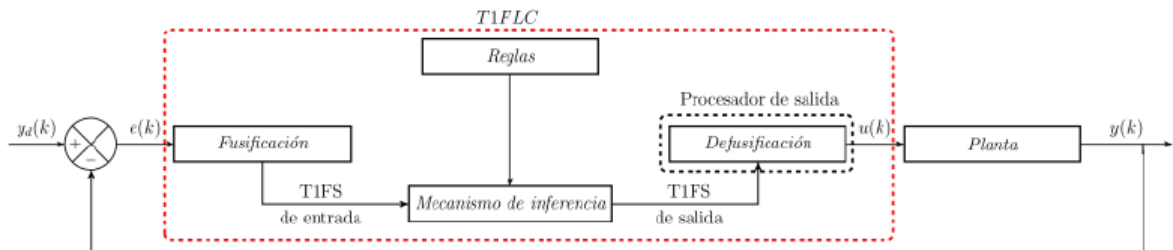


Figura 8: Estructura de retroalimentación de un T1FLC con una sola variable de entrada (error).

## 2.7. Modelo matemático Ackermann

En 1818, Rudolph Ackermann obtuvo la patente británica 4212 en nombre del inventor alemán George Lankensperger para un mecanismo de dirección que garantiza que las ruedas de dirección se muevan en círculos con un centro común. El mismo esquema fue propuesto y probado por Erasmus Darwin (abuelo de Charles) en la década de 1760. El refinamiento posterior realizado por el francés Charles Jeantaud condujo al mecanismo utilizado en los automóviles hasta el día de hoy y que se conoce como dirección Ackermann [72]. Este modelo, también conocido como el modelo cinemático de dirección Ackermann, es una representación matemática utilizada para describir cómo se mueve un vehículo con un sistema de

dirección convencional, como el que se encuentra en los tractores agrícolas y muchos robots terrestres.

El modelo es ampliamente usado en robótica móvil y en la teoría de control para simular y controlar vehículos de cuatro ruedas. El propósito del modelo es describir la geometría de la dirección para permitir que el vehículo gire de acuerdo a sus restricciones mecánicas. El modelo cinemático de Ackermann se utiliza para representar el movimiento del vehículo sin considerar las fuerzas dinámicas, aproximación válida cuando las velocidades de operación son bajas y los deslizamientos laterales se pueden ignorar. Las ecuaciones diferenciales que describen el movimiento del vehículo en el dominio del tiempo continuo son:

$$\begin{aligned}\dot{x}(t) &= v(t) \cos(\theta(t)), \\ \dot{y}(t) &= v(t) \sin(\theta(t)), \\ \dot{\theta}(t) &= \frac{v(t)}{L} \tan(\delta(t)),\end{aligned}\tag{42}$$

donde:

- $\dot{x}$  y  $\dot{y}$  representan las velocidades del vehículo en las direcciones  $x$  e  $y$  del marco inercial,
- $\dot{\theta}$  es la velocidad angular del vehículo (la tasa de cambio del ángulo de orientación o cabeceo),
- $v$  es la velocidad lineal del vehículo,
- $\delta$  es el ángulo de dirección de las ruedas delanteras,
- $L$  es la distancia entre los ejes delantero y trasero.

El modelo de dirección Ackermann suele ser simplificado usando el modelo de bicicleta, en el cual se asume que el vehículo tiene una sola rueda delantera y una rueda trasera, como se observa en la Figura 9. Esta simplificación facilita el análisis matemático del movimiento, sin perder la esencia de la dinámica de la dirección. El radio de giro  $R = L / \tan(\delta)$  es la distancia entre el centro del arco de la trayectoria y el eje longitudinal del vehículo. Como cabría esperar, el radio de giro aumenta en relación con la longitud del vehículo. El ángulo de dirección  $\delta$  suele estar limitado mecánicamente y su valor máximo determina el valor mínimo de  $R$ .

Para su implementación digital, el modelo cinemático se representa en tiempo discreto. Aplicando la derivada discreta de Euler, las ecuaciones Ec. (42) se transforman en las siguientes ecuaciones en diferencias:

$$\begin{aligned}x(k+1) &= x(k) + v(k) \cos(\theta(k))T_s, \\ y(k+1) &= y(k) + v(k) \sin(\theta(k))T_s, \\ \theta(k+1) &= \theta(k) + \frac{v(k)}{L} \tan(\delta(k))T_s,\end{aligned}\tag{43}$$

donde  $k$  y  $k+1$  indican el instante actual y el siguiente, respectivamente, y  $T_s$  es el tiempo de muestreo.

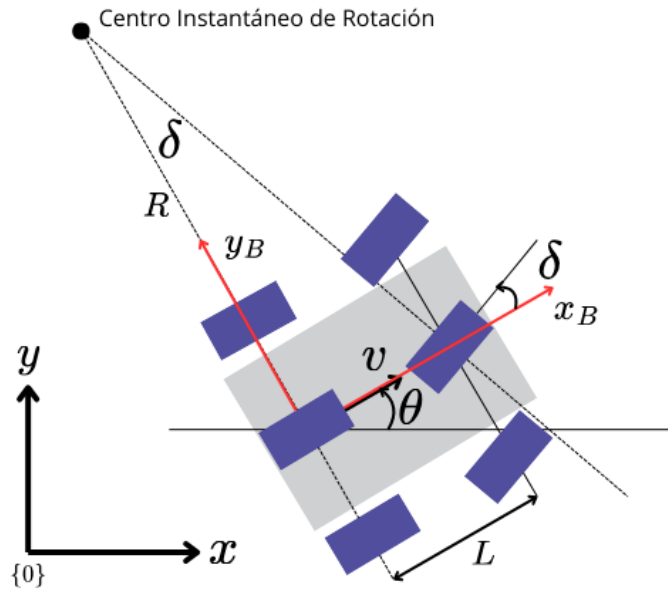


Figura 9: Modelo cinemático de bicicleta para un vehículo con dirección Ackermann.

## 2.8. Algoritmo de persecución pura

El algoritmo de persecución pura (PPA, por sus siglas en inglés) se ideó originalmente para calcular el arco necesario para que un robot retomara un camino. La primera aplicación del método se produjo con el Terragator, un robot de seis ruedas con dirección deslizante que se utilizó para experimentos de visión en exteriores a principios de los años 80 [73]. El trabajo con Terragator se trasladó a un robot llamado NavLab, dando paso al algoritmo de comando de arco. A lo largo del proyecto NavLab se propusieron y aplicaron varios algoritmos de seguimiento de trayectorias, incluido el enfoque del polinomio quintico y un enfoque de teoría del control. Las pruebas de estos algoritmos demostraron que el PPA era el más sólido y fiable [74].

El PPA funciona calculando la curvatura que moverá un vehículo desde su posición actual hasta un objetivo. El algoritmo consiste en elegir una posición de meta que se encuentre a cierta distancia por delante del vehículo en la trayectoria, esto se define como distancia de anticipación  $l_a$  y es un parámetro crítico para el rendimiento del algoritmo. El nombre de persecución pura procede de la analogía que utilizaron para describir el método. Se tiende a pensar que el vehículo persigue un punto del camino que se encuentra a cierta distancia por delante de él: persigue ese punto en movimiento. Esta analogía se utiliza a menudo para comparar este método con la forma de conducir de los humanos. Tendemos a mirar cierta distancia por delante del coche y nos dirigimos hacia ese punto. Esta distancia cambia a medida que conducimos para reflejar las curvas de la carretera y las oclusiones visuales [75].

El PPA es un método geométrico ampliamente utilizado para el seguimiento de trayectorias

en vehículos autónomos, tanto terrestres como móviles. Su objetivo principal es ajustar el ángulo de dirección del vehículo para seguir una ruta de referencia, minimizando el error lateral y asegurando que el vehículo siga la trayectoria de manera precisa y suave. La ruta de referencia es una parte clave entre el sistema de planificación y controlador, pudiéndose definir de varias formas. La manera más sencilla es utilizar una serie de segmentos de línea recta conectadas en varios puntos, como se muestra en la Figura 10 (a). Sin embargo, este enfoque tiene discontinuidades en el rumbo representando un desafío para los vehículos con dirección, que no pueden hacer cambios de sentido de manera abrupta.

El segundo enfoque aborda el problema anterior, utilizando una mayor cantidad de segmentos de línea que generalmente tienen el mismo espaciado como se observa en la Figura 10 (b). De esta forma, la posición de los puntos de referencia puede ajustarse para respetar una restricción aproximada de curvatura. Las rutas basadas en puntos de referencia son bastante comunes debido a que son fáciles de implementar y se pueden construir directamente a partir de estimaciones de estado o datos GPS obtenidos en recorridos previos de una ruta específica. El último enfoque consiste en utilizar secuencias de curvas parametrizadas continuas como se aprecia en la Figura 10 (c), estas curvas pueden ser generadas a partir de un conjunto determinado de primitivas de movimiento u optimizadas con base a una planificación de movimiento.

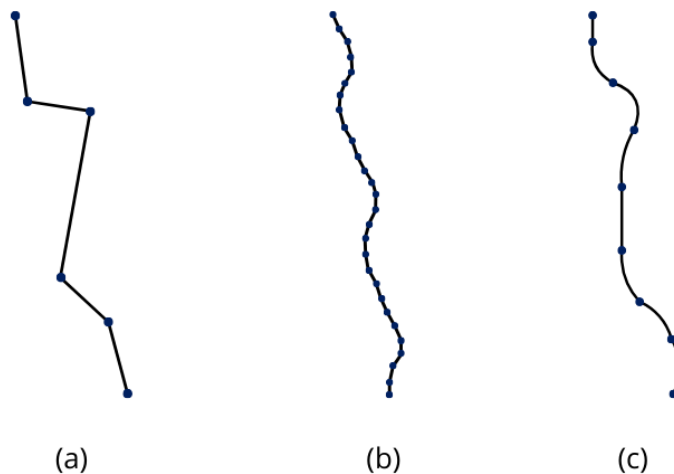


Figura 10: Comparación de métodos de suavizado de trayectorias.

Como se mencionó anteriormente, la distancia de anticipación  $l_a$  es fundamental para determinar cómo el vehículo ajusta su dirección para seguir la ruta de referencia. Este parámetro afecta directamente el rendimiento del seguimiento de varias maneras:

- Un valor grande de  $l_a$  indica que el punto de visión adelantada está más lejos, lo cual hace que el vehículo tome decisiones de dirección más graduales. Esto resulta en un movimiento más suave, pero puede generar problemas al tratar de seguir trayectorias muy curvas, ya que el vehículo tiende a “recortar” las curvas y no sigue exactamente la trayectoria (Figura 11) (a).

- Un valor pequeño de  $l_a$  significa que el punto de visión adelantada está muy cerca del vehículo, lo cual resulta en un seguimiento más preciso, pero puede hacer que el vehículo oscile o reaccione de manera brusca a los cambios en la trayectoria, especialmente si hay pequeñas imperfecciones o curvas pronunciadas (Figura 11 (b)).

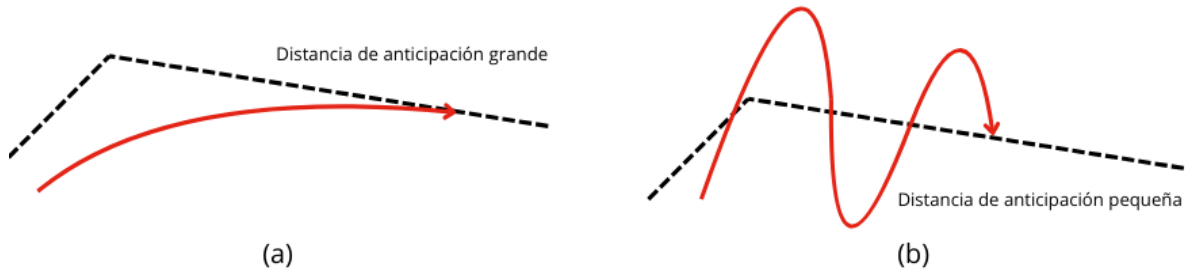


Figura 11: Efecto de la distancia de anticipación en el seguimiento de trayectorias.

El objetivo del PPA es minimizar el error lateral moviendo el vehículo hacia un punto adelantado en la ruta. La distancia a este punto siempre será mayor que la distancia de visión anticipada  $l_a$  y denotaremos a esta distancia entre el vehículo y el punto de referencia en la ruta como  $l_d$ . El objetivo es apuntar la dirección del vehículo hacia el punto objetivo, para asegurar que el vehículo esté sobre la trayectoria con mínimo error lateral. Con base en la Figura 9, y como se mencionó anteriormente el radio de giro se calcula como:  $R = L/\tan(\delta)$ . Analizando la Figura 12, podemos calcular  $R = l_d/(2\sin(\alpha))$ , entonces igualando ambas ecuaciones obtenemos:

$$\frac{L}{\tan(\delta)} = \frac{l_d}{2\sin(\alpha)}, \quad (44)$$

despejando  $\tan(\delta)$  esta ecuación se obtiene:

$$\tan(\delta) = \frac{2L\sin(\alpha)}{l_d}. \quad (45)$$

Finalmente, aplicamos la función arcotangente para obtener el ángulo de dirección y lo que se conoce como GLC:

$$\delta = \tan^{-1} \left( \frac{2L\sin(\alpha)}{l_d} \right). \quad (46)$$

## 2.9. Raíz del error cuadrático medio (RMSE)

El RMSE es una métrica utilizada para medir la precisión de un modelo o la diferencia entre los valores predichos por un modelo y los valores observados reales. Es una de las métricas más comunes para evaluar modelos de regresión, así como sistemas de control. El RMSE proporciona una interpretación en la misma unidad de las variables de interés, facilitando la

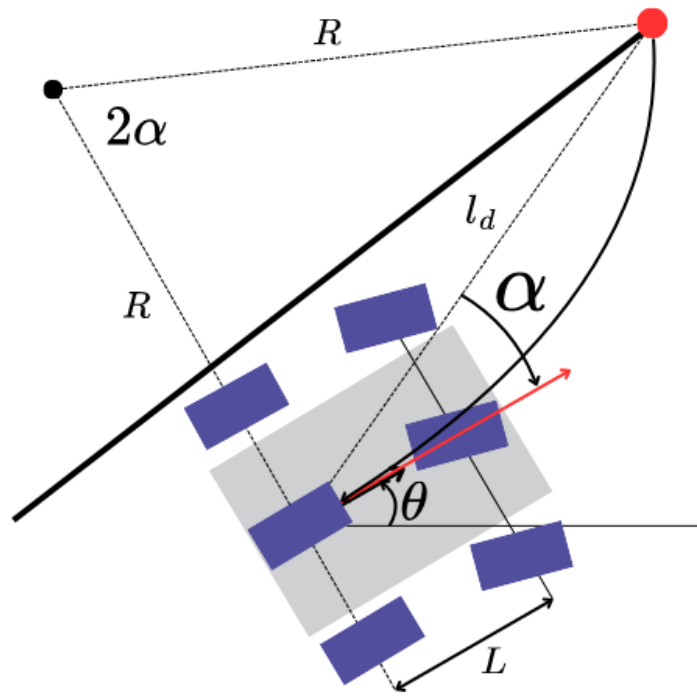


Figura 12: Geometría del PPA para el seguimiento de trayectorias.

comprensión del nivel de error. La forma de calcular este error es con la siguiente ecuación:

$$RMSE = \sqrt{\frac{1}{n} \sum_{i=1}^n (y_i - \hat{y}_i)^2} \quad (47)$$

La diferencia clave entre el error cuadrático medio y el RMSE, es que se toma la raíz cuadrada del valor del error cuadrático medio. Esto significa que el RMSE está en las mismas unidades que la variable que estamos intentando predecir, lo que facilita la interpretación de la magnitud del error. El MSE, por otro lado, proporciona un error cuadrático, por lo que sus unidades están elevadas al cuadrado, lo que puede dificultar la interpretación directa. Un valor pequeño de RMSE, indica que el modelo tiene un buen ajuste y que las predicciones están cercanas a los valores reales. El RMSE se utiliza ampliamente en aprendizaje automático para medir el rendimiento de los modelos de regresión, donde se busca minimizar este valor durante el entrenamiento del modelo. Por este motivo, la métrica de comparación de los resultados de entrenamiento con ANFIS, se miden utilizando el RMSE.

## 3. Metodología

### 3.1. Introducción

El presente trabajo de investigación tiene como objetivo el seguimiento de trayectorias de forma automática con un TR. Este TR es el resultado de años de trabajo de investigadores, principalmente el Dr. Julio Cesar Ramos Fernández como líder del proyecto y un conjunto de colaboradores [76, 77, 78, 79, 80, 81]. En la Universidad Politécnica de Pachuca se cuenta con un LANAVEX, con convenio No. 295536 en alianza con el CINVESTAV Zacatenco donde se realiza investigación y desarrollo tecnológico en las áreas de robótica aérea, terrestre y submarina para el desarrollo de aplicaciones basadas en navegación autónoma, así como robótica aplicada a rehabilitación, aumento de fuerza y resistencia.

El desarrollo de la plataforma TR involucró trabajos previos de instrumentación de navegación RTK-GNSS y servomecanismos de dirección y frenado, así como experiencias iniciales en modelado difuso T-S empleando datos entrada-salida [80]. Dichos antecedentes demostraron que el control de sistemas vehiculares basado en modelos de E/S es una alternativa altamente viable frente a las formulaciones diferenciales clásicas. Partiendo de este precedente, la presente tesis aborda la limitante de la complejidad y precisión en el modelado, proponiendo la estructura IT2-ANFIS OSTR. Este modelo se entrena algorítmicamente con datos reales provenientes de un GLC derivado del modelo matemático, sintetizando un IT2FLC capaz de gestionar las incertidumbres direccionales durante el seguimiento de trayectorias del vehículo. Adicionalmente, las modificaciones constructivas de *hardware* y *software* implementadas sobre la plataforma experimental del TR para posibilitar dichas validaciones se detallan exhaustivamente en el Apéndice A.

### 3.2. Desarrollo de una nueva estructura IT2-ANFIS

Se realizó un estudio de las diferentes metodologías de modelado difuso existentes en la literatura, con especial énfasis en ANFIS y su aplicación en el modelado de sistemas dinámicos. Con esto, se identificó que las estructuras actuales presentan limitaciones para manejar la incertidumbre y la imprecisión inherentes a sistemas complejos. Además, se observó que no existía una estructura IT2-ANFIS similar al T1-ANFIS disponible en MatLab, limitando el acceso a herramientas que permitan modelar sistemas con niveles más altos de incertidumbre. Esta carencia motivó el desarrollo de una nueva estructura que pudiera aprovechar las ventajas de los IT2FLS en el modelado y control de sistemas dinámicos. Se diseñó e implementó una nueva estructura IT2-ANFIS capaz de modelar sistemas dinámicos con base en datos E/S de manera automática. Esta estructura permite manejar la incertidumbre de manera más efectiva mediante el uso de MF con intervalos y la incorporación de algoritmos de optimización para los parámetros premisa y consecuentes. La estructura desarrollada es compatible con herramientas de software existentes y está diseñada para facilitar su implementación en aplicaciones prácticas.

### 3.2.1. Desarrollo del algoritmo de reducción de tipo de un solo paso (OSTRA)

Los TRA existentes suelen ser computacionalmente costosos debido a su naturaleza iterativa, lo que puede dificultar su aplicación en sistemas en tiempo real o con recursos limitados. Además, la complejidad de estos algoritmos puede afectar la eficiencia y precisión del sistema de control. Para superar estas limitaciones, en este trabajo de tesis se propone OSTRA. Este algoritmo permite realizar la reducción de tipo de manera directa, reduciendo significativamente el costo computacional y mejorando la velocidad de respuesta del sistema. El OSTRA simplifica el proceso de defuzificación y facilita la implementación de la nueva estructura IT2-ANFIS en aplicaciones en tiempo real. Los pasos que definen esta propuesta son los siguientes:

1. Ordenar los pesos  $\underline{w}_i$  y  $\bar{w}_i$  de la Eq. (18) en orden ascendente obteniendo un nuevo índice  $n$ .
2. Obtener  $w_n$  con los pesos inferior y superior ordenados, como  $w_n = \frac{w_n + \bar{w}_n}{2}$ .
3. Calcular los puntos de referencia  $\underline{y}$  y  $\bar{y}$  como se muestra en la Ec. (48).

$$\underline{y} = \frac{\sum_{n=1}^N \underline{w}_n w_n}{\sum_{n=1}^N w_n}, \quad \bar{y} = \frac{\sum_{n=1}^N \bar{w}_n w_n}{\sum_{n=1}^N w_n}. \quad (48)$$

4. Encontrar el punto de cambio  $l$  para  $\underline{w}_n < \underline{y} \leq \underline{w}_{n+1}$  y  $r$  para  $\bar{w}_n < \bar{y} \leq \bar{w}_{n+1}$
5. Calcular  $X$  siendo  $X = \frac{\underline{X} + \bar{X}}{2}$ , donde:

$$\underline{X} = \frac{[\bar{w}_n, \dots, \bar{w}_l, \underline{w}_{l+1}, \dots, \underline{w}_N]}{\sum_{n=1}^l \bar{w}_n + \sum_{n=l+1}^N \underline{w}_n}, \quad \bar{X} = \frac{[\underline{w}_n, \dots, \underline{w}_r, \bar{w}_{r+1}, \dots, \bar{w}_N]}{\sum_{n=1}^r \underline{w}_n + \sum_{n=r+1}^N \bar{w}_n}. \quad (49)$$

6. El vector fila  $X$  cuenta con los pesos ordenados. Sin embargo, el vector  $X$  debe estar sin ordenar:

$$\text{Desordenar}(X) = [w_i, w_{i+1}, \dots, w_N]. \quad (50)$$

7. Calcular  $y$  con el vector  $X$  desordenado como se muestra a continuación:

$$y = X \begin{bmatrix} f_i \\ f_{i+1} \\ \vdots \\ f_N \end{bmatrix}. \quad (51)$$

Las diferencias entre OSTRÁ y el M-KMA son:

1. En OSTRÁ, la ordenación se realiza con respecto a los pesos  $w_i$  y  $\bar{w}_i$  de la regla  $i$ -ésima y no con los salida de la reglas  $f_i$ .
2. La reducción de tipo de IT2 a T1 en OSTRÁ se obtiene con una sola ejecución durante cada época de entrenamiento de la estructura IT2-ANFIS. El M-KMA requiere un número desconocido de ejecuciones para encontrar  $y_l$  y  $y_r$  en cada época de entrenamiento, lo que aumenta el tiempo de ejecución del algoritmo.
3. La matriz  $X$  obtenida mediante OSTRÁ no contiene las entradas del sistema y se desordena tanto en el entrenamiento ANFIS como en el uso del modelo. Por otro lado, con M-KMA el paso de desordenamiento sólo se lleva a cabo cuando se entrena el modelo con IT2-ANFIS.
4. OSTRÁ puede utilizarse en la estructura IT2-ANFIS y con el modelo difuso, sin necesidad de modificarlo. El M-KMA sólo se utiliza durante el entrenamiento del IT2-ANFIS, para el modelo difuso debe utilizarse el KMA en el paso de reducción de tipo.

### 3.2.2. Validación de la estructura IT2-ANFIS OSTRÁ

Para evaluar la eficacia de la nueva estructura, se llevó a cabo pruebas utilizando datos sintéticos generados a partir de sistemas no lineales conocidos. Posteriormente, la estructura se aplicó a un caso real, utilizando datos de E/S del sistema de dirección (volante) del TR. Los datos fueron recolectados en diferentes entornos de trabajo del TR y reflejan las condiciones reales de funcionamiento, incluyendo la incertidumbre y la variabilidad del entorno. Los resultados obtenidos con la estructura propuesta fueron comparados con los de otras estructuras IT2-ANFIS y con el T1-ANFIS de MatLab. La comparación mostró que el IT2-ANFIS OSTRÁ proporciona una aproximación con menor error en términos de manejo de la incertidumbre, validando así nuestra propuesta.

### 3.3. Implementación del algoritmo de persecución pura en el TR

Una vez validada la estructura IT2-ANFIS OSTRÁ mediante datos sintéticos y reales del sistema de dirección, se implementó la metodología de modelado y control directamente en el TR. El objetivo inicial era minimizar el RMSE del seguimiento de trayectoria utilizando el GLC y/o un controlador basado en el error (EBC, por sus siglas en inglés). Una vez que el TR logró el seguimiento de trayectorias, se recolectaron datos reales durante la ejecución de una ruta específica. Utilizando GLC para calcular la referencia del volante, se registró el error de cabeceo y su derivada como entradas, y la referencia calculada del volante como salida. Con los datos recolectados, se modelaron los FLC utilizando tanto el T1-ANFIS de MatLab como la estructura IT2-ANFIS OSTRÁ propuesta.

El objetivo principal fue crear controladores capaces de replicar y/o mejorar el comportamiento del GLC, incorporando el manejo de la incertidumbre y la imprecisión en el control

del TR. Se implementaron los FLC en el TR, para evaluar el desempeño en condiciones reales. Los resultados obtenidos entre los diferentes controladores, se analizaron principalmente mediante el RMSE del error lateral durante la trayectoria, validando con esto la efectividad de la metodología propuesta.

### **3.4. Desarrollo de un ambiente de simulación**

Complementariamente a la validación experimental en el TR físico, se desarrolló en Python un ambiente de simulación que reproduce las condiciones cinemáticas de operación del TR. Este simulador fue concebido como una herramienta de extensión para explorar escenarios operativos que excedían las capacidades del hardware implementado, en particular el control de la velocidad longitudinal mediante FLC, para el cual el TR físico no contaba con el actuador correspondiente durante el periodo experimental. El simulador incorpora la posibilidad de introducir ruidos y perturbaciones en los sensores y actuadores, aproximando las condiciones de operación a situaciones más realistas. Se evaluó la capacidad de los controladores para mantener el seguimiento de la trayectoria ante la inyección de ruido sintético en los sensores, extendiendo así el análisis de desempeño más allá de los límites del entorno físico disponible.

## 4. Resultados y discusión

### 4.1. Nueva estructura IT2-ANFIS OSTR

En este trabajo de tesis se presenta una nueva estructura denominada IT2-ANFIS OSTR. Esta estructura es un enfoque prometedor para modelar sistemas dinámicos representados por conjuntos de datos de E/S. En el contexto del seguimiento de trayectorias de forma automática con el TR, es necesario contar con modelos de control que sean capaces de manejar las perturbaciones e incertidumbre presentes en el entorno, los mecanismos y en el sistema. Los enfoques tradicionales basados en T1-ANFIS pueden ser insuficientes debido a su limitada capacidad para representar la incertidumbre. Por otro lado, los T2FLS pueden ser computacionalmente costosos en comparación con los T1FLS. Con IT2-ANFIS OSTR se busca equilibrar estas necesidades, proporcionando un sistema capaz de modelar de manera automática sistemas basados en únicamente datos E/S, utilizando TRA de baja complejidad computacional y además, brindando al usuario como resultado un IT2FLS que representa fielmente el sistema modelado.

El desarrollo de IT2-ANFIS OSTR no surgió de manera inmediata, sino que fue el resultado de una serie de pruebas y ajustes progresivos que permitieron identificar fortalezas y limitaciones en diferentes enfoques de modelado difuso. En una primera etapa, se realizaron experimentos con diversas configuraciones de T1-ANFIS, empleando distintos algoritmos de optimización y funciones de membresía, con el objetivo de determinar la estructura que ofreciera el menor RMSE. Los resultados de estas pruebas sirvieron como plataforma de referencia para el desarrollo del IT2-ANFIS.

Posteriormente, se avanzó hacia el uso de IT2-ANFIS, en donde se introdujo el parámetro  $h$ , asociado a la amplitud de la FOU. En esta etapa inicial, aún no se disponía de un mecanismo automático para ajustar dicho parámetro, por lo que se realizaron pruebas manuales con variaciones incrementales de 0.1. Cada valor de  $h$  se evaluó mediante el RMSE, seleccionando el que ofreciera el mejor desempeño como modelo de salida. Aunque este procedimiento permitió comprobar el potencial del IT2-ANFIS frente al T1-ANFIS, también puso en evidencia la necesidad de integrar la optimización del parámetro  $h$  dentro del propio proceso de ajuste de los parámetros premisa, evitando así un ajuste manual.

De manera paralela, surgió la necesidad de contar con un algoritmo de reducción de tipo para los modelos IT2. Los algoritmos iterativos tradicionales, como EKMA y EIASC, resultaban costosos en términos computacionales y poco adecuados para aplicaciones en tiempo real. Para superar esta limitación se desarrolló OSTR, un enfoque no iterativo que permite obtener el centroide del IT2FS en un solo paso, reduciendo la carga computacional y facilitando su implementación en sistemas embebidos.

En las subsecciones siguientes se presentan los resultados de este proceso en cuatro etapas: 4.1.1 Pruebas iniciales con T1-ANFIS, 4.1.3 Experimentos preliminares con IT2-ANFIS, 4.1.2 Desarrollo del OSTR y 4.1.4 Validación de IT2-ANFIS OSTR.

Como parte del desarrollo de la estructura IT2-ANFIS OSTRÁ, se llevó a cabo un análisis de las metodologías de modelado difuso reportadas en la literatura, con especial énfasis en el enfoque ANFIS y sus variantes. Identificando que en relación con T1-ANFIS, existe una amplia variedad de metodologías y aplicaciones reportadas en la literatura. Sin embargo, a pesar de contar con publicaciones sobre T2-ANFIS e IT2-ANFIS, no existe una plataforma explotable como T1-ANFIS de MatLab. Esto motivó la propuesta de la estructura IT2-ANFIS OSTRÁ, que permite manejar la incertidumbre de manera más precisa mediante IT2FS, logrando un balance entre precisión y eficiencia computacional.

Previo al diseño de la nueva estructura IT2-ANFIS OSTRÁ, se realizaron diversas pruebas exploratorias utilizando tres plataformas de T1-ANFIS disponibles para su uso, con distintos algoritmos de optimización. El objetivo fue identificar la combinación que generara el menor RMSE en el modelado, lo cual serviría como referencia para el posterior desarrollo de la propuesta de esta tesis.

Para establecer una base de comparación, se seleccionaron dos estructuras de la literatura: T1-ANFIS de MatLab [9] y ANFIS2 Fuzzy [30]. A continuación, se presenta una tabla comparativa que muestra los algoritmos de optimización utilizados para los parámetros premisa y consecuente en cada una de las estructuras seleccionadas, incluyendo la estructura IT2-ANFIS OSTRÁ propuesta en este trabajo de tesis. La comparación entre estas estructuras se realizará más adelante, evaluando la precisión del modelado.

#### 4.1.1. Pruebas iniciales con T1-ANFIS

Con el objetivo de establecer una base metodológica para el desarrollo de la propuesta IT2-ANFIS, se llevó a cabo una comparación sistemática de tres plataformas T1-ANFIS aplicadas a conjuntos de datos heterogéneos. La métrica de evaluación fue el RMSE en entrenamiento. Cabe destacar que el tiempo de entrenamiento no fue un factor determinante en la selección: se privilegió estrictamente el valor mínimo de RMSE alcanzado. La primera plataforma, que denominamos “T1-ANFIS James”, fue desarrollada por James Power [82] y corresponde a una implementación en Python utilizando la librería Pytorch, que es una biblioteca de aprendizaje automático de código abierto. La segunda que llamamos “T1-ANFIS Muhammet”, fue desarrollada por Muhammet Balcilar [83] e implementada en código .m para MatLab, sin requerir librerías adicionales. La última plataforma, “T1-ANFIS MatLab” corresponde a la herramienta interactiva “Neuro-Fuzzy Designer” [9]. Cada plataforma emplea algoritmos de optimización distintos para los parámetros premisa y consecuentes, la Tabla 5 resume la configuración utilizada en cada caso.

Para evaluar el desempeño de las diferentes plataformas T1-ANFIS, se realizaron una serie de pruebas de modelado sobre cuatro conjuntos de datos. En cada caso se documenta:

- Número de entradas y cantidad de datos empleados en entrenamiento.
- Parámetros premisa y consecuentes obtenidos tras el entrenamiento.

Tabla 5: Plataformas T1-ANFIS comparadas y sus algoritmos de optimización.

Plataforma	Optimización de premisas	Optimización de consecuentes
T1-ANFIS James	Retropropagación	Gradiente descendente
T1-ANFIS Muhammet	Retropropagación	Filtro de Kalman
T1-ANFIS MatLab	Retropropagación	Estimación de mínimos cuadrados

- Valor de RMSE alcanzado y tiempo de entrenamiento requerido.

Se incluyen gráficas comparativas que muestran el comportamiento de cada plataforma y su desempeño relativo.

**Ejemplo 1:** El primer ejemplo de datos empleado contiene un total de 216 muestras y tres entradas independientes, generadas a partir de la función algebraica separable:

$$y = \left(1 + x_1^{0.5} + x_2^{-1} + x_3^{-1.5}\right)^2. \quad (52)$$

Este ejemplo se diseñó como un problema de regresión no lineal multivariable, en el que las entradas  $x_1$ ,  $x_2$  y  $x_3$  se combinan mediante operaciones de suma, inversión y potencias negativas [21]. A diferencia de series temporales con dinámica, esta función no presenta dependencia en el tiempo ni comportamiento caótico, lo que la convierte en un escenario adecuado para evaluar la capacidad de los modelos T1-ANFIS de aproximar relaciones algebraicas no lineales complejas. A continuación se muestran los resultados obtenidos para este ejemplo de datos.

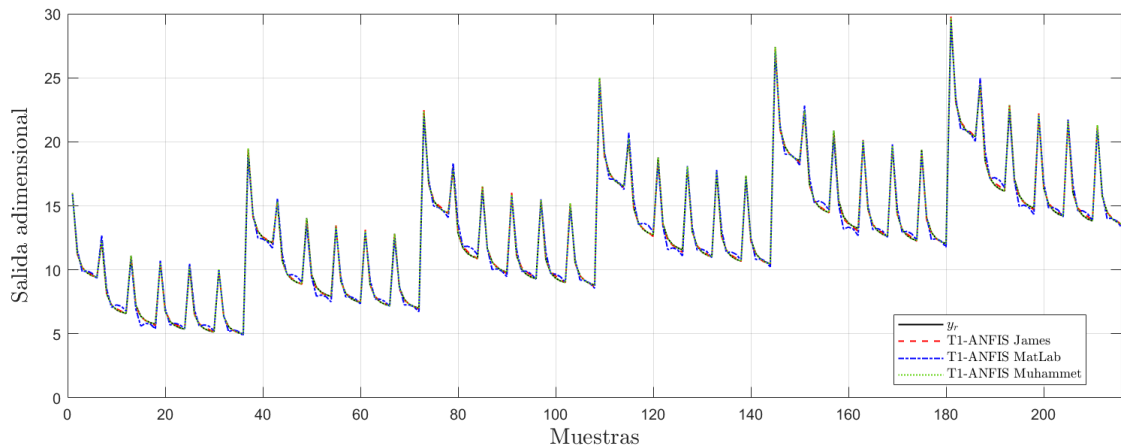


Figura 13: Aproximación de los modelos de las plataformas T1-ANFIS a la salida real del Ejemplo 1.

Tabla 6: Parámetros premisa y consecuentes de los modelos obtenidos con las plataformas T1-ANFIS para el Ejemplo 1.

Plataforma	MF	Parámetros premisa			Parámetros consecuentes			
		$a$	$b$	$c$	$C_1$	$C_2$	$C_3$	$C_4$
T1-ANFIS James	$A_{1,1}$	0.536959	2.049143	0.825728	6.4039	-7.1407	-5.2801	48.8377
	$A_{1,2}$	1.111274	1.798878	5.957784	3.0448	-3.0574	-0.2435	13.0941
	$A_{2,1}$	-2.593912	3.280021	-1.316513	5.3633	-0.6883	-4.1871	29.2706
	$A_{2,2}$	3.828057	0.261524	5.176091	2.0041	-0.2515	-0.1536	5.3313
	$A_{3,1}$	-2.869493	4.070285	-2.689668	4.3546	-8.1550	-6.1560	55.1575
	$A_{3,2}$	5.180723	-0.443843	4.820798	2.2693	-4.0721	-0.3157	16.349
					3.7086	-0.7968	-5.0637	34.3033
T1-ANFIS Muhammet	$A_{1,1}$	0.3123	0.5678	-0.1335	26.8413	12.7047	38.1349	48.0746
	$A_{1,2}$	1.2950	0.6175	6.1689	3.4291	-8.8573	0.0226	19.3004
	$A_{2,1}$	0.2285	0.7689	-0.0289	4.8220	0.3192	2.1720	24.7448
	$A_{2,2}$	1.3596	0.3680	6.2141	1.7968	-0.248	-0.0158	4.1765
	$A_{3,1}$	0.1563	1.0102	0.1509	9.3221	-7.8561	52.0763	38.3646
	$A_{3,2}$	1.3927	0.5179	6.1005	2.7448	-9.9760	-0.0353	28.9835
					3.9393	-0.3727	26.4791	24.6527
T1-ANFIS MatLab	$A_{1,1}$	2.502	1.993	0.9963	3.085	-3.786	-5.798	22.17
	$A_{1,2}$	2.520	1.997	5.983	2.359	-2.842	-1.158	18.53
	$A_{2,1}$	2.503	1.811	0.8666	2.486	-0.9769	-4.585	18.6
	$A_{2,2}$	2.781	2.057	5.697	1.76	-0.7306	-0.9171	15.14
	$A_{3,1}$	2.495	1.605	0.7314	2.33	-5.149	-7.918	28.18
	$A_{3,2}$	2.940	2.18	5.45	1.867	-4.204	-1.58	25.03
					1.948	-1.332	-6.706	25
				1.485	-1.086	-1.339	22.04	

Tabla 7: RMSE de entrenamiento y tiempo de cómputo obtenidos con las plataformas T1-ANFIS para el Ejemplo 1.

Plataforma	RMSE	Tiempo de entrenamiento (s)
T1-ANFIS James	0.10126	0.24
T1-ANFIS Muhammet	0.01113	2.47
T1-ANFIS MatLab	0.34238	0.86

Los resultados obtenidos con las tres plataformas T1-ANFIS al aproximar la función algebraica definida en el Ejemplo 1 se ilustran en la Figura 13, donde puede observarse que todas las implementaciones logran reproducir la salida de referencia, con diferencias en la precisión del ajuste. La Tabla 7 resume los valores de RMSE y tiempos de entrenamiento, mientras que la Figura 14 muestra la distribución del error adimensional a lo largo de las muestras. Este ejemplo constituye una primera evaluación comparativa entre plataformas, cuyos resultados preliminares serán complementados con los análisis de los ejemplos posteriores.

**Ejemplo 2:** El segundo ejemplo de datos corresponde a la función algebraica no lineal:

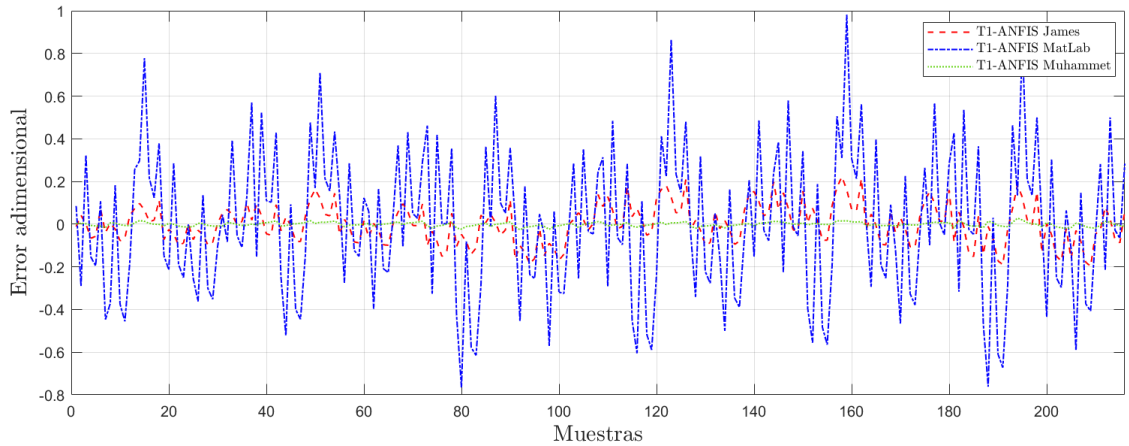


Figura 14: Error ( $y_r - y_m$ ) de los modelos de las plataformas T1-ANFIS para el Ejemplo 1.

$$z = \frac{\sin(x_1)}{x_1} \cdot \frac{\sin(x_2)}{x_2}, \quad (53)$$

la cual constituye un problema de aproximación de superficies con dos entradas y una salida. Esta función ha sido utilizada en la literatura como un banco de pruebas para modelos neuro-difusos debido a su marcada no linealidad y a la presencia de regiones de alta variación alrededor del origen [21]. Para este trabajo se emplearon 121 muestras tomadas de una discretización uniforme en el dominio  $[-10, 10] \times [-10, 10]$ . El objetivo fue evaluar la capacidad de las tres plataformas T1-ANFIS para aproximar la salida a partir de estos datos, analizando el ajuste obtenido mediante el RMSE de entrenamiento.

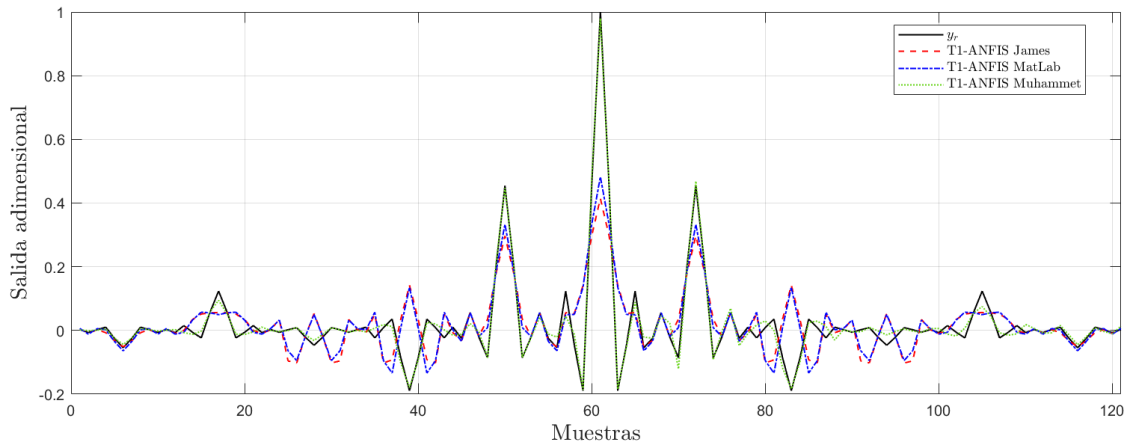


Figura 15: Aproximación de los modelos de las plataformas T1-ANFIS a la salida real del Ejemplo 2.

Los resultados obtenidos con las tres plataformas T1-ANFIS al aproximar la función se presentan en la Figura 15. En ella se observa que los modelos logran capturar la forma global de

Tabla 8: Parámetros premisa y consecuentes de los modelos obtenidos con las plataformas T1-ANFIS para el Ejemplo 2.

Plataforma	MF	Parámetros premisa			Parámetros consecuentes		
		$a$	$b$	$c$	$C_1$	$C_2$	$C_3$
T1-ANFIS James	$A_{1,1}$	3.35525	1.99428	-9.98217	-0.01816	-0.01816	-0.36918
	$A_{1,2}$	3.33070	1.99345	-3.32729	0.07704	-0.01923	0.73622
	$A_{1,3}$	3.33070	1.99345	3.32729	0.07704	0.01923	0.73621
	$A_{1,4}$	3.35525	1.99428	9.98217	-0.01815	0.01815	-0.36918
	$A_{2,1}$	3.35525	1.99428	-9.98217	-0.01923	0.07704	0.73621
	$A_{2,2}$	3.33070	1.99345	-3.32729	0.08158	0.08158	0.39856
	$A_{2,3}$	3.33070	1.99345	3.32729	0.08158	-0.08158	0.39856
	$A_{2,4}$	3.35525	1.99428	9.98217	-0.01923	-0.07704	0.73621
					0.01923	0.07704	0.73621
					-0.08158	0.08158	0.39856
					-0.08158	-0.08158	0.39856
					0.01923	-0.07704	0.73621
					0.01815	-0.01815	-0.36917
					-0.07704	-0.01923	0.73621
					-0.07704	0.01923	0.73621
					0.01815	0.01815	-0.36917
T1-ANFIS Muhammet	$A_{1,1}$	4.9816	4.1977	-7.0833	0.0567	-0.0004	0.7707
	$A_{1,2}$	3.5724	1.0566	-1.0587	1.2053	0.0178	16.4678
	$A_{1,3}$	2.4611	1.6786	0.3009	-0.5329	-0.0292	-7.4375
	$A_{1,4}$	5.1821	4.2527	6.8557	0.0946	-0.0071	1.4390
	$A_{2,1}$	5.3730	4.2686	-6.5859	0.0833	-0.0478	-1.4921
	$A_{2,2}$	2.2142	1.9216	0.0807	1.2424	-0.5111	-32.7170
	$A_{2,3}$	3.6322	1.0112	0.7070	-0.6705	0.5802	15.5966
	$A_{2,4}$	4.9275	4.1994	7.1262	0.1336	0.2031	-5.3224
					-0.3745	0.2549	2.6672
					-7.9302	2.3451	44.7849
					3.5129	-2.5101	-26.6018
					-0.6242	-0.9312	16.3649
					-0.0342	-0.0137	0.3315
					-0.8133	-0.1133	9.4470
					0.3398	0.1153	-3.5138
					-0.0579	0.0450	0.0315
T1-ANFIS MatLab	$A_{1,1}$	3.679	1.963	-9.692	-0.02359	-0.02359	-0.4875
	$A_{1,2}$	3.293	1.874	-3.210	0.08671	-0.03064	0.8341
	$A_{1,3}$	3.293	1.874	3.210	0.08671	0.03064	0.8341
	$A_{1,4}$	3.679	1.963	9.692	-0.02359	0.02359	-0.4875
	$A_{2,1}$	3.679	1.963	-9.692	-0.03064	0.08671	0.8341
	$A_{2,2}$	3.293	1.874	-3.210	0.1126	0.1126	0.4550
	$A_{2,3}$	3.293	1.874	3.210	0.1126	-0.1126	0.4550
	$A_{2,4}$	3.679	1.963	9.692	-0.03064	-0.08671	0.8341
					0.03064	0.08671	0.8341
					-0.1126	0.1126	0.4550
					-0.1126	-0.1126	0.4550
					0.03064	-0.08671	0.8341
					0.02359	-0.02359	-0.4875
					-0.08671	-0.03064	0.8341
					-0.08671	0.03064	0.8341
					0.02359	0.02359	-0.4875

la superficie, aunque con diferencias en la fidelidad local cerca del origen, donde la función presenta mayor variación. La Tabla 9 muestra los valores de RMSE y tiempos de entrenamiento alcanzados, mientras que las gráficas de error (Figura 16) permiten apreciar la dis-

Tabla 9: RMSE de entrenamiento y tiempo de cómputo obtenidos con las plataformas T1-ANFIS para el Ejemplo 2.

Plataforma	RMSE	Tiempo de entrenamiento (s)
T1-ANFIS James	0.10465	1.233
T1-ANFIS Muhammet	0.02006	33.363
T1-ANFIS MatLab	0.09829	10.140

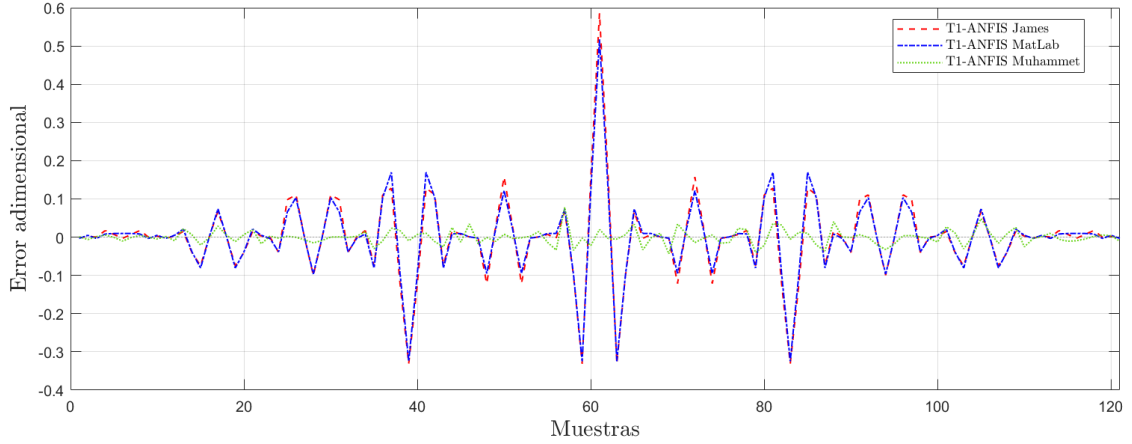


Figura 16: Error ( $y_r - y_m$ ) de los modelos de las plataformas T1-ANFIS para el Ejemplo 2.

persión de cada implementación. Estos resultados refuerzan el análisis comparativo iniciado en el Ejemplo 1 y aportan evidencia adicional sobre las capacidades y limitaciones de cada plataforma en tareas de modelado no lineal.

**Ejemplo 3:** El tercer ejemplo de datos fue obtenido a partir de la planta real TR, durante los estudios de maestría del autor. En esta configuración, el sistema era controlado mediante un único microcontrolador LaunchPad TM4C1294, tal como se describe en la sección A.1.1. El objetivo de este ejemplo es modelar la dinámica de la dirección del vehículo, expresada como el ángulo del volante en grados. El esquema de identificación se construyó con dos entradas, una salida y 456 muestras: la primera entrada corresponde al valor de la salida en el instante anterior  $y(k)$ , y la segunda es la señal de control  $u(k)$ , normalizada en el rango  $[-1, 1]$ . La salida objetivo es  $y(k+1)$ , que representa la posición del volante en grados un paso adelante. De esta forma, el problema puede representarse como:

$$y(k+1) = y(k) + u(k). \quad (54)$$

Este ejemplo resulta relevante porque refleja un escenario de aplicación experimental y real, distinto a los ejemplos anteriores basados en funciones algebraicas o superficies sintéticas. De este modo, permite evaluar la capacidad de los modelos para capturar la dinámica de un sistema físico bajo condiciones de operación reales.

Tabla 10: Parámetros premisa y consecuentes de los modelos obtenidos con las plataformas T1-ANFIS para el Ejemplo 3.

Plataforma	MF / Regla	Parámetros premisa			Parámetros consecuentes		
		$a$	$b$	$c$	$C_1$	$C_2$	$C_3$
T1-ANFIS James	$A_{1,1}$	12.33925	1.94162	-33.99908	1.00414	0.97203	-0.00071
	$A_{1,2}$	12.32781	2.02013	40.00279	0.22936	-3.47207	5.68877
	$A_{2,1}$	0.57791	1.91491	-0.98982	0.99825	1.14983	-0.07924
	$A_{2,2}$	0.20533	1.97835	1.20865	0.87147	-1.52932	5.2498
T1-ANFIS Muhammet	$A_{1,1}$	12.4153	1.2064	-33.9826	1.1095	-28.2677	-40.3558
	$A_{1,2}$	12.2827	1.9062	40.0810	0.9720	-2.9940	8.5409
	$A_{2,1}$	0.4726	0.8854	-1.7910	1.2254	-36.0538	-64.8156
	$A_{2,2}$	1.1000	0.4236	1.7079	0.9427	-5.1097	13.9032
T1-ANFIS MatLab	$A_{1,1}$	37.02	2.026	-33.98	1.107	-37.05	-52.9
	$A_{1,2}$	36.99	1.845	40.01	0.8786	-42.3	58.31
	$A_{2,1}$	1.21	2.061	-0.7068	1.081	0.6172	-4.65
	$A_{2,2}$	1.283	1.941	0.8688	0.906	-0.856	5.053

Tabla 11: RMSE de entrenamiento y tiempo de cómputo obtenidos con las plataformas T1-ANFIS para el Ejemplo 3.

Plataforma	RMSE	Tiempo de entrenamiento (s)
T1-ANFIS James	0.43863	1.3954
T1-ANFIS Muhammet	0.435031	18.6541
T1-ANFIS MatLab	0.43519	6.7200

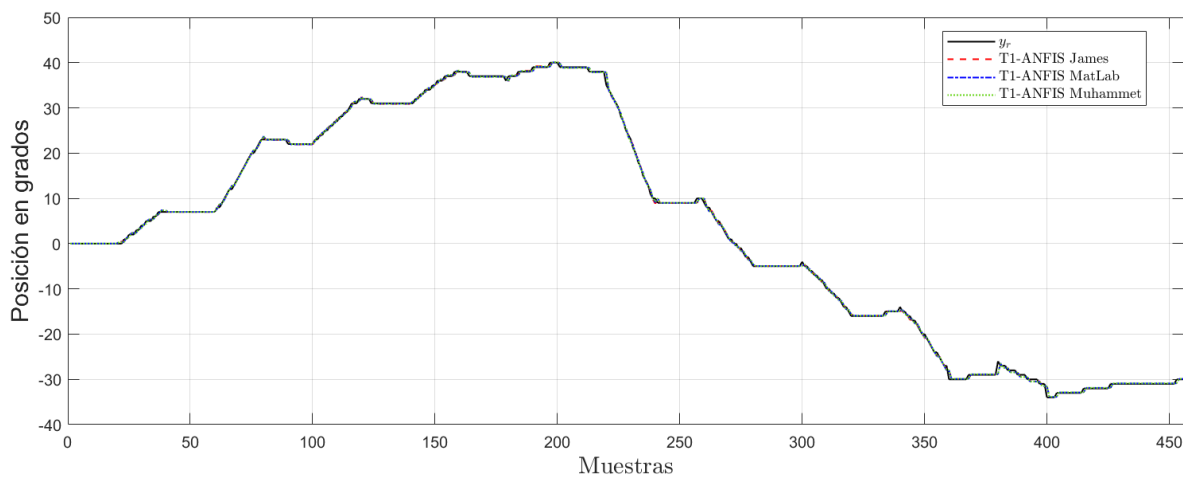


Figura 17: Aproximación de los modelos de las plataformas T1-ANFIS a la salida real del Ejemplo 3.

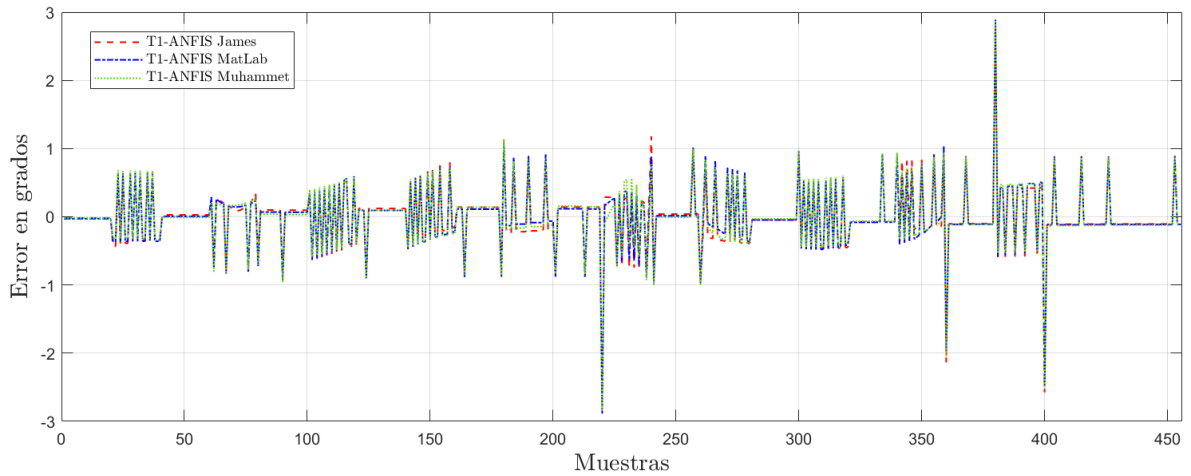


Figura 18: Error ( $y_r - y_m$ ) de los modelos de las plataformas T1-ANFIS para el Ejemplo 3.

En síntesis, los resultados obtenidos con el Ejemplo 3 ilustrados en las Figuras 17 y 18, muestran que las tres plataformas T1-ANFIS fueron capaces de reproducir con precisión la evolución de la posición del volante en grados. La Tabla 11 confirma que, al igual que en los ejemplos anteriores, el menor RMSE de entrenamiento correspondió a la plataforma T1-ANFIS Muhammet. Considerando de manera conjunta los tres ejemplos de datos empleados, puede concluirse que dicha implementación basada en la optimización mediante KF proporcionó consistentemente el mejor ajuste, superando a las alternativas en términos de exactitud de modelado y consolidándose como la base para el desarrollo posterior.

#### 4.1.2. Desarrollo del OSTR

El desarrollo del OSTR surge como una necesidad de mejorar la eficiencia computacional en la defuzzificación de sistemas IT2, especialmente en aplicaciones de control en tiempo real. Los TRA existentes, como el algoritmo de KM y sus variantes, suelen ser computacionalmente intensivos debido a la naturaleza iterativa del proceso de reducción de tipo. Esto representa una limitación en sistemas donde se requiere una respuesta rápida y precisa, como es el caso del TR. El objetivo de OSTR es simplificar el proceso de reducción de tipo mediante un enfoque no iterativo que permite obtener el centroide del IT2FS en un solo paso, reduciendo la carga computacional y facilitando su implementación en sistemas embebidos.

Para ilustrar las ventajas del TRA propuesto, se realiza una comparación de OSTR con el EIASC [12], KM [10] y EKM [11]. Los algoritmos se utilizan para calcular la salida de tres IT2FS cuyas FOUs se representan en la Figura 19. La idea de la comparación computacional es obtener el tiempo medio de cálculo que cada algoritmo emplea en encontrar la salida de una FOU dada. El procedimiento experimental para llevar a cabo esta comparación se describe a continuación:

- Establecer el algoritmo en la plataforma computacional.
- Establecer una FOU con una discretización de N puntos.
- Medir el tiempo de cálculo demandado por M=100 ejecuciones del algoritmo.
- Registrar los valores de salida obtenidos con el fin de establecer una comparación numérica.

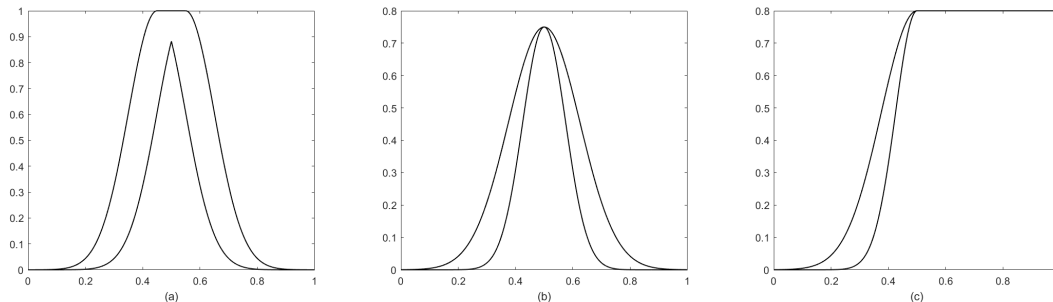


Figura 19: FOU utilizadas para la comparación computacional entre los algoritmos OSTR, EIASC, KM y EKM. (a) Gaussiana con media incierta, (b) Gaussiana con desviación típica incierta y (c) Gaussiana con media incierta (lado derecho).

Los experimentos se llevan a cabo en un Intel Core i9-12900H con 16GB de RAM y entorno informático Python 3.11.5. Las Tablas 12-14 presentan los resultados relativos al tiempo de computación del algoritmo y las salidas. Los algoritmos son equivalentes en términos de precisión aritmética, ya que proporcionan los mismos resultados para las funciones de salida. La diferencia entre los tiempos de cálculo de cada algoritmo es similar, sin embargo, OSTR tiene el tiempo de cálculo más corto. La diferencia en los tiempos de cálculo está relacionada con la complejidad computacional de los algoritmos más que con sus propiedades matemáticas. Por este motivo, OSTR se ha propuesto como solución alternativa para calcular el centroide de un IT2FS con una sola ejecución.

El OSTR propuesto en este trabajo permite realizar el proceso de defuzificación de IT2FLS. Los resultados experimentales muestran que mantiene la precisión de los algoritmos convencionales, mientras que reduce el tiempo de cálculo, convirtiéndolo en una opción viable para aplicaciones en tiempo real y de control de trayectorias.

#### 4.1.3. Experimentos preliminares con IT2-ANFIS

Con el propósito de evaluar preliminarmente el desempeño de un modelo IT2-ANFIS en comparación con T1-ANFIS, se llevaron a cabo experimentos empleando tres plataformas: T1-ANFIS MatLab, IT2-ANFIS con reducción de tipo mediante el M-KMA e IT2-ANFIS con reducción de tipo mediante el OSTR. En esta etapa, los modelos IT2-ANFIS utilizados no contaban aún con un mecanismo de optimización del parámetro  $h$ . En consecuencia,

Tabla 12: Resultados para un IT2FS: gaussiana con media incierta.

N	OSTRA		EIASC		KMA		EKMA	
	Salida	Tiempo	Salida	Tiempo	Salida	Tiempo	Salida	Tiempo
10	0.9021	0.001	0.9021	0.002	0.9021	0.005	0.9021	0.003
100	0.8903	0.006	0.8903	0.008	0.8903	0.019	0.8903	0.013
1000	0.8408	0.04	0.8408	0.05	0.8408	0.14	0.8408	0.16
10000	0.7209	0.4	0.7209	0.5	0.7209	1.3	0.7209	1.4

Tabla 13: Resultados para un IT2FS: gaussiana con desviación típica incierta.

N	OSTRA		EIASC		KMA		EKMA	
	Salida	Tiempo	Salida	Tiempo	Salida	Tiempo	Salida	Tiempo
10	0.4877	0.001	0.4877	0.003	0.4877	0.005	0.4877	0.005
100	0.5701	0.005	0.5701	0.008	0.5701	0.02	0.5701	0.01
1000	1.0151	0.04	1.0151	0.06	1.0151	0.15	1.0151	0.17
10000	1.0855	0.4	1.0855	0.5	1.0855	1.4	1.0855	1.5

Tabla 14: Resultados para un IT2FS: gaussiana con media incierta (lado derecho).

N	OSTRA		EIASC		KMA		EKMA	
	Salida	Tiempo	Salida	Tiempo	Salida	Tiempo	Salida	Tiempo
10	0.8847	0.002	0.8847	0.003	0.8847	0.006	0.8847	0.004
100	0.8624	0.006	0.8624	0.007	0.8624	0.019	0.8624	0.018
1000	0.9069	0.03	0.9069	0.06	0.9069	0.12	0.9069	0.18
10000	1.2895	0.2	1.2895	0.6	1.2895	1.2	1.2895	1.6

dicho parámetro fue ajustado de manera manual realizando variaciones en incrementos de 0.1 dentro del rango  $[0.1, 1]$ , a fin de observar su influencia directa sobre el RMSE de entrenamiento. Este análisis constituye un primer acercamiento a la evaluación del IT2-ANFIS bajo condiciones controladas, y permitió establecer una base de referencia para el desarrollo posterior, en el que la integración de la optimización automática de  $h$  es un elemento clave para consolidar la propuesta IT2-ANFIS OSTRA.

Con el fin de profundizar en la evaluación del desempeño del IT2-ANFIS y contrastarlo con su contraparte T1-ANFIS, se empleó un conjunto ampliado de experimentos que abarcan doce ejemplos de datos distintos (del Ejemplo 4 al 16). En todos los casos se compararon tres configuraciones: T1-ANFIS MatLab, IT2-ANFIS con reducción de tipo mediante el M-KMA, cuya implementación en código MatLab fue desarrollada en este trabajo a partir de la interpretación del esquema metodológico descrito en [35], dado que el artículo no proporciona un código fuente explícito; e IT2-ANFIS con reducción de tipo mediante el algoritmo OSTR, propuesto en este trabajo como alternativa no iterativa.

Cabe señalar que en esta fase, los modelos IT2-ANFIS empleados no cuentan con un mecanismo de optimización automática del parámetro  $h$ . Por tanto, dicho parámetro se ajusta de manera manual, explorando valores en incrementos de 0.1 dentro del intervalo  $[0.1, 1]$ . Este procedimiento permite analizar la sensibilidad del modelo respecto a la selección de  $h$  y establecer patrones de comportamiento del RMSE de entrenamiento. De forma similar a los ejemplos evaluados anteriormente, para cada ejemplo de datos se presentan:

- Tablas comparativas con los valores de RMSE alcanzados por las tres plataformas, junto con los valores de  $h$  utilizados en los modelos IT2-ANFIS.
- Gráficas de aproximación, donde se contrastan las salidas de cada modelo con la señal de referencia.
- Gráficas del error, que permiten observar las diferencias de precisión entre las plataformas en cada caso.

Los resultados constituyen un bloque de experimentación sistemática que amplía el análisis inicial, ofreciendo una evaluación detallada del comportamiento de los modelos IT2-ANFIS bajo diferentes condiciones de entrada y sirviendo de base para la posterior integración de la optimización automática de  $h$  en la propuesta final IT2-ANFIS OSTR.

**Ejemplo 4 (versión reducida del Ejemplo 1):** Este ejemplo corresponde a la misma función algebraica descrita previamente en la Sección 4.1.1 Ejemplo 1, definida por la Eq. (52), pero en este caso se utilizó una muestra reducida de 100 datos, en contraste con las 216 empleadas previamente para la comparación entre plataformas T1-ANFIS. La reducción del tamaño de los datos tuvo como finalidad analizar el comportamiento de los modelos IT2-ANFIS contra T1-ANFIS en condiciones de entrenamiento más acotadas, permitiendo explorar la influencia del parámetro  $h$  en contextos de menor disponibilidad de información.

Los resultados correspondientes a este ejemplo se presentan en la Tabla 15. En ella se aprecia que la plataforma T1-ANFIS MatLab alcanzó un RMSE de entrenamiento menor en comparación con las configuraciones IT2-ANFIS. Entre estas últimas, aunque los valores obtenidos no resultan competitivos frente al modelo T1-ANFIS, se observa una diferencia cuando se emplea OSTR como algoritmo de reducción de tipo frente a M-KMA. Las Figuras 20 y

21 muestran respectivamente, la aproximación de las tres plataformas respecto a la señal de referencia y la evolución del error de entrenamiento. Puede apreciarse que, si bien todas las configuraciones lograron reproducir cualitativamente el comportamiento de la función objetivo, los modelos IT2-ANFIS evidenciaron una mayor dispersión del error. Estos hallazgos consolidan al Ejemplo 4 como un escenario preliminar para analizar la sensibilidad de los modelos IT2-ANFIS antes de la incorporación de un mecanismo de optimización del parámetro  $h$ .

Tabla 15: RMSE de entrenamiento de las plataformas T1-ANFIS, IT2-ANFIS M-KMA e IT2-ANFIS para el Ejemplo 4, considerando diferentes valores manuales del parámetro  $h$ .

$h$	T1-ANFIS MatLab	IT2-ANFIS M-KMA	IT2-ANFIS OSTRA
1.0	0.00921	1.64793	0.92632
0.9		1.82298	0.78064
0.8		1.69208	1.25315
0.7		2.06771	1.20172
0.6		1.13644	1.31944
0.5		2.02305	1.10260
0.4		1.89715	1.27504
0.3		2.30743	1.37352
0.2		1.29905	1.17012
0.1		1.56737	1.42202

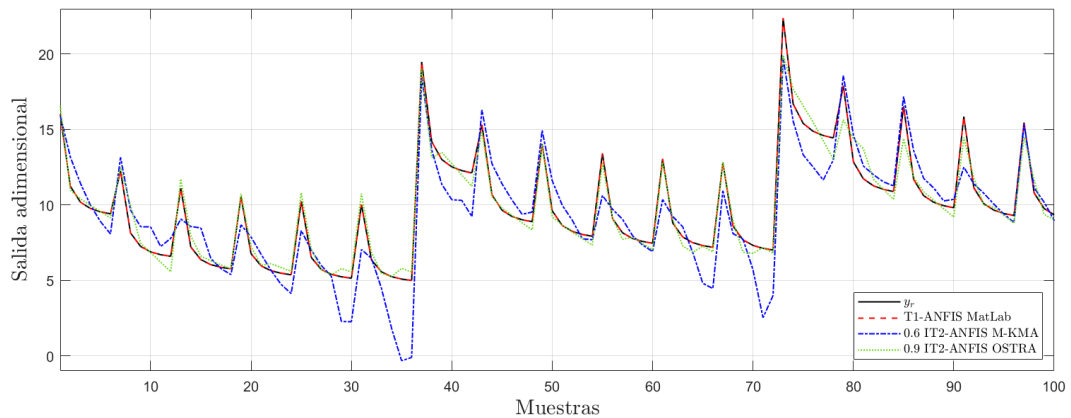


Figura 20: Aproximación obtenida con las plataformas T1-ANFIS MatLab, IT2-ANFIS M-KMA e IT2-ANFIS OSTRA a la salida real del Ejemplo 4.

**Ejemplo 5:** Este ejemplo de datos se construye a partir de una función de la misma familia que la empleada en el Ejemplo 2 de la Sección 4.1.1, pero en este caso restringida a una dimensión. En lugar de considerar la forma bidimensional de la función sinc, se evalúa únicamente a lo largo de la diagonal lo que conduce a la ecuación:

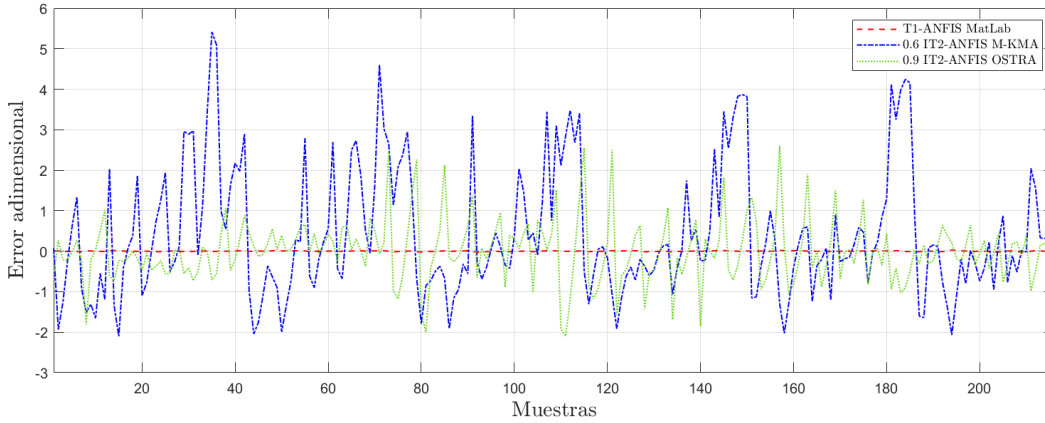


Figura 21: Error ( $y_r - y_m$ ) de las plataformas T1-ANFIS MatLab, IT2-ANFIS M-KMA e IT2-ANFIS OSTR para el Ejemplo 4.

$$z(t) = \left( \frac{\sin t}{t} \right)^2, t \in [0.1, 10]. \quad (55)$$

El ejemplo se compone de 100 muestras equidistantes con dos entradas coincidentes ( $x_1 = x_2 = t$ ) y una salida  $z(t)$ . Esta formulación simplificada conserva las propiedades oscilatorias y la atenuación característica de la sinc, pero reduce la complejidad dimensional, permitiendo analizar el comportamiento de los modelos bajo condiciones más acotadas.

Los resultados obtenidos se presentan en la Tabla 16, donde se aprecia que la plataforma T1-ANFIS MatLab alcanzó un RMSE de entrenamiento más bajo en comparación con los modelos IT2-ANFIS, aunque de magnitud cercana al obtenido mediante OSTR. Entre las configuraciones IT2, el uso de OSTR como método de reducción de tipo mostró un desempeño analizado frente al de M-KMA, lo que evidencia su comportamiento en este escenario. Las Figuras 22 y 23 ilustran la aproximación de las salidas frente a la referencia y la distribución del error de entrenamiento. Puede observarse que tanto T1-ANFIS como IT2-ANFIS con OSTR lograron reproducir la tendencia general de la función, mientras que la implementación basada en M-KMA presentó una mayor dispersión del error. Este ejemplo refuerza el análisis al poner de manifiesto que, en ausencia de un mecanismo de optimización automática del parámetro  $h$  los resultados de los modelos IT2-ANFIS explorados en intervalos de 0.1 dentro de  $[0.1, 1]$ , deja sin cubrir posibles valores intermedios que podrían conducir a soluciones con menor error.

**Ejemplo 6:** Corresponde al denominado Ejemplo 2 en [21] con la diferencia de que no se eleva la función al cuadrado, resultando en la ecuación que se muestra a continuación:

$$y_r = 1 + x_1^{0.5} + x_2^{-1} + x_3^{-1.5}, \quad (56)$$

donde las tres entradas son  $x_1, x_2, x_3 \in [1, 6]$  y la salida es  $y_r$ . Para la generación de este con-

Tabla 16: RMSE de entrenamiento de las plataformas T1-ANFIS, IT2-ANFIS M-KMA e IT2-ANFIS para el Ejemplo 5, considerando diferentes valores manuales del parámetro  $h$ .

$h$	T1-ANFIS MatLab	IT2-ANFIS M-KMA	IT2-ANFIS OSTRA
1.0	0.006389834	0.105203652	0.009520328
0.9		0.101446213	0.010295249
0.8		0.189254855	0.010088661
0.7		0.105203664	0.010527346
0.6		0.171984094	0.010609069
0.5		0.105203652	0.008313221
0.4		0.164421199	0.006592430
0.3		0.105203635	0.013750356
0.2		0.233366421	0.011427492
0.1		0.193155732	0.012273932

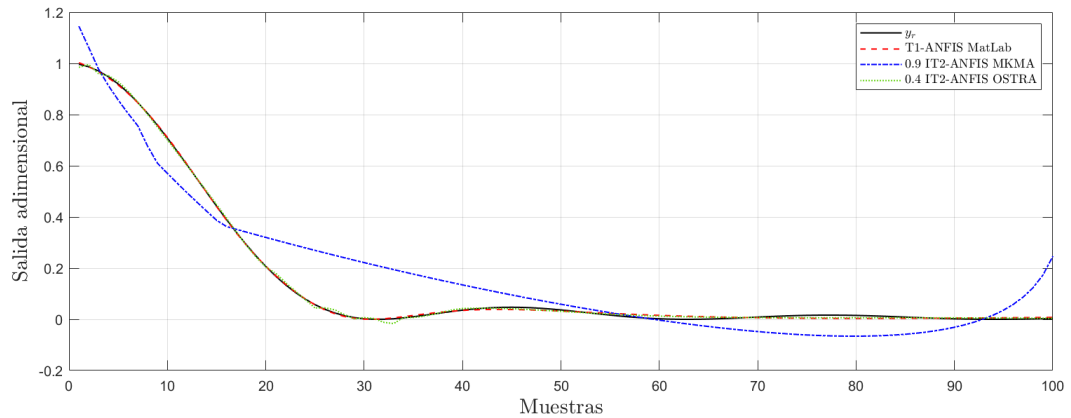


Figura 22: Aproximación obtenida con las plataformas T1-ANFIS MatLab, IT2-ANFIS M-KMA e IT2-ANFIS OSTRA a la salida real del Ejemplo 5.

junto se consideraron 216 muestras. Adicionalmente, con el fin de evaluar la robustez de los modelos frente a condiciones anómalas, en la muestra 100 se introdujo un efecto de desconexión momentánea del sensor, lo que provoca una variación abrupta en la señal de referencia. Este conjunto de datos resulta particularmente relevante, ya que permite analizar la capacidad de los modelos no solo para aproximar funciones no lineales con tres entradas, sino también para responder ante perturbaciones bruscas.

Los resultados obtenidos se muestran en la Tabla 17, donde se aprecia que el modelo IT2-ANFIS OSTRA alcanzó un RMSE de entrenamiento menor que el T1-ANFIS MatLab, confirmando su mayor precisión en este escenario. Las Figuras 24 y 25 presentan, respectivamente, la comparación entre las salidas de los modelos y la referencia, así como la distribución del error de entrenamiento. Cabe destacar que la estructura IT2-ANFIS OSTRA logró una

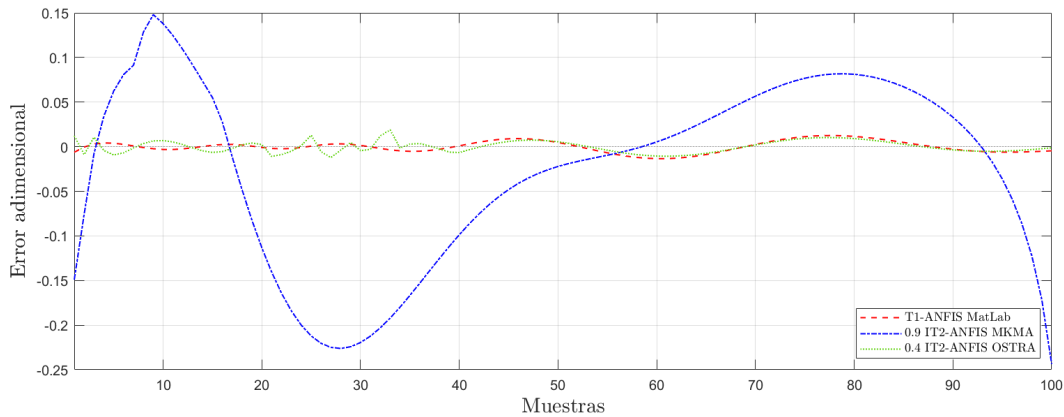


Figura 23: Error ( $y_r - y_m$ ) de las plataformas T1-ANFIS MatLab, IT2-ANFIS M-KMA e IT2-ANFIS OSTRÁ para el Ejemplo 5.

aproximación más estable, con un sobreimpulso reducido y una rápida recuperación en la muestra 100, donde se introdujo la perturbación. Estos resultados refuerzan la efectividad de OSTRÁ como método de reducción de tipo en situaciones que exigen robustez ante variaciones abruptas en los datos.

Tabla 17: RMSE de entrenamiento de las plataformas T1-ANFIS, IT2-ANFIS M-KMA e IT2-ANFIS para el Ejemplo 6, considerando diferentes valores manuales del parámetro  $h$ .

$h$	T1-ANFIS MatLab	IT2-ANFIS M-KMA	IT2-ANFIS OSTRÁ
1.0	0.213805827	1.017835885	0.151540413
0.9		1.076273190	0.145823679
0.8		0.813276696	0.106410243
0.7		0.959934127	0.089604804
0.6		1.061633669	0.124807551
0.5		0.931242802	0.104882151
0.4		0.880294209	0.091077129
0.3		0.739093217	0.081393375
0.2		1.241977881	0.109057751
0.1		0.744976534	0.055279567

**Ejemplo 7:** Corresponde a datos reales del ámbito financiero, obtenidos de la base de datos pública Istanbul Stock Exchange del repositorio UCI Machine Learning [84]. Este ejemplo incluye los rendimientos de la Bolsa de Valores de Estambul junto con siete índices bursátiles internacionales: SP, DAX, FTSE, NIKKEI, BOVESPA, MSCI EU y MSCI EM, en el periodo comprendido entre el 5 de junio de 2009 y el 22 de febrero de 2011. Para este estudio se seleccionaron como variables de entrada los índices ISE, FTSE y MSCI EU, con el objetivo de predecir el índice MSCI EM. Este tipo de datos resulta especialmente relevante, pues re-

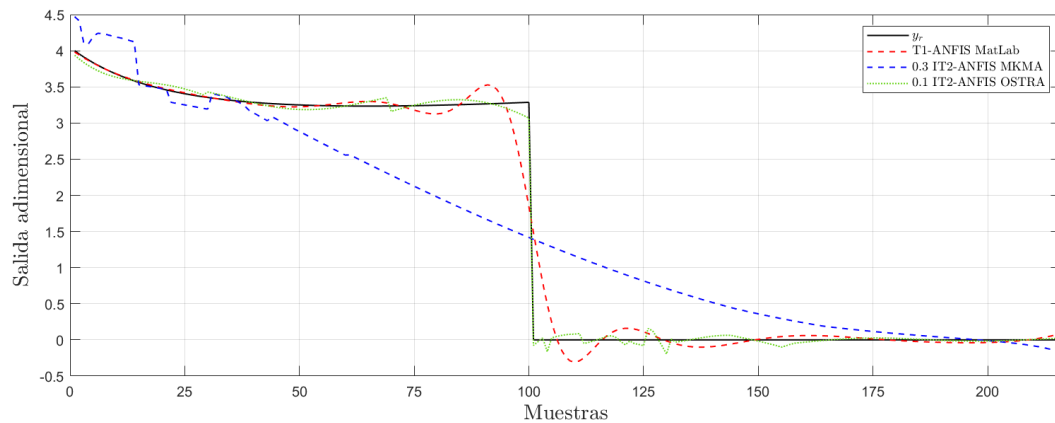


Figura 24: Aproximación obtenida con T1-ANFIS MatLab, IT2-ANFIS M-KMA e IT2-ANFIS OSTR para el Ejemplo 6, en comparación con la referencia.

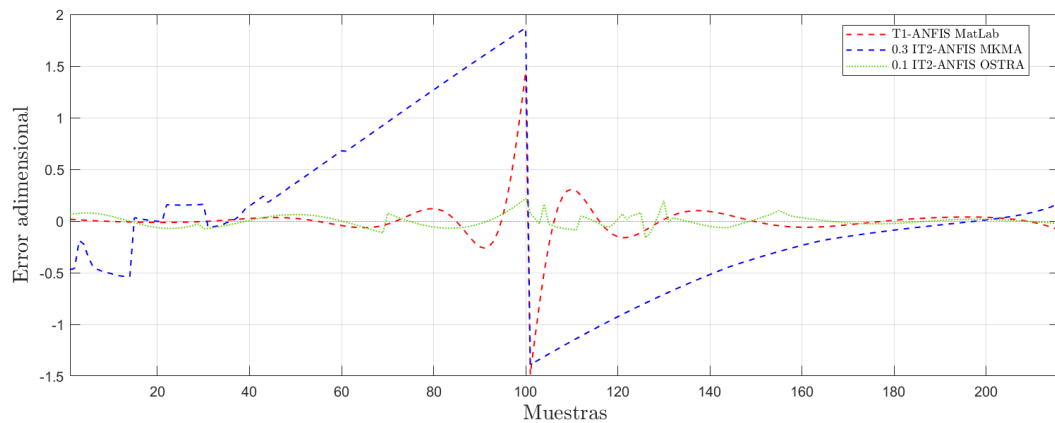


Figura 25: Error de entrenamiento de T1-ANFIS MatLab, IT2-ANFIS M-KMA e IT2-ANFIS OSTR para el Ejemplo 6.

presenta un escenario no controlado y de alta variabilidad, distinto a las funciones sintéticas o experimentales de ejemplos anteriores.

Los resultados se presentan en la Tabla 18 donde se observa que la configuración IT2-ANFIS OSTR alcanzó el menor RMSE del ejemplo (0.005332165 con  $h = 0.8$ ), y además superó consistentemente a T1-ANFIS MatLab para todos los valores de  $h$  ensayados. En contraste, IT2-ANFIS M-KMA mostró errores superiores, aun en su mejor caso. Las Figuras 26 y 27 exhiben, respectivamente, la aproximación frente a la referencia y la distribución del error: se aprecia que OSTR entrega el ajuste más preciso dentro de las variantes IT2, confirmando su ventaja como método de reducción de tipo en este escenario financiero.

**Ejemplos 8 y 9:** Se basan en la base de datos de eficiencia energética disponible en la UCI Machine Learning Repository [85], la cual contiene ocho variables de entrada relacionadas con las características arquitectónicas y de diseño de edificios (por ejemplo, relación de as-

Tabla 18: RMSE de entrenamiento de las plataformas T1-ANFIS, IT2-ANFIS M-KMA e IT2-ANFIS para el Ejemplo 7, considerando diferentes valores manuales del parámetro  $h$ .

$h$	T1-ANFIS MatLab	IT2-ANFIS M-KMA	IT2-ANFIS OSTR
1.0	0.005652374	0.006490951	0.005472415
0.9		0.006581128	0.005418749
0.8		0.006491413	0.005332165
0.7		0.006950823	0.005451813
0.6		0.006659436	0.005438343
0.5		0.006647122	0.005361593
0.4		0.006342589	0.005486104
0.3		0.006640622	0.005423063
0.2		0.006478239	0.005468495
0.1		0.006552363	0.005527229

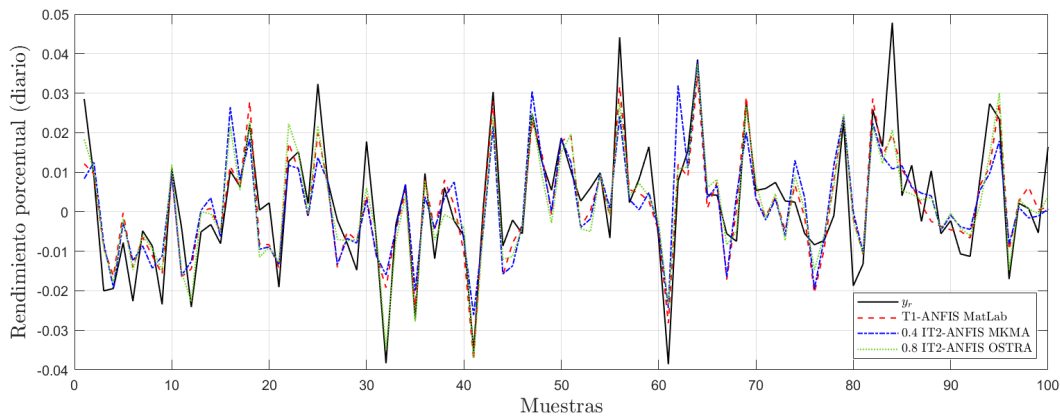


Figura 26: Aproximación obtenida con T1-ANFIS MatLab, IT2-ANFIS M-KMA e IT2-ANFIS OSTR para el Ejemplo 7, en comparación con la referencia.

pecto, área de superficie, altura, orientación, entre otras). Dicha base proporciona dos salidas diferentes: la carga de calefacción y la carga de enfriamiento, motivo por el cual se consideraron dos ejemplos independientes para los experimentos. En ambos casos, se evaluó el desempeño comparativo de T1-ANFIS, IT2-ANFIS con reducción de tipo M-KMA, e IT2-ANFIS OSTR. Estos ejemplos representan un escenario de alta dimensionalidad en las entradas, lo que permite observar cómo los modelos responden a la complejidad creciente de las relaciones no lineales.

Los resultados obtenidos muestran que, mientras que T1-ANFIS e IT2-ANFIS M-KMA no logran un desempeño competitivo, el IT2-ANFIS OSTR presenta un rendimiento superior, alcanzando un RMSE de entrenamiento varias órdenes de magnitud menores en ambos ejemplos. Este hallazgo refuerza la capacidad de IT2-ANFIS OSTR para manejar escenarios con

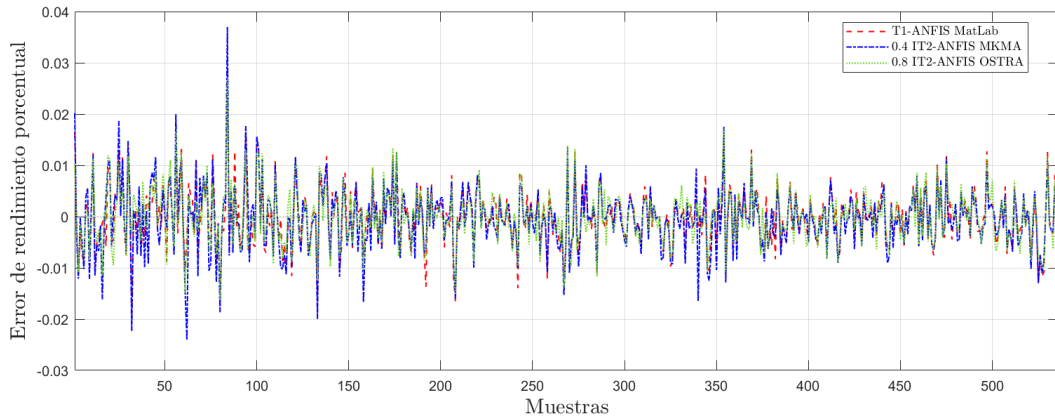


Figura 27: Error de entrenamiento de T1-ANFIS MatLab, IT2-ANFIS M-KMA e IT2-ANFIS OSTR para el Ejemplo 7.

múltiples entradas y funciones altamente no lineales, manteniendo estabilidad y precisión en la salida modelada.

Tabla 19: RMSE de entrenamiento de las plataformas T1-ANFIS, IT2-ANFIS M-KMA e IT2-ANFIS para el Ejemplo 8, considerando diferentes valores manuales del parámetro  $h$ .

$h$	T1-ANFIS MatLab	IT2-ANFIS M-KMA	IT2-ANFIS OSTR
1.0	0.257912	9.868407579	0.000300050
0.9		10.68296293	0.000256151
0.8		9.948764366	0.000196128
0.7		8.246696608	0.000274154
0.6		8.866349267	0.000272418
0.5		9.868407579	0.000285146
0.4		9.948764366	0.000223523
0.3		8.866349267	0.000202883
0.2		9.948764366	0.000167826
0.1		9.948764366	0.000321947

Las Figuras 30 y 31 ilustran la comparación entre la señal de referencia y las salidas de cada modelo, así como la distribución de los errores respectivos. En ellas se aprecia que tanto T1-ANFIS como IT2-ANFIS OSTR siguen la tendencia de la resistencia hidrodinámica, mientras que la implementación con M-KMA se aleja de la referencia. Estos resultados consolidan el papel del esquema OSTR como estrategia para reducción de tipo en modelos IT2-ANFIS cuando se trabaja con sistemas complejos y de múltiples entradas.

**Ejemplo 10:** Los datos pertenecen al problema de predicción de la resistencia hidrodinámica de un casco de yate, utilizando el conjunto de datos Yacht Hydrodynamics disponible en el re-

Tabla 20: RMSE de entrenamiento de las plataformas T1-ANFIS, IT2-ANFIS M-KMA e IT2-ANFIS para el Ejemplo 9, considerando diferentes valores manuales del parámetro  $h$ .

$h$	T1-ANFIS MatLab	IT2-ANFIS M-KMA	IT2-ANFIS OSTRÁ
1.0	0.604484	9.128943899	0.000340062
0.9		9.142407581	0.000300024
0.8		9.144625850	0.000164143
0.7		9.377985916	0.000243508
0.6		9.633329207	0.000359141
0.5		9.128943899	0.000427607
0.4		9.144625850	<b>0.000155810</b>
0.3		9.633329207	0.000589048
0.2		9.144625850	0.000375091
0.1		9.144625850	0.000355663

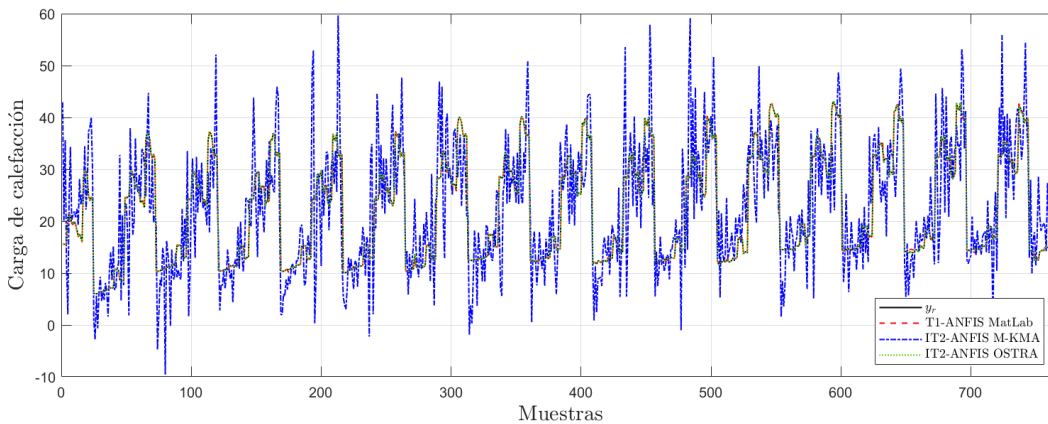


Figura 28: Aproximación obtenida con T1-ANFIS MatLab, IT2-ANFIS M-KMA e IT2-ANFIS OSTRÁ para el Ejemplo 8, en comparación con la referencia.

positorio UCI [86]. Este dataset se basa en simulaciones físicas y mediciones experimentales, donde la salida representa la resistencia hidrodinámica total al avance del yate expresada en kilonewtons (kN), mientras que las entradas corresponden a parámetros geométricos y físicos del casco (coeficiente prismático, relación de bloques, razón de área de secciones, etc.). Este escenario resulta especialmente relevante debido a la naturaleza no lineal y multivariable del problema, lo que lo convierte en un banco de pruebas adecuado para validar el desempeño de los modelos T1-ANFIS e IT2-ANFIS bajo condiciones realistas de ingeniería naval.

Los resultados de RMSE se presentan en la Tabla 21 donde, puede observarse que la plataforma IT2-ANFIS OSTRÁ alcanzó el mejor desempeño general, con un RMSE menor en varios órdenes de magnitud que las plataformas T1-ANFIS y IT2-ANFIS M-KMA. Esto confirma la capacidad del IT2-ANFIS OSTRÁ para gestionar la incertidumbre en problemas multiva-

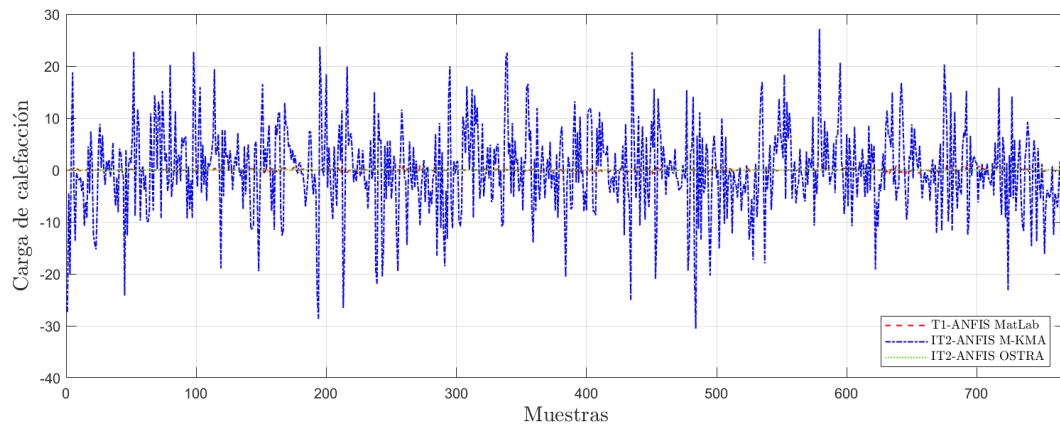


Figura 29: Error de entrenamiento de T1-ANFIS MatLab, IT2-ANFIS M-KMA e IT2-ANFIS OSTRÁ para el Ejemplo 8.

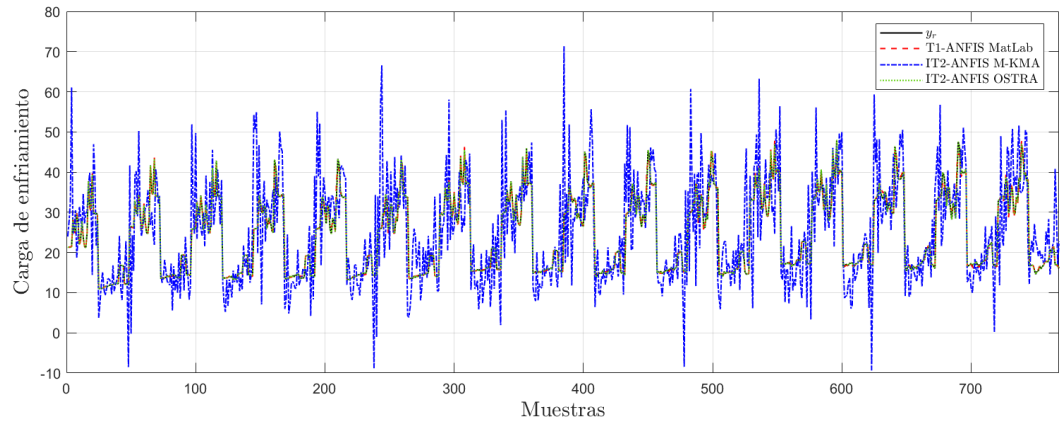


Figura 30: Aproximación obtenida con T1-ANFIS MatLab, IT2-ANFIS M-KMA e IT2-ANFIS OSTRÁ para el Ejemplo 9, en comparación con la referencia.

riables. No obstante en las Figuras 32 y 33, se aprecia que el método M-KMA evidenció una dispersión elevada del error, lo cual limita su aplicabilidad en este tipo de escenarios.

El análisis comparativo realizado a lo largo de los siete ejemplos (4-10) presentados en esta sección permitió observar el comportamiento relativo de las tres plataformas evaluadas. En dichos ejemplos, la estructura T1-ANFIS logró un desempeño superior únicamente en dos casos, mientras que el modelo IT2-ANFIS con reducción de tipo mediante OSTRÁ alcanzó los mejores resultados en cinco de ellos, superando tanto a T1-ANFIS como a IT2-ANFIS M-KMA. Este patrón de desempeño no sólo se mantiene, sino que se consolida al considerar la totalidad de los veintiún ejemplos de datos evaluados: T1-ANFIS resultó ganador en seis ocasiones, frente a quince en las que IT2-ANFIS OSTRÁ obtuvo el menor RMSE de entrenamiento.

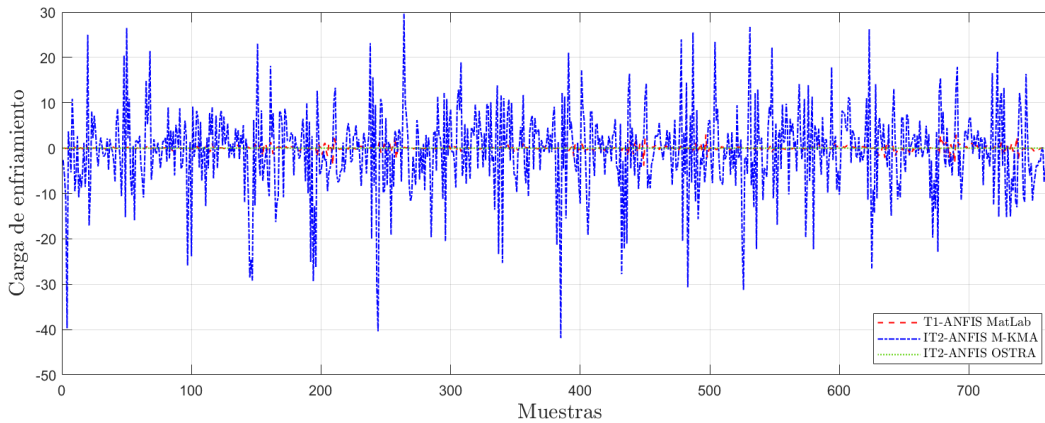


Figura 31: Error de entrenamiento de T1-ANFIS MatLab, IT2-ANFIS M-KMA e IT2-ANFIS OSTR para el Ejemplo 9.

Tabla 21: RMSE de entrenamiento de las plataformas T1-ANFIS, IT2-ANFIS M-KMA e IT2-ANFIS para el Ejemplo 10, considerando diferentes valores manuales del parámetro  $h$ .

$h$	T1-ANFIS MatLab	IT2-ANFIS M-KMA	IT2-ANFIS OSTR
1.0	0.276512	29.78580446	0.185555531
0.9		33.48263326	0.156514009
0.8		33.10589516	0.163226015
0.7		34.89030356	0.377188570
0.6		29.81693673	1.103683664
0.5		32.74548240	0.198786843
0.4		33.10589516	0.118481869
0.3		30.69965338	0.306494204
0.2		24.62100059	0.305634579
0.1		28.06163645	0.423058144

Estos hallazgos respaldan la propuesta de IT2-ANFIS OSTR como un esquema alternativo para el modelado de sistemas, que en quince de veintidós ejemplos obtuvo el menor RMSE de entrenamiento frente a su contraparte T1. Además, la evidencia acumulada muestra que la herramienta desarrollada no sólo opera en escenarios de prueba controlados, sino que también permite modelar sistemas dinámicos del mundo real. Lo anterior permite obtener modelos capaces de capturar la dinámica de procesos complejos, como el TR, y con ello sentar las bases para el diseño de un controlador difuso capaz de ejecutar tareas de seguimiento de trayectorias.

#### 4.1.4. Validación de IT2-ANFIS OSTR

Tras los experimentos preliminares presentados en la sección anterior, donde el parámetro  $h$  fue ajustado manualmente en intervalos fijos, se identificó la necesidad de incorporar un

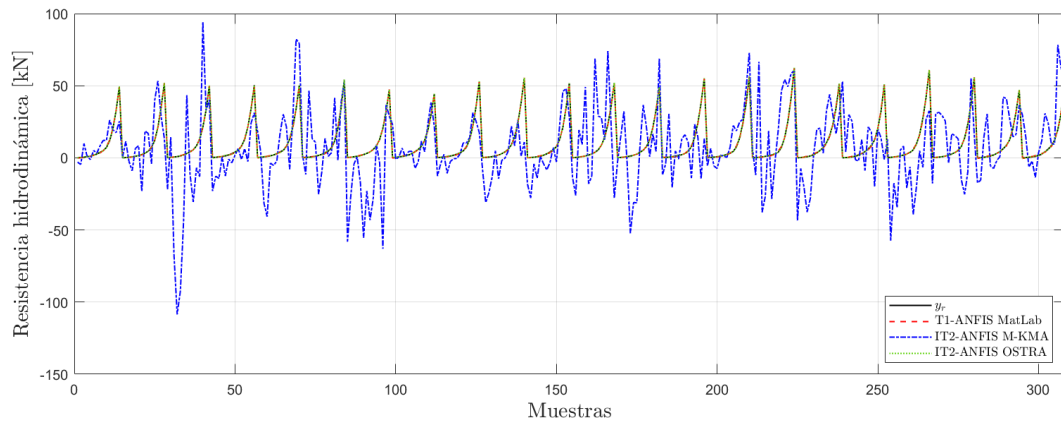


Figura 32: Aproximación obtenida con T1-ANFIS MatLab, IT2-ANFIS M-KMA e IT2-ANFIS OSTR para el Ejemplo 10, en comparación con la referencia.

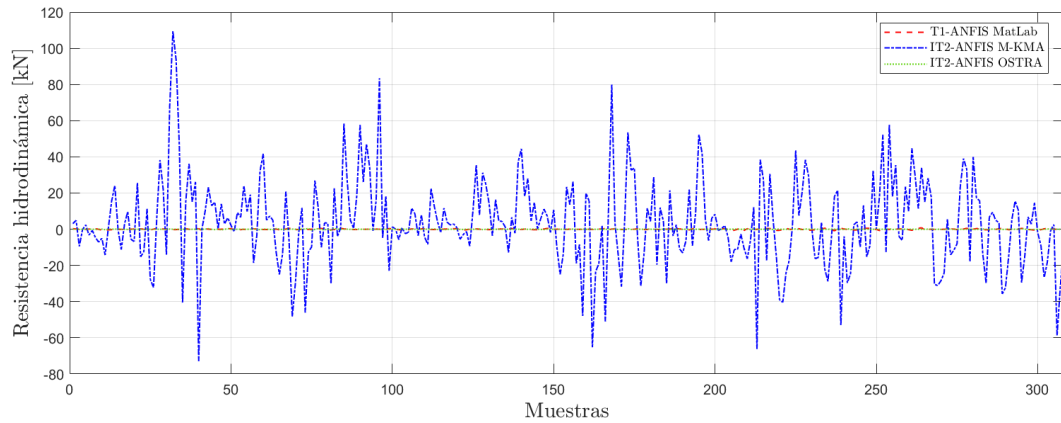


Figura 33: Error de entrenamiento de T1-ANFIS MatLab, IT2-ANFIS M-KMA e IT2-ANFIS OSTR para el Ejemplo 10.

mecanismo de optimización automática que permitiera definir la forma óptima del FOU en cada función de pertenencia intervalar. En esta etapa se introduce la versión consolidada del IT2-ANFIS OSTR, cuya principal innovación radica en la optimización del parámetro  $h_{j,k}$  (Ec. 17) durante la fase de entrenamiento, evitando la dependencia de un ajuste manual y aumentando la capacidad del modelo para adaptarse a distintas condiciones de los datos.

Con el fin de validar formalmente la propuesta, se realiza una comparación sistemática entre el IT2-ANFIS OSTR optimizado, un T1-ANFIS y un IT2-ANFIS, como se muestra en la Tabla 22. Para garantizar una evaluación robusta al igual que en los ejemplos anteriores, se emplean dos GBSMF por variable de entrada y un número fijo de 100 épocas de entrenamiento en cada uno de los ejemplos.

Tabla 22: Comparación de estructuras ANFIS y algoritmos de optimización utilizados.

Estructura	Optimización de premisas	Optimización de consecuentes
T1-ANFIS (MatLab)	BP	LSE
ANFIS2 Fuzzy	BP	BP
IT2-ANFIS OSTR	BP	KF

### Datos sintéticos

**Ejemplo 11:** Con el objetivo de realizar una comparación con el sistema de modelado proporcionado en [30], tomamos su ejemplo que es la siguiente planta no lineal de segundo orden variable en el tiempo:

$$y_r(k) = \frac{y_r(k-1)y_r(k-2)y_r(k-3)u(k-1)(y_r(k-3) - b(k)) + c(k)u(k)}{a(k) + y_r(k-2)^2 + y_r(k-3)^2}, \quad (57)$$

donde  $y_r(k-1)$ ,  $y_r(k-2)$ , e  $y_r(k-3)$  son salidas retardadas de primer, segundo y tercer orden de la planta,  $u(k)$  y  $u(k-1)$  son la señal de control y un retardo de la misma respectivamente. Los parámetros variables en el tiempo  $a(k)$ ,  $b(k)$  y  $c(k)$  están dados por:

$$a(k) = 1.2 - 0.2 \cos\left(\frac{2\pi k}{T}\right); \quad b(k) = 1 - 0.4 \sin\left(\frac{2\pi k}{T}\right); \quad c(k) = 1 + 0.4 \sin\left(\frac{2\pi k}{T}\right); \quad (58)$$

donde  $T = 1000$ . La señal de control  $u(k)$  se elige como sigue:

$$u(k) = \begin{cases} \sin\left(\frac{\pi k}{25}\right), & k < 250, \\ 1, & 250 \leq k < 500, \\ -1, & 500 \leq k < 750, \\ 0.3 \sin\left(\frac{\pi k}{25}\right) + 0.1 \sin\left(\frac{\pi k}{32}\right) + 0.6 \sin\left(\frac{\pi k}{10}\right), & 750 \leq k < 1000. \end{cases} \quad (59)$$

En este ejemplo  $u(k)$ ,  $k = 1, \dots, 1000$  alimenta al sistema dinámico no lineal Ec. (57) y se obtienen 1000 muestras de  $y_r(k)$ . La Figura 34 muestra los resultados de identificación de las estructuras T1-ANFIS e IT2-ANFIS OSTR, dado que el error es similar, no se aprecian diferencias. Por otro lado, en la Figura 35 se muestra el error entre la señal de referencia y el modelo de cada estructura. Se puede observar que IT2-ANFIS OSTR tiene un error menor que T1-ANFIS.

Los resultados obtenidos con T1-ANFIS e IT2-ANFIS OSTR se compararon con otras tres plataformas simuladas numéricamente tomadas de [30], tal y como se muestra en la Tabla 23, donde se puede observar que IT2-ANFIS OSTR obtiene un error menor respecto al conjunto de plataformas.

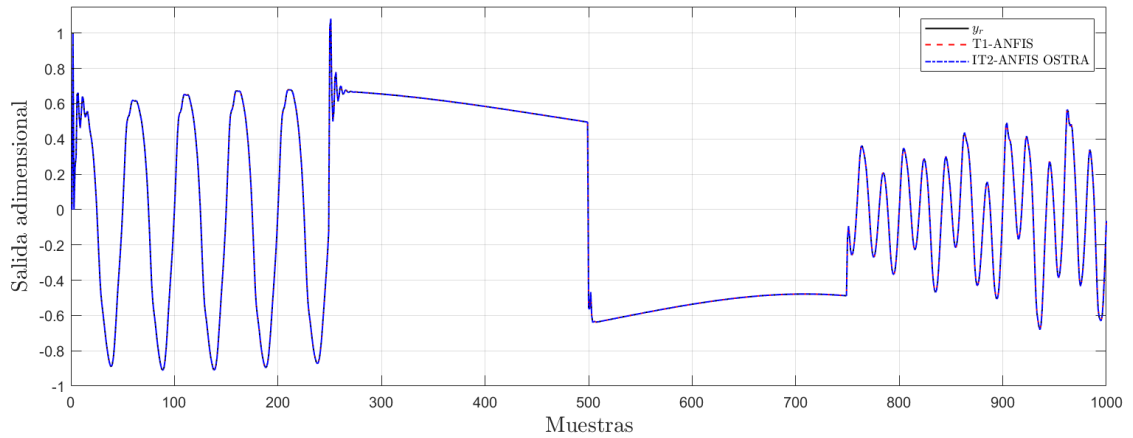


Figura 34: Aproximación del modelo T1-ANFIS vs IT2-ANFIS OSTR a la salida real del Ejemplo 11.

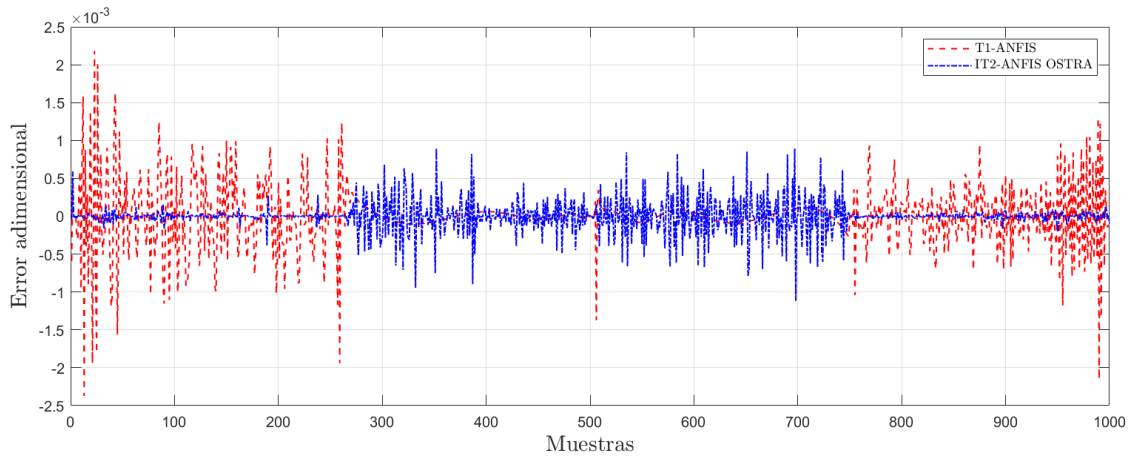


Figura 35: Error ( $y_r - y_m$ ) del modelo T1-ANFIS vs IT2-ANFIS OSTR para el Ejemplo 11.

Tabla 23: Resultados RMSE de los modelos T1-ANFIS vs IT2-ANFIS para el Ejemplo 11.

T1-ANFIS	IT2-ANFIS OSTR	Método [87]	Método [88]	ANFIS2 Fuzzy [30]
0.000396	0.000192	0.0253	0.0042	0.0009

**Ejemplo 12:** Tomado de [21], este ejemplo corresponde a la ecuación no lineal  $y_r = 1 + x_1^{0,5} + x_2^{-1} + x_3^{-1,5}$  con tres entradas  $x_1, x_2, x_3 \in [1, 6]$  y una salida  $y_r$ . El ejemplo se compone de 216 muestras de entrenamiento, dentro de las cuales se introdujo en la muestra 100 un efecto de desconexión momentánea del sensor con el propósito de evaluar la robustez de los modelos frente a perturbaciones abruptas. A diferencia de lo realizado en el Ejemplo 6 donde el parámetro  $h$  fue ajustado manualmente, en este caso se emplea el IT2-ANFIS OSTR

con optimización automática de  $h$ , lo que permite ajustar dinámicamente la forma del FOU durante el proceso de entrenamiento.

La Figura 36 muestra los resultados de identificación de las estructuras T1-ANFIS e IT2-ANFIS OSTRÁ. Puede observarse que la señal del modelo obtenido con IT2-ANFIS OSTRÁ tiene un menor sobreimpulso aproximando la señal de referencia, incluso en la muestra 100 cuando ocurre la perturbación brusca, evidenciando una respuesta rápida y precisa. Asimismo, la Figura 37 compara el error  $e = y_r - y_m$ , entre la señal de salida  $y_r$  y la salida de los modelos  $y_m$  de cada estructura. Finalmente la Tabla 24, muestra el RMSE obtenido con cada estructura, confirmando que IT2-ANFIS OSTRÁ obtiene mejores resultados que T1-ANFIS, superando incluso el resultado obtenido en el Ejemplo 6. Lo anterior demuestra la mejora que proporciona la optimización automática del parámetro  $h$ .

Tabla 24: Resultados RMSE de los modelos T1-ANFIS vs IT2-ANFIS para el Ejemplo 12.

T1-ANFIS	IT2-ANFIS OSTRÁ
0.30819	0.01532

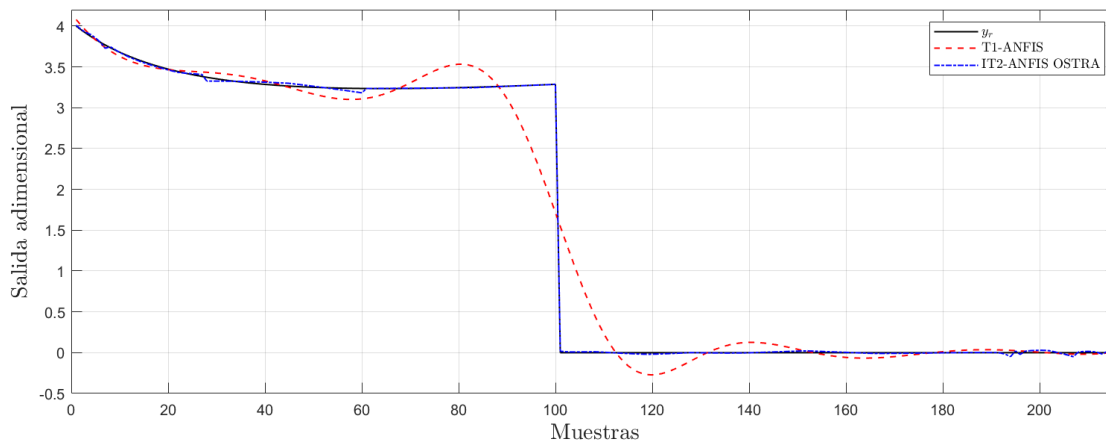


Figura 36: Aproximación del modelo T1-ANFIS vs IT2-ANFIS OSTRÁ a la salida real del Ejemplo 12.

### Datos reales del sistema de volante

**Ejemplo 13:** El seguimiento de trayectorias con robots móviles en campos de cultivo es un reto debido a la naturaleza variable del entorno. Los campos de cultivo suelen tener obstáculos, como árboles, postes y vallas, y el suelo puede ser irregular e inestable [89]. Entonces, la obtención de un modelo robusto, a pesar de las perturbaciones del entorno, es necesario.

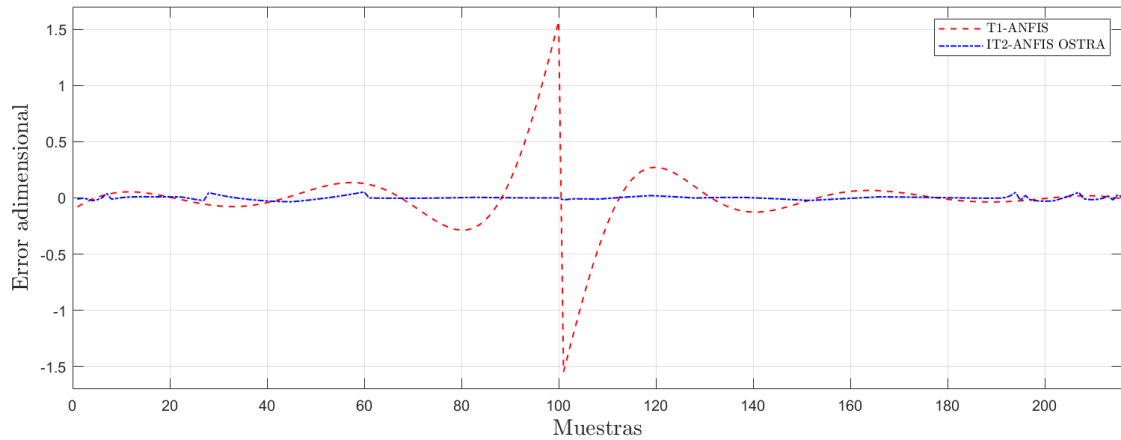


Figura 37: Error ( $y_r - y_m$ ) del modelo T1-ANFIS vs IT2-ANFIS OSTRÁ para el Ejemplo 12.

En este trabajo, se propone un modelo de un sistema de volante del TR para poner a prueba nuestro esquema de modelado. El modelado y control del volante ha sido abordado por diferentes autores utilizando técnicas de T1FLS. También, en [46, 90] los subsistemas dinámicos del volante y la velocidad se modelan utilizando redes neuronales difusas de T2, para lograr el seguimiento de trayectorias de TRs.

Para este ejemplo, se utiliza el TR automatizado para obtener los datos de E/S de su sistema de volante. El TR se utilizó en una versión previa con respecto a la actual, teniendo como arquitectura modular los microcontroladores Arduino que se describen en la sección A.1.1. El TR dispone de un servomecanismo para el movimiento del ángulo de giro del volante. En concreto, los datos se obtienen mediante un sensor codificador LPD3806 600BM-G5-24C de 600 pulsos por revolución, acoplado a un mecanismo de cremallera y piñón donde se mide el ángulo de dirección. Para el movimiento del volante se utiliza un motor de corriente continua de 24 V de la marca SHINANO KENSHI, con un par de 1.5 Nm y un máximo de 300 revoluciones por minuto, así como un puente en H 2x30 de la marca Sabertooth. La señal de entrada  $u(k)$  tiene valores entre 1 y 127, donde 1 es un giro a máxima velocidad en sentido negativo y 127 en sentido positivo, como se muestra en la Figura 39. La señal  $y_r(k)$  es un retardo unitario de la salida  $y_r(k + 1)$  de la planta. El sistema de adquisición de datos está integrado en microcontrolador Arduino Uno con un tiempo de muestreo (TM) de 20 ms.

Se obtuvo un conjunto de datos de 811 muestras de E/S para llevar a cabo la identificación mediante los métodos T1-ANFIS e IT2-ANFIS OSTRÁ. La Figura 40 muestra los resultados de aproximación obtenidos para T1-ANFIS e IT2-ANFIS OSTRÁ. Del mismo modo, la Figura 41 muestra el error de cada modelo con respecto a la salida real de la planta. Los resultados de IT2-ANFIS OSTRÁ son menores en comparación con T1-ANFIS por milésimas de grado de la posición del volante, como se observa en la Tabla 25.

Utilizando el modelo obtenido por IT2-ANFIS OSTRÁ programado en el microcontrolador,

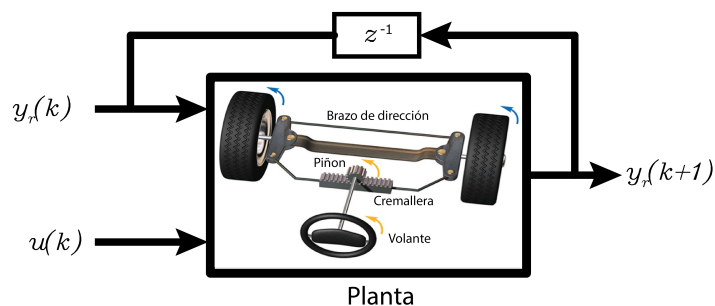


Figura 38: Estructura del sistema de volante del TR, con 2 entradas y 1 salida.

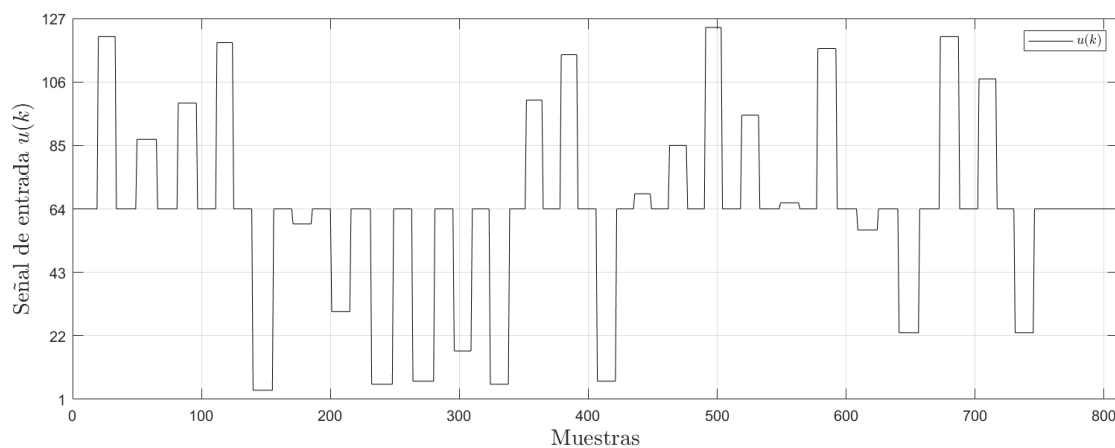


Figura 39: Señal de entrada  $u(k)$  para entrenamiento.

Tabla 25: Resultados RMSE de los modelos T1-ANFIS vs IT2-ANFIS para el Ejemplo 13.

T1-ANFIS	IT2-ANFIS OSTR
0.319667	0.31769

con los parámetros antecedentes y consecuentes óptimos mostrados en la Tabla 26, realizamos dos pruebas adicionales en tiempo real para verificar la eficacia del modelo. Para ambas pruebas, se utilizó la misma señal de entrada  $u(k)$  con un conjunto de datos diferente para una validación cruzada, como se muestra en la Figura 42. La señal de control se alimentó a la planta y al modelo, mientras que la salida obtenida de la planta se realimenta al modelo. Además, se utilizaron dos terrenos diferentes para que el modelo estuviera expuesto a perturbaciones desconocidas. La prueba No.1 se realizó con el TR parado y sobre una superficie plana, como se muestra en la Figura 43. Para la prueba No.2, el tractor estaba en una superficie de tierra y en movimiento, para generar una diferencia entre los datos de la superficie plana, como se muestra en la Figura 44. Los resultados de la prueba en terreno llano y de tierra se muestran en las Figuras 43 y 44 respectivamente.

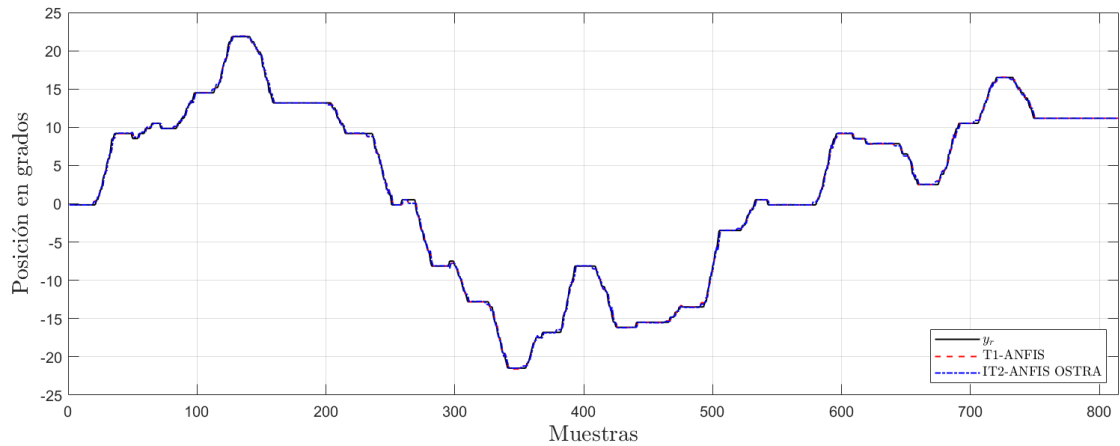


Figura 40: Aproximación del modelo T1-ANFIS vs IT2-ANFIS OSTR a la salida real del Ejemplo 13.

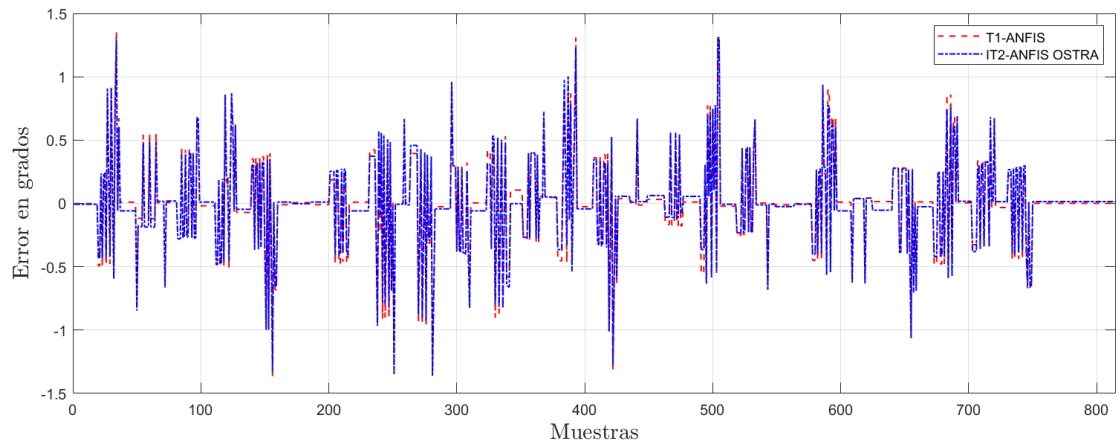


Figura 41: Error ( $y_r - y_m$ ) del modelo T1-ANFIS vs IT2-ANFIS OSTR para el Ejemplo 13.

De ambas pruebas se observa que el modelo IT2-ANFIS OSTR mantiene un RMSE similar (0.3084 y 0.3088) incluso con datos de entrada no vistos durante el entrenamiento y cuando hay perturbaciones que no se encontraron en el conjunto de datos de entrenamiento. Una comparación del rendimiento del OSTR con otros tres algoritmos: EIASC, KMA y EKMA, se presenta en la Tabla 27. Utilizando los parámetros del modelo mostrados en la Tabla 26, se pretende validar el IT2FLS utilizando los datos de la Prueba No.1 (Figura 43) y No.2 (Figura 44), pero con cada uno de los algoritmos mencionados en el paso de reducción de tipo. El objetivo de la prueba es evaluar la eficacia de OSTR en la reducción del error RMSE y del tiempo de cálculo en IT2FLS. Se puede observar que el algoritmo OSTR tiene el RMSE más bajo en ambas pruebas. En la prueba No.1, el menor tiempo es de OSTR y en la prueba No.2 es del algoritmo EIASC. Los experimentos se llevaron a cabo en un procesador Intel Core i9-12900H con 16 GB de RAM y en el lenguaje de programación Python 3.11.5.

Tabla 26: Parámetros premisa y consecuentes del modelo obtenido con IT2-ANFIS OSTRa para el Ejemplo 13.

FS	MF	Parámetros premisa					Parámetros consecuentes			
		$a$	$\bar{a}$	$b$	$c$	$h$	$C_{i,j}$	$j = 1, \dots, J + 1$ ; Inputs		
$\tilde{A}_{1,1}$	$\underline{A}_{1,1}$	8.6449	-	1.1764	-21.4879	0.1	$i = 1, \dots, N$ ; Rules	0.9846	0.0090	-1.0204
	$\bar{A}_{1,1}$	-	10.3708			1				
$\tilde{A}_{1,2}$	$\underline{A}_{1,2}$	8.6754	-	0.8790	21.8428	0.6583		0.9861	0.0018	-0.4040
	$\bar{A}_{1,2}$	-	10.4218			1				
$\tilde{A}_{2,1}$	$\underline{A}_{2,1}$	23.9982	-	1.0303	4.0002	0.1189		1.0076	0.0046	-0.2355
	$\bar{A}_{2,1}$	-	28.8013			1				
$\tilde{A}_{2,2}$	$\underline{A}_{2,2}$	23.9984	-	1.0851	124.0004	0.1		0.9850	0.0127	-0.5921
	$\bar{A}_{2,2}$	-	28.7957			1				

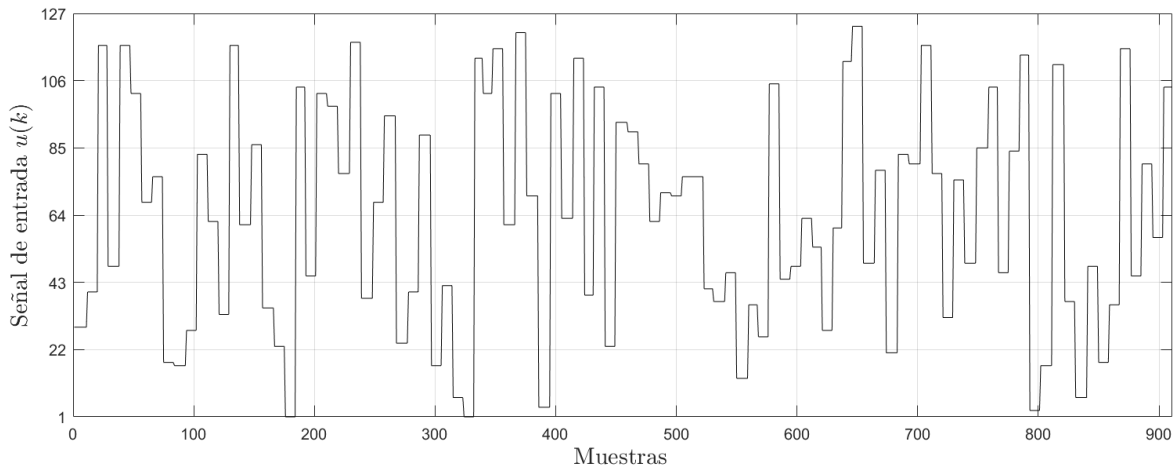


Figura 42: Señal de entrada  $u(k)$  para validación.

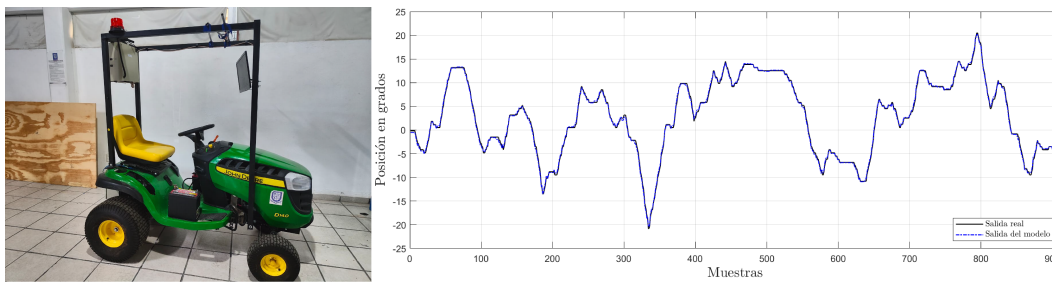


Figura 43: Prueba No.1 el TR en un terreno plano, resultados del modelo frente al sistema real con RMSE=0.3084.

**Prueba de reducción de ruido:** Para comprobar la propiedad de reducción de ruido inhe-

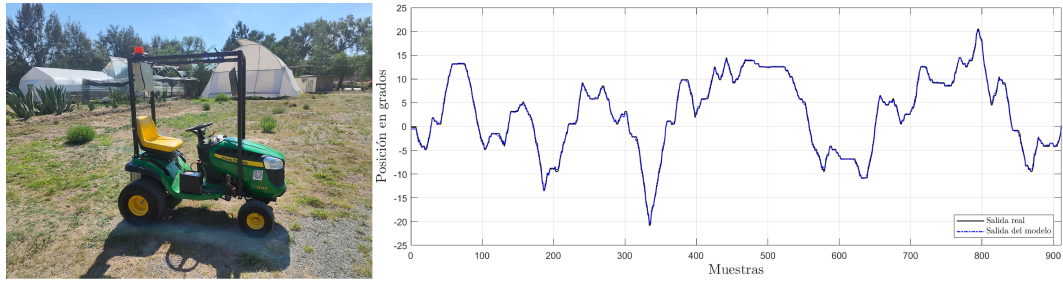


Figura 44: Prueba No.2 el TR en un camino de tierra, resultados del modelo frente al sistema real con RMSE=0.3088.

Tabla 27: Comparación del RMSE y tiempo de cálculo de los TRA del Ejemplo 13.

Prueba	OSTRA		EIASC		KMA		EKMA	
	RMSE	Tiempo	RMSE	Tiempo	RMSE	Tiempo	RMSE	Tiempo
No.1	0.3084	0.05	0.4694	0.06	0.4694	0.1	0.4743	0.06
No.2	0.3088	0.05	0.6095	0.04	0.6095	0.09	0.6170	0.06

rente a IT2FLS, utilizando una señal  $u(k)$  con ruido blanco gaussiano como la mostrada en la Figura 45, se obtiene un modelo utilizando IT2-ANFIS OSTRA. La señal de entrada  $y(k)$  es la misma que la utilizada para obtener el modelo descrito en la Tabla 26. El RMSE de identificación obtenido es de 0.3193, muy similar al obtenido utilizando una señal de entrada  $u(k)$  sin ruido como la mostrada en la Tabla 41. En la Figura 46 podemos ver como la adición de ruido a la entrada  $u(k)$  no afecta considerablemente a la señal de salida  $y(k+1)$ , y la diferencia en RMSE es de 0.0016. Por este motivo, y como se ha comentado anteriormente con las pruebas No.1 y No.2, el modelo obtenido con IT2-ANFIS OSTRA es capaz de responder a perturbaciones no incluidas en los datos de entrenamiento y también es capaz de reducir el ruido presente en los datos de entrada.

Los resultados experimentales obtenidos muestran que IT2-ANFIS OSTRA puede lograr un menor RMSE que el método tradicional T1-ANFIS y otras estructuras IT2-ANFIS de la literatura, en caso de implementación real. El IT2-ANFIS OSTRA propuesto es un enfoque para modelar sistemas dinámicos representados con conjuntos de datos de E/S. IT2-ANFIS OSTRA optimiza los parámetros antecedentes de cada MF del IT2FLS, incluido el parámetro  $h$  que representa la medida de incertidumbre y determina la amplitud del FOU. Además, el modelo obtenido con IT2-ANFIS OSTRA, es capaz de responder a diferentes datos de entrada y perturbaciones no presentes en los datos de entrenamiento, mostrando que permite modelar la incertidumbre inherente a los sistemas complejos. El IT2-ANFIS OSTRA propuesto elimina el proceso iterativo en el paso de reducción de tipo, para calcular el centroide de un T2FS, lo que resulta en un menor tiempo de entrenamiento y un menor costo computacional. Además,

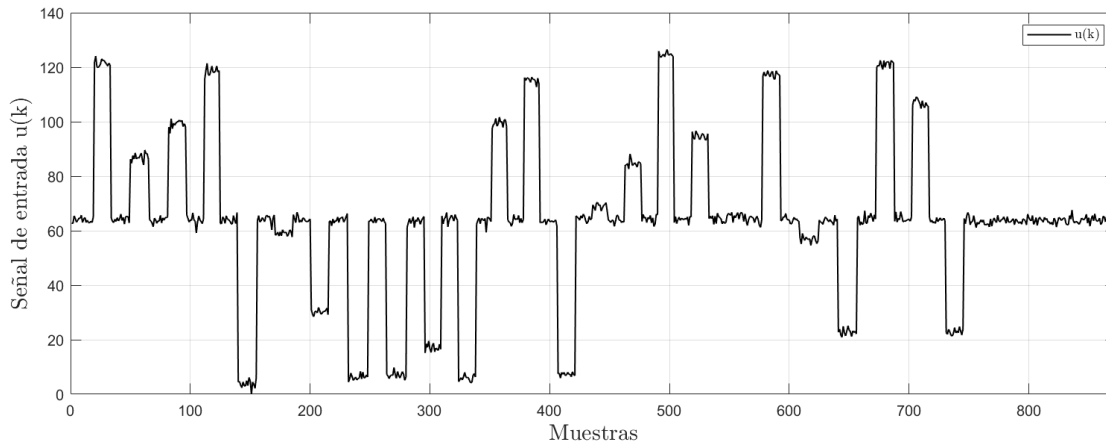


Figura 45: Señal de entrada  $u(k)$  con ruido blanco gaussiano.

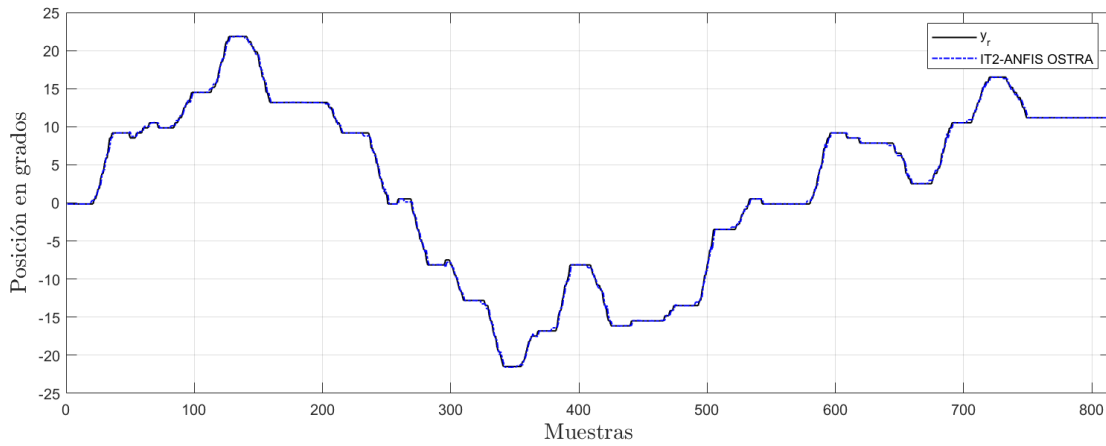


Figura 46: Aproximación de IT2-ANFIS OSTRa a la señal de salida del Ejemplo 13 con ruido blanco gaussiano en  $u(k)$ .

el modelo obtenido con el IT2-ANFIS OSTRa se puede programar en un microcontrolador de bajo costo, permitiendo su aplicación efectiva en sistemas como el TR.

## 4.2. Seguimiento de trayectorias

En esta sección se describe en detalle el proceso llevado a cabo para lograr el seguimiento de trayectorias del TR en diferentes rutas. Inicialmente, se realizaron pruebas con trayectorias en línea recta con el objetivo de verificar la funcionalidad del PPA embebido en el TR y la integración efectiva de los módulos que lo componen, como se observa en la Figura 47. Una vez que se comprobó la funcionalidad de estos módulos, se procedió a realizar pruebas en una ruta con tres surcos de 2 m de radio (Figura 48). Estas pruebas preliminares se llevaron a cabo utilizando el GLC y el EBC, dado que en ese momento no se había definido la imple-

mentación de los conjuntos difusos IT2 ni de la plataforma IT2-ANFIS OSTRAL. Esta etapa inicial sentó las bases operativas para evaluar el seguimiento de trayectorias en entorno real.

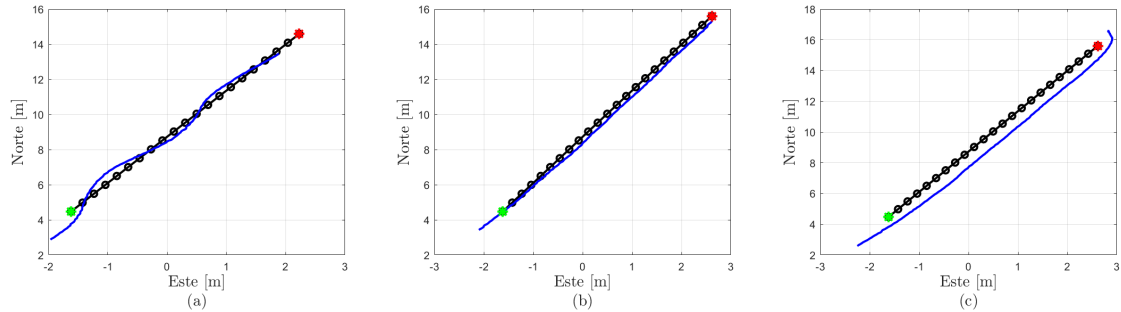


Figura 47: Seguimiento de trayectorias en línea recta utilizando el GLC; (a) con distancia de anticipación de 0.5 m, (b) de 2 m y (c) de 3 m.

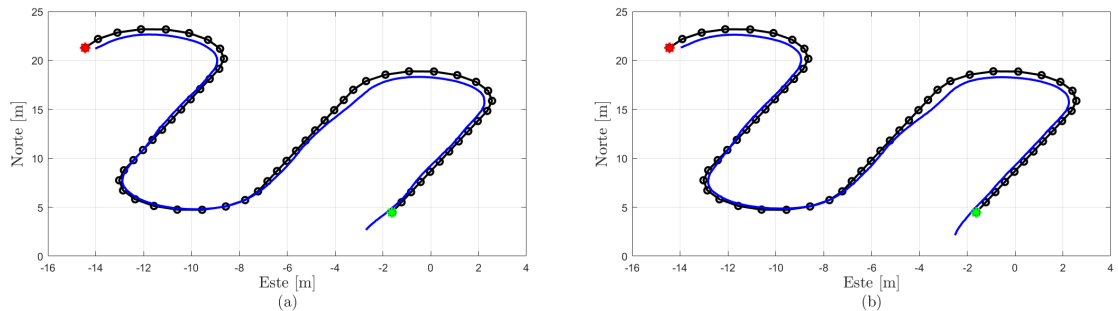


Figura 48: Seguimiento de trayectorias en tres surcos con radio 2 m; (a) utilizando el GLC con distancia de anticipación de 2 m y (b) utilizando el controlador BOE con distancia de anticipación de 2 m.

En la Figura 47, se observa que la variación en la distancia de anticipación  $l_a$  afecta los resultados que se obtienen. Por este motivo, se seleccionó  $l_a=2$  m para realizar todas las pruebas que se desarrollan en este trabajo de tesis. Como se mencionó anteriormente, las pruebas del TR en línea recta permitieron definir conceptos importantes que sentaron la base para realizar el seguimiento de trayectorias a través de rutas con mayor complejidad. Dicho esto, en la Figura 48 se comparan los métodos de control GLC y EBC, para llevar a cabo una ruta en forma de surcos. Utilizando el GLC se obtiene un RMSE del error lateral de 0.649 m, mientras que con el EBC se obtiene un RMSE de 0.654 m. Los resultados obtenidos en esta fase proporcionan una base importante para el análisis comparativo posterior y permiten debatir las limitaciones y los desafíos que se enfrentaron con los métodos de control iniciales. Además, estas pruebas permitieron evaluar la precisión de los sistemas de control y la capacidad del TR para mantener una trayectoria.

Como parte de la exploración metodológica para obtener una estrategia que empleara IT2-ANFIS OSTRA en el modelado directo del comportamiento cinemático del TR, se evaluó un enfoque alternativo de identificación de sistema. Tras probar diversas combinaciones de variables de entrada, la configuración de mayor desempeño es la que se describe a continuación. La idea consistió en entrenar dos IT2FLS independientes usando IT2-ANFIS OSTRA: el primero utilizando como datos de entrada  $x(k)$ ,  $\theta(k)$ ,  $v(k)$  y  $\delta(k)$ , con el objetivo de predecir la salida  $x(k+1)$ ; el segundo modelo utilizó  $y(k)$ ,  $\theta(k)$ ,  $v(k)$  y  $\delta(k)$ , para predecir la salida  $y(k+1)$ . Los datos para entrenar ambos modelos fueron tomados de una prueba real en la que el TR fue conducido manualmente, como se observa en la Figura 49. El objetivo de estos dos modelos era ser capaces de describir una trayectoria con el menor error lateral posible, aproximando así las condiciones reales y logrando modelar el comportamiento del TR.

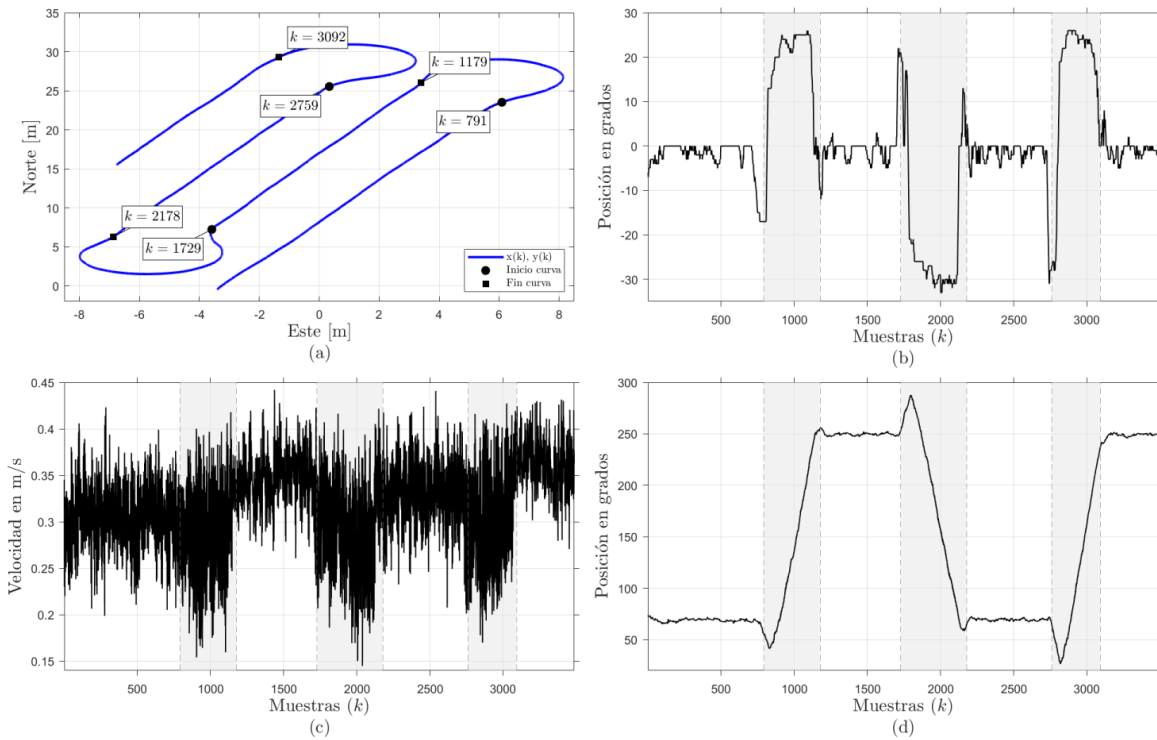


Figura 49: Variables cinemáticas registradas durante una trayectoria manual para el entrenamiento de los modelos difusos. (a) Trayectoria con identificadores de muestra ( $k$ ) para el inicio y fin de las curvas. (b) Ángulo de dirección del volante  $\delta(k)$ . (c) Velocidad longitudinal del vehículo  $v(k)$ . (d) Orientación del TR  $\theta(k)$ . Las regiones sombreadas establecen la correspondencia exacta con las curvas delimitadas en (a).

La forma de validar este enfoque consistió en proporcionar como entrada inicial a los modelos todas las variables antes mencionadas, de una ruta real diferente (Figura 50). A partir del segundo evento, el modelo solo recibe la velocidad  $v(k)$  y el ángulo del volante  $\delta(k)$  de

los datos reales, mientras que las variables  $x(k)$  e  $y(k)$  son retroalimentadas a partir de las salidas de cada modelo con un retardo. La orientación  $\theta(k)$  se calcula utilizando la siguiente ecuación:

$$\theta(k) = \arctan2(y(k) - y(k-1), x(k) - x(k-1)), \quad (60)$$

donde  $\arctan2$  es la función arcotangente de dos parámetros, que calcula el ángulo considerando el signo del numerador y denominador para determinar el cuadrante correcto del ángulo resultante.

Se encontraron varias limitaciones con este enfoque. En primer lugar, el modelo no es capaz de predecir más allá del universo de discurso de las variables con las que fue entrenado, lo cual representó un problema principalmente para  $x$  e  $y$ , ya que estas pueden variar dependiendo de la ubicación del TR con respecto a la estación base. Las demás variables, como la velocidad, el ángulo del volante y la orientación, tenían valores máximos y mínimos conocidos, lo que facilitaba su gestión. Además, lograr que el modelo generalice con precisión el comportamiento real del TR es un problema de identificación no lineal con alta sensibilidad a perturbaciones [21, 52], dado que los errores de predicción en cada paso se acumulan al retroalimentarse, generando una deriva creciente que degrada la predicción de la ruta completa. Se exploraron diversas arquitecturas durante este proceso, incluyendo distintas combinaciones de variables de entrada y salida ( $x$ ,  $y$ ,  $\theta$ ,  $v$ ,  $\delta$  y sus variantes en grados y radianes), diferentes rutas de entrenamiento, y estructuras con una o dos salidas simultáneas. Como resultado, ninguna de las configuraciones evaluadas logró predecir una ruta completa únicamente a partir de la velocidad y el ángulo del volante en un esquema de retroalimentación autónoma.

Avanzando con el objetivo de implementar FLC utilizando ANFIS, se entrenaron tanto controladores con T1-ANFIS como con IT2-ANFIS OSTR. Se determinó que la manera de entrenar estos controladores era utilizando el error de cabeceo ( $e_\theta$ ) y su derivada ( $\dot{e}_\theta$ ) como variables de entrada, para generar como salida el ángulo del volante ( $\delta$ ). La elección de estas variables se fundamenta en la similitud con un controlador PD clásico, ya que el error de cabeceo actúa como una señal proporcional y su derivada actúa como el término diferencial ajustando la velocidad de corrección, proporcionando así una acción de control directa. Otro aspecto importante es que, a diferencia del enfoque exploratorio descrito anteriormente, las variables seleccionadas en este caso (error de cabeceo y su derivada) tienen un universo de discurso acotado, lo cual elimina la incertidumbre en el entrenamiento del modelo y facilita la generalización. Además, el uso de unidades consistentes (todos los valores en grados) evita problemas de escala y asegura una coherencia en el procesamiento de las señales, lo cual es necesario para el desempeño del sistema de control.

La inclusión de estos FLC permite generar una señal de control con mayor capacidad de manejo de la incertidumbre presente en el desplazamiento físico del TR, en comparación con los controladores clásicos de referencia. Una vez entrenados los controladores difusos, se procedió a aplicar y validar directamente la metodología en el entorno físico real. Para lo

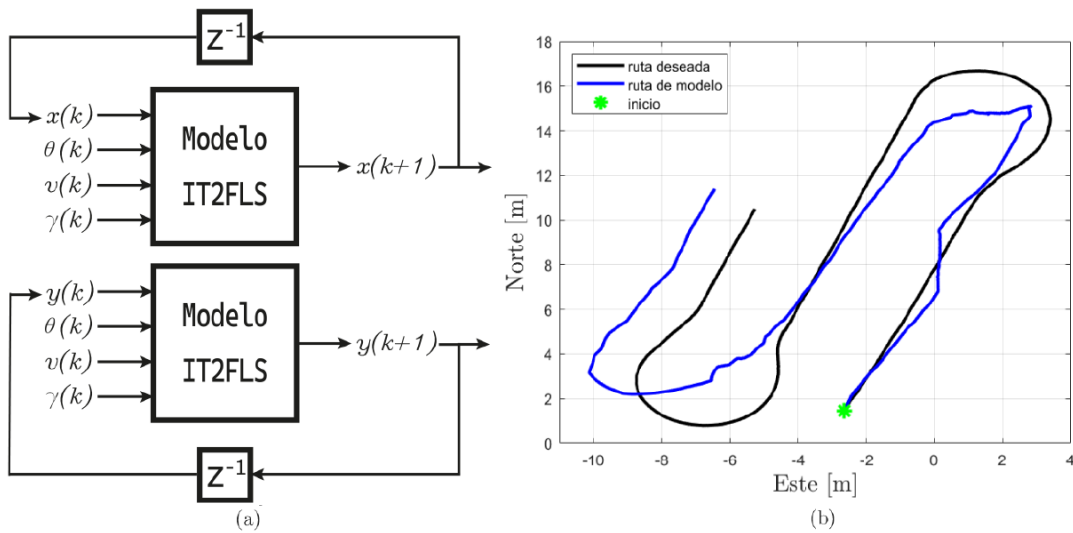


Figura 50: Estructura de los modelos difusos y prueba de validación; (a) estructura propuesta para los IT2FLS utilizados para predecir  $x(k + 1)$  e  $y(k + 1)$  a partir de las variables de entrada y (b) comparación entre la ruta deseada y la ruta obtenida con el modelo propuesto durante la prueba de validación.

anterior, se realizó un recorrido de inducción en la ruta básica de tres surcos con radio de 2 m, obteniendo los datos base que se muestran en la Figura 51.

Tabla 28: Resultados RMSE del modelado con T1-ANFIS vs IT2-ANFIS para los datos de la Figura 51.

Error	T1-ANFIS	IT2-ANFIS OSTR
RMSE del ángulo de dirección	1.27°	1.24°

Los resultados de los RMSE obtenidos con cada plataforma de modelado se muestran en la Tabla 28, donde se observa que la diferencia del RMSE del ángulo de dirección entre T1-ANFIS e IT2-ANFIS OSTR es de  $0.03^\circ$ . Para validar la efectividad de los controladores difusos, se llevaron a cabo un total de ocho pruebas: cuatro en rutas rectas (Figura 52) y cuatro en rutas con surcos (Figura 54). En estas pruebas, se utilizaron los siguientes métodos de control: EBC, GLC, T1FLC e IT2FLC. Los resultados obtenidos demuestran que, en las pruebas en línea recta el IT2FLC presentó el menor RMSE tanto en el error lateral como en el error de cabeceo, como se muestra en la Tabla 29. Sin embargo, en las pruebas realizadas en rutas con surcos, a pesar de que GLC, T1FLC e IT2FLC mostraron resultados similares (Tabla 30), el GLC obtuvo el menor error.

En la trayectoria de línea recta (Figura 53), se observa que el IT2FLC presenta menor error

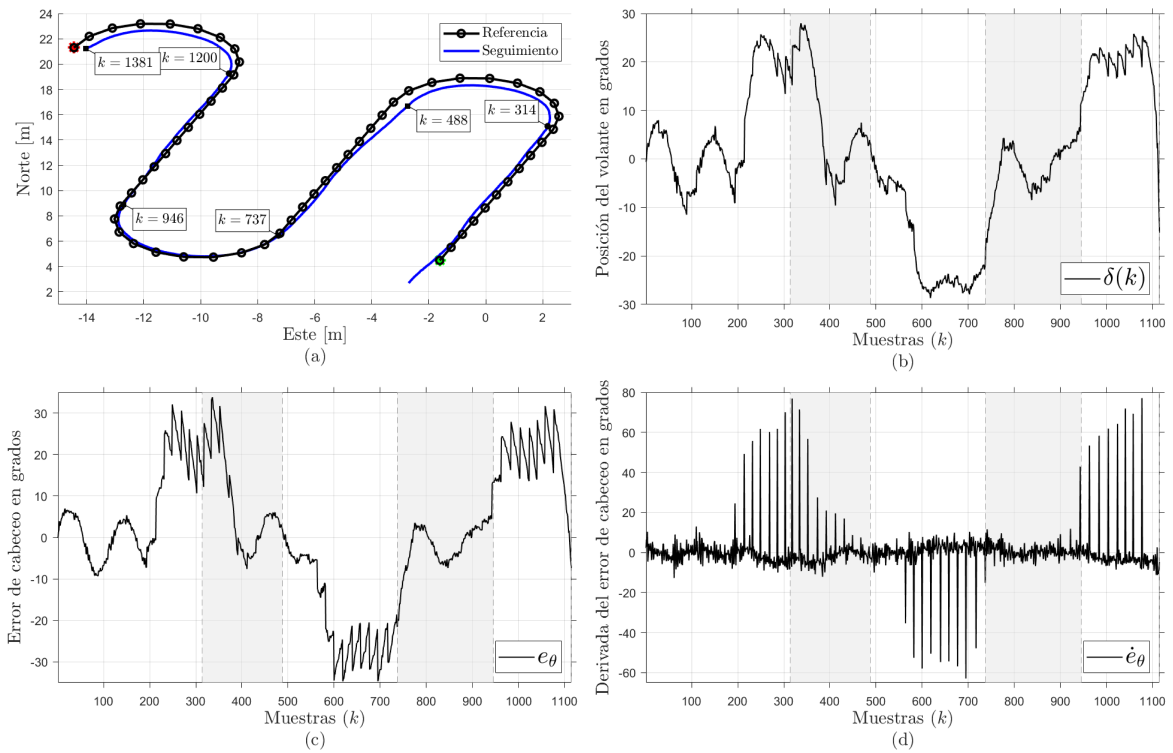


Figura 51: Conjunto de datos del seguimiento de trayectoria mediante el GLC para el modelado con ANFIS. (a) Referencia y trayectoria con identificadores de muestra  $k$  para el inicio y fin de las curvas. (b) Señal de control del ángulo del volante  $\delta(k)$ . (c) Evolución temporal del error de cabeceo  $e_\theta$ . (d) Evolución de la derivada del error de cabeceo  $\dot{e}_\theta$ . Las regiones sombreadas establecen la correspondencia exacta con las curvas delimitadas en (a).

en comparación con el T1FLC. Asimismo, la señal de posición del volante generada por el T1FLC registra picos de mayor amplitud, mientras que el IT2FLC produce una señal con menores variaciones. Por su parte, en el seguimiento del surco con radio de 2 metros (Figura 55), el desempeño de ambos controladores es similar; el RMSE de posición se mantiene equivalente para ambos sistemas y solo se registra una variación en el error de cabeceo. Este comportamiento equiparable puede atribuirse a que dicha trayectoria corresponde a la ruta utilizada para obtener los datos de entrenamiento.

Tabla 29: Comparativa de RMSE del error lateral y de cabeceo de diferentes controladores para el seguimiento de una línea recta.

Errores	EBC	GLC	T1FLC 2MF	IT2FLC 2MF
RMSE error lateral	0.117 m	0.23 m	0.119 m	0.094 m
RMSE error de cabeceo	3.13°	5.88°	3.07°	2.49°

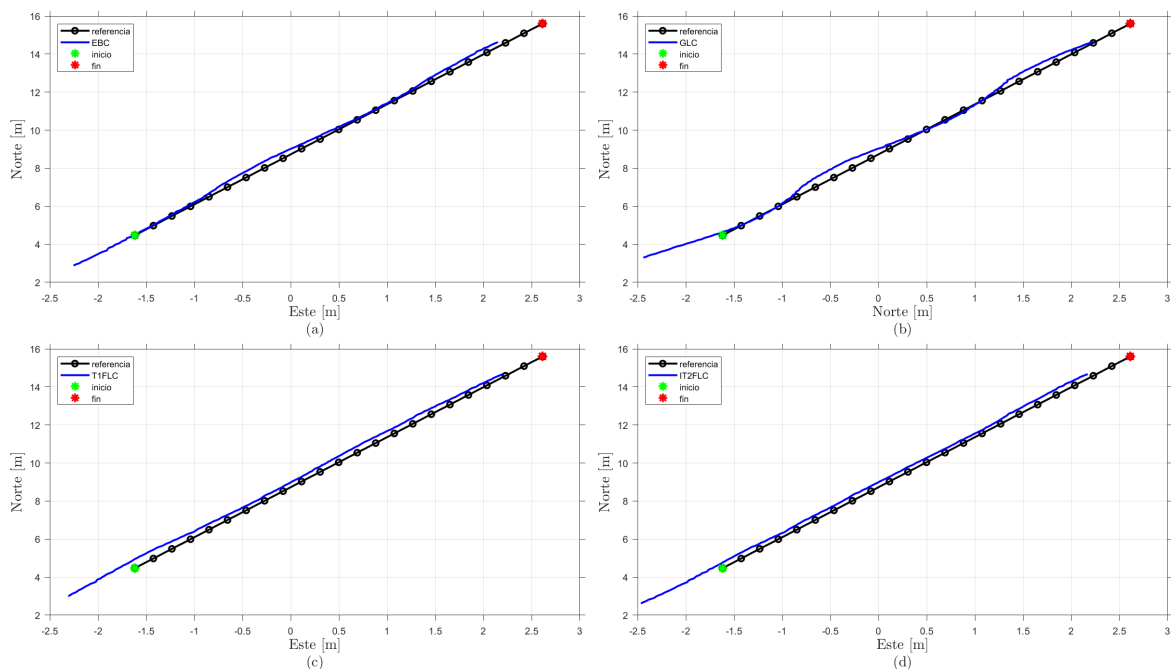


Figura 52: Comparación del desempeño de diferentes controladores para el seguimiento de una línea recta.

Tabla 30: Comparativa de RMSE del error lateral y de cabeceo de diferentes controladores para el seguimiento de tres surcos con radio de 2 m.

Errores	EBC	GLC	T1FLC 2MF	IT2FLC 2MF
RMSE error lateral	1.01 m	0.683 m	0.704 m	0.704 m
RMSE error de cabeceo	26.23°	15.89°	16.52°	16.51°

Una vez que contamos con un controlador de T1 y uno de IT2 sintonizados en igualdad de condiciones y con resultados aceptables, se observó una diferencia importante al calcular el error lateral. Inicialmente, el error lateral se calculaba utilizando la distancia al punto objetivo multiplicada por el seno del error de rumbo (Ec. 61), formando así un triángulo que permitía conocer la distancia proyectada sobre una recta, correspondiente al cateto adyacente como se observa en la Figura 56 (a). Este método presenta problemas en trayectorias con curvas, ya que, al no estar alineado el punto de ataque con el punto más cercano a la ubicación del vehículo, la línea que se intentaba calcular con el triángulo resultaba ser mayor, por consecuencia se genera un error calculado más grande. Para abordar esta limitación, se optó por encontrar el punto más cercano de la ruta a la ubicación actual del vehículo y calcular el triángulo tomando la distancia a este punto, multiplicándola por el seno del ángulo que se produce (Ec. 62), obteniendo así el cateto opuesto que representa el error lateral, como se muestra en la Figura 56 (b).

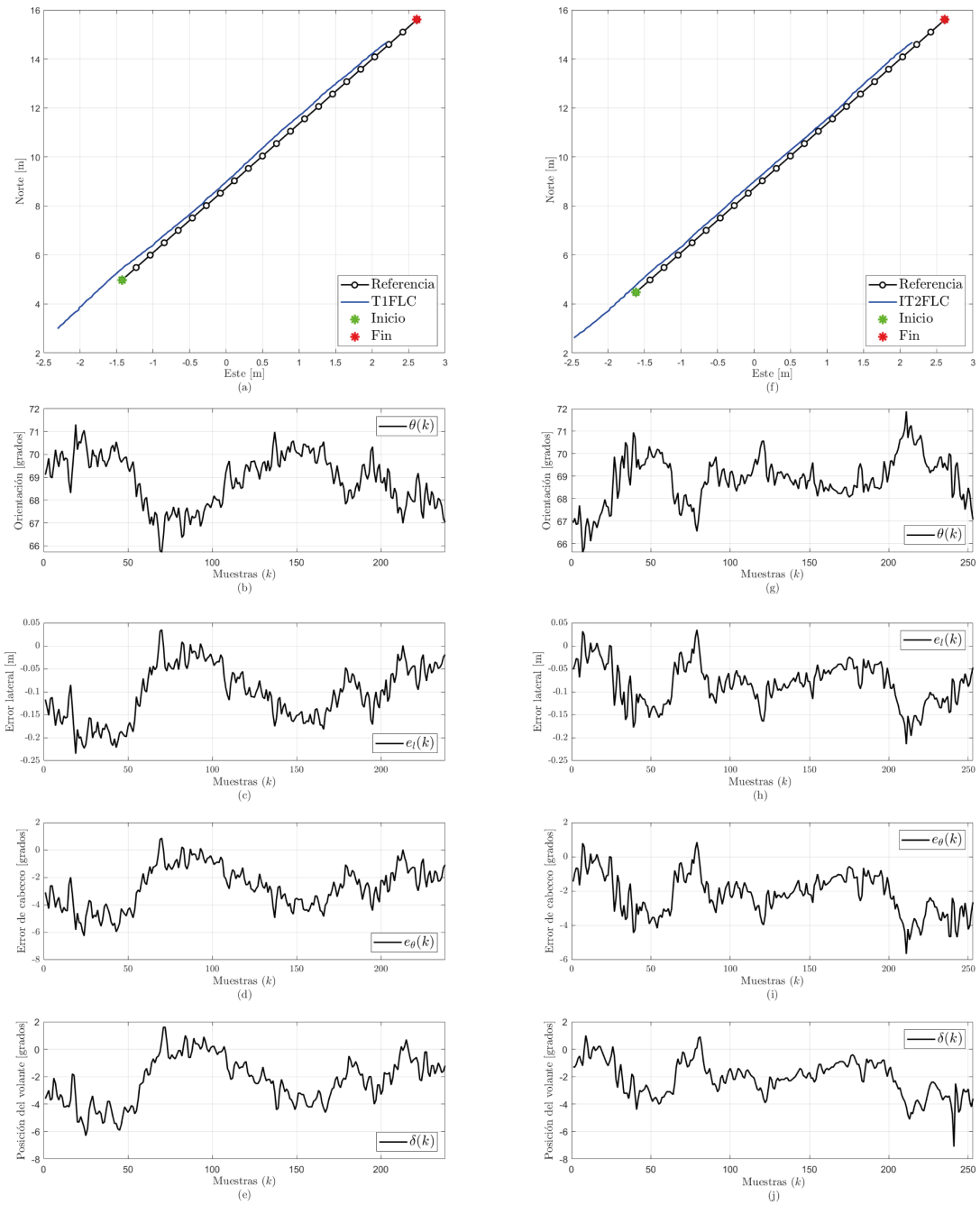


Figura 53: Desempeño del seguimiento de trayectorias en una línea recta: comparativa entre el controlador T1FLC (columna izquierda) y el IT2FLC (columna derecha). (a) y (f) Seguimiento de la ruta de referencia en el plano Este-Norte; (b) y (g) Evolución de la orientación  $\theta(k)$ ; (c) y (h) Evolución del error lateral  $e_l(k)$ ; (d) y (i) Evolución del error de cabeceo  $e_\theta(k)$ ; (e) y (j) Señal de control de dirección (posición del volante)  $\delta(k)$  aplicada al actuador.

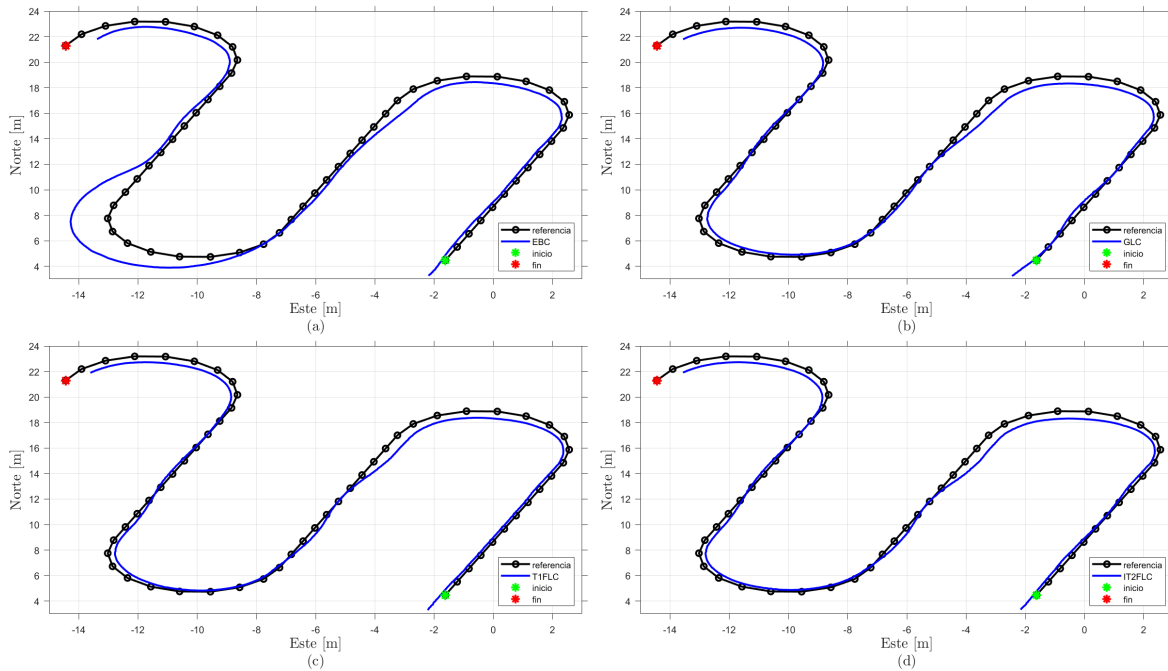


Figura 54: Comparación del desempeño de diferentes controladores para el seguimiento de tres surcos con radio de 2 m.

$$e_l = l_d \sin(e_\theta) \quad (61)$$

$$e_l = l \sin(\beta) \quad (62)$$

Ahora que se contaba con esta forma de calcular el error lateral, se realizaron ocho pruebas adicionales para validar los FLC con recorridos diferentes a los tres surcos con radio de 2 metros utilizado anteriormente. El objetivo de estas pruebas fue comprobar la capacidad de los controladores para adaptarse a rutas distintas de aquellas utilizadas durante su entrenamiento. Las nuevas rutas incluyeron tres surcos con un radio de 3 m, así como trayectos con una mayor distancia en línea recta (Figura 57). Además, se diseñó una ruta en forma de lemniscata como se muestra en la Figura 59, con el fin de poner a prueba los controladores en una trayectoria que requiere ajustes constantes debido a la curvatura continua. Estas pruebas proporcionaron una evaluación más completa del desempeño de los controladores en condiciones variadas y desafiantes.

Los resultados de los diferentes RMSE del error lateral y error de cabeceo se muestran en las Tablas 31 y 32. En estas se puede observar como el IT2FLC obtiene un menor error lateral en comparación con los demás controladores, sin embargo T1FLC tiene un menor error de cabeceo. El error de cabeceo está relacionado con la alineación del vehículo respecto a la trayectoria, pero un bajo error de cabeceo no necesariamente implica que el vehículo esté en la posición correcta en la trayectoria, sino que solo está alineado con ella. Por lo tanto, un

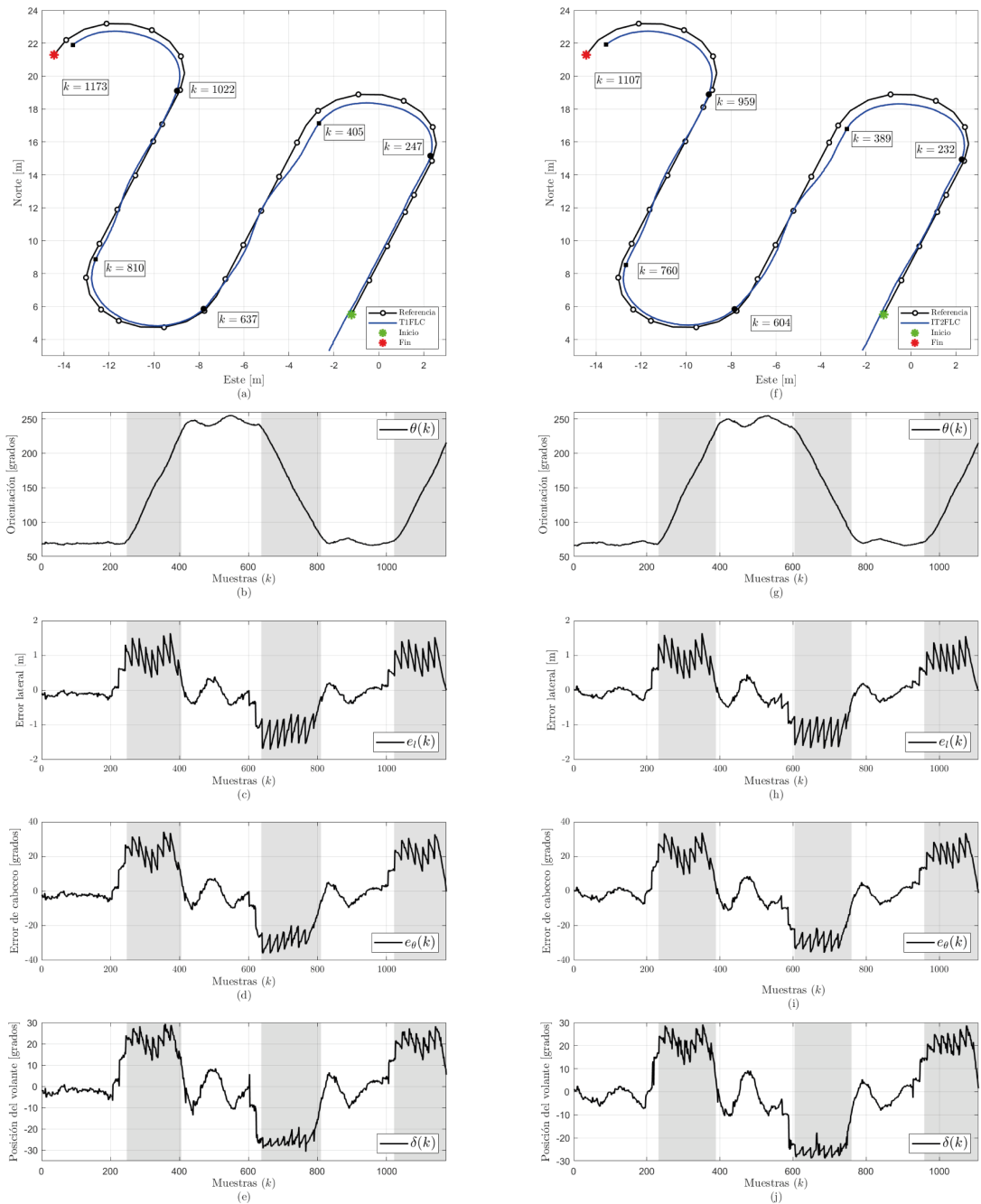


Figura 55: Desempeño del seguimiento de trayectorias en tres surcos con radio de 2 m: comparativa entre el controlador T1FLC (columna izquierda) y el IT2FLC (columna derecha). (a) y (f) Seguimiento de la ruta de referencia en el plano Este-Norte; (b) y (g) Evolución de la orientación  $\theta(k)$ ; (c) y (h) Evolución del error lateral  $e_l(k)$ ; (d) y (i) Evolución del error de cabeceo  $e_\theta(k)$ ; (e) y (j) Señal de control de dirección (posición del volante)  $\delta(k)$  aplicada al actuador.

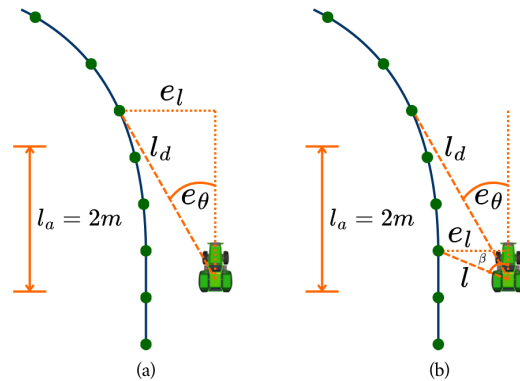


Figura 56: Métodos para calcular el error lateral en el seguimiento de una trayectoria con curvas.

error lateral bajo garantiza que la posición del vehículo sea la más cercana posible a la ruta ideal, asegurando así la eficiencia y la precisión de la operación. Por este motivo, la métrica que consideramos para determinar si un controlador es mejor que otro, es el error lateral.

El análisis visual para el surco con radio de 3 metros (Figura 58) muestra un comportamiento dinámico similar entre T1FLC e IT2FLC; las señales de la posición del volante presentan variaciones de amplitud equiparables. Sin embargo, como lo demuestra el RMSE, el IT2FLC logra mayor precisión a lo largo de este recorrido. Por otro lado, la diferencia en la acción de control se hace evidente en la trayectoria de lemniscata (Figura 60). Ante los requerimientos de ajuste constante impuestos por la curvatura continua, la señal de dirección generada por el T1FLC registra fluctuaciones continuas y picos más pronunciados. En contraste, el IT2FLC produce una señal visiblemente más suave. Este comportamiento confirma que, ante trayectorias dinámicamente complejas y distintas a las de entrenamiento, el IT2FLC no solo asegura un menor error lateral, sino que lo consigue mediante una acción de dirección más estable.

Tabla 31: Comparativa de RMSE del error lateral y de cabeceo de diferentes controladores para el seguimiento de tres surcos con radio de 3 m.

Errores	EBC	GLC	T1FLC 2MF	IT2FLC 2MF
RMSE error lateral	0.318 m	0.324 m	0.326 m	0.314 m
RMSE error de cabeceo	27.39°	14.48°	12.54°	14.34°

En conclusión, los resultados obtenidos en el seguimiento de trayectorias del TR utilizando el IT2FLC modelado con IT2-ANFIS OSTRÁ, se compararon con los métodos de control tradicionales y del T1FLC modelado con T1-ANFIS. Cabe señalar que el desempeño numérico en todas las pruebas se vio afectado por una perturbación física inherente a la plataforma de

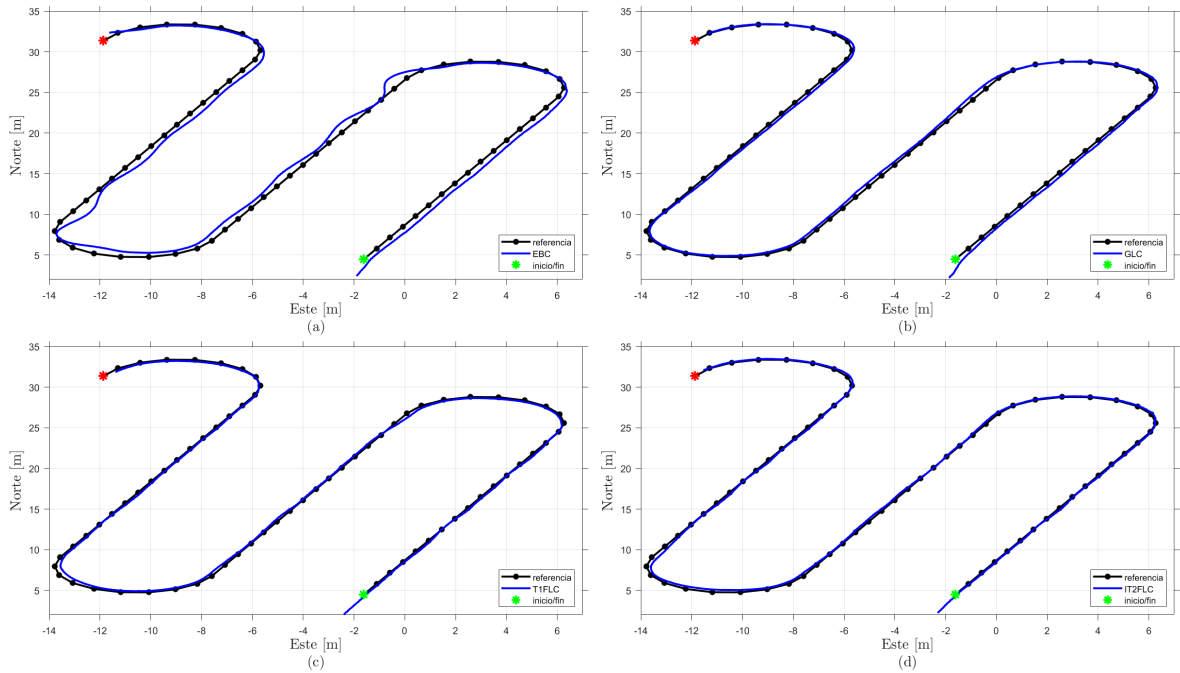


Figura 57: Comparación del desempeño de diferentes controladores para el seguimiento de tres surcos con radio de 3 m.

Tabla 32: Comparativa de RMSE del error lateral y de cabeceo de diferentes controladores para el seguimiento de una lemniscata.

Errores	EBC	GLC	T1FLC 2MF	IT2FLC 2MF
RMSE error lateral	0.331 m	0.259 m	0.2694 m	0.245 m
RMSE error de cabeceo	11.68°	12.09°	10.49°	10.80°

pruebas: una desalineación en las llantas que induce una deriva lateral continua, la cual exige correcciones permanentes incluso cuando el actuador de la dirección se encuentra en  $0^\circ$ . A pesar de esta condición, sumada a la latencia de respuesta de los actuadores y la fricción de los neumáticos frente al pavimento (cuya magnitud difiere de la resolución milimétrica puramente inercial del sensor RTK-GNSS), el IT2FLC logró reducir las métricas de RMSE en las diferentes pruebas frente a sus contrapartes. La capacidad del IT2FLC para generalizar, compensar esta deriva mecánica y adaptarse a diferentes tipos de trayectorias resalta las capacidades de la técnica propuesta y de los IT2FLS para manejar la incertidumbre y perturbaciones no modeladas. Estos avances sientan una base para futuras mejoras en la automatización del TR y el uso de técnicas de inteligencia computacional para optimizar las operaciones en el campo.

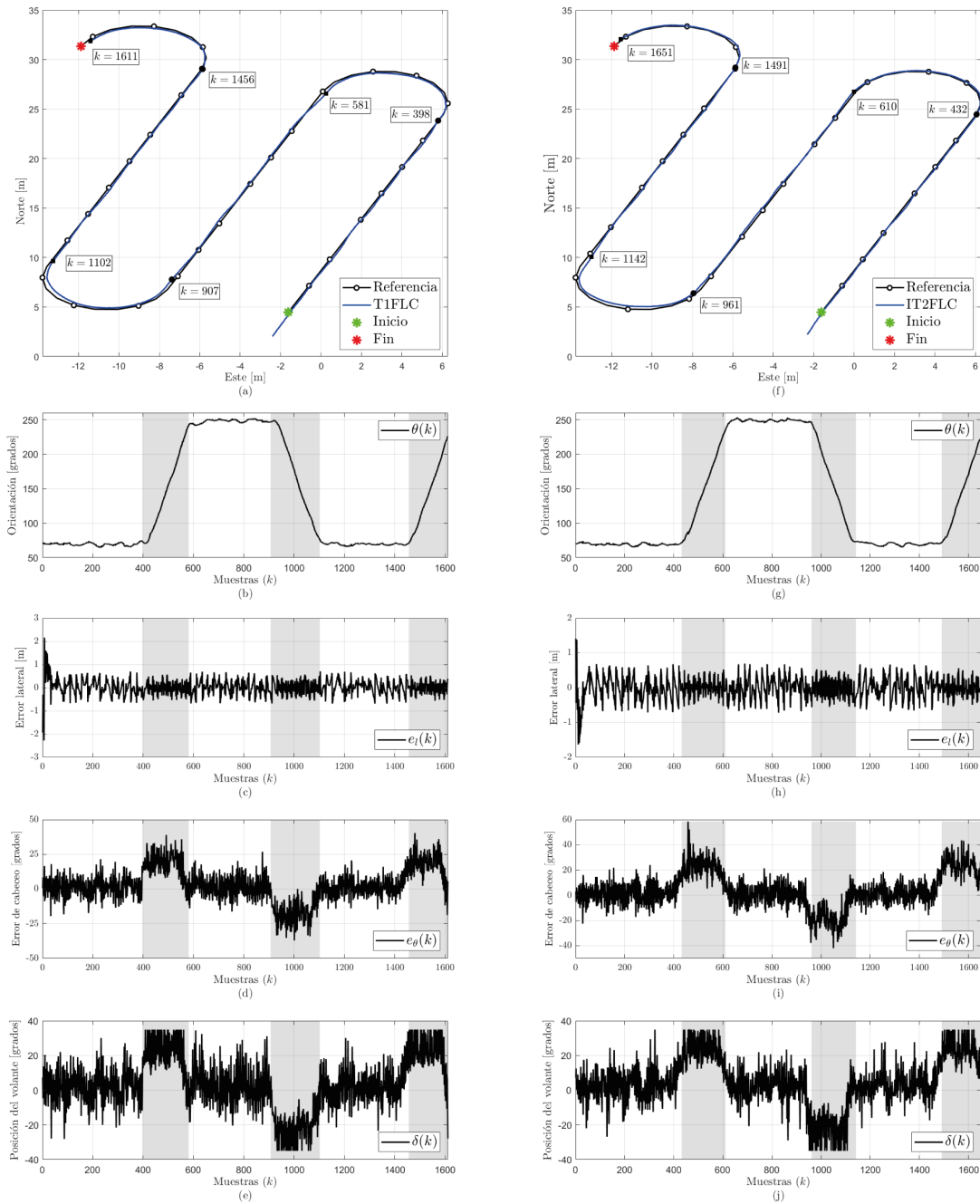


Figura 58: Desempeño del seguimiento de trayectorias en tres surcos con radio de 3m: comparativa entre el controlador T1FLC (columna izquierda) y el IT2FLC (columna derecha). (a) y (f) Seguimiento de la ruta de referencia en el plano Este-Norte; (b) y (g) Evolución de la orientación  $\theta(k)$ ; (c) y (h) Evolución del error lateral  $e_l(k)$ ; (d) y (i) Evolución del error de cabeceo  $e_\theta(k)$ ; (e) y (j) Señal de control de dirección (posición del volante)  $\delta(k)$  aplicada al actuador.

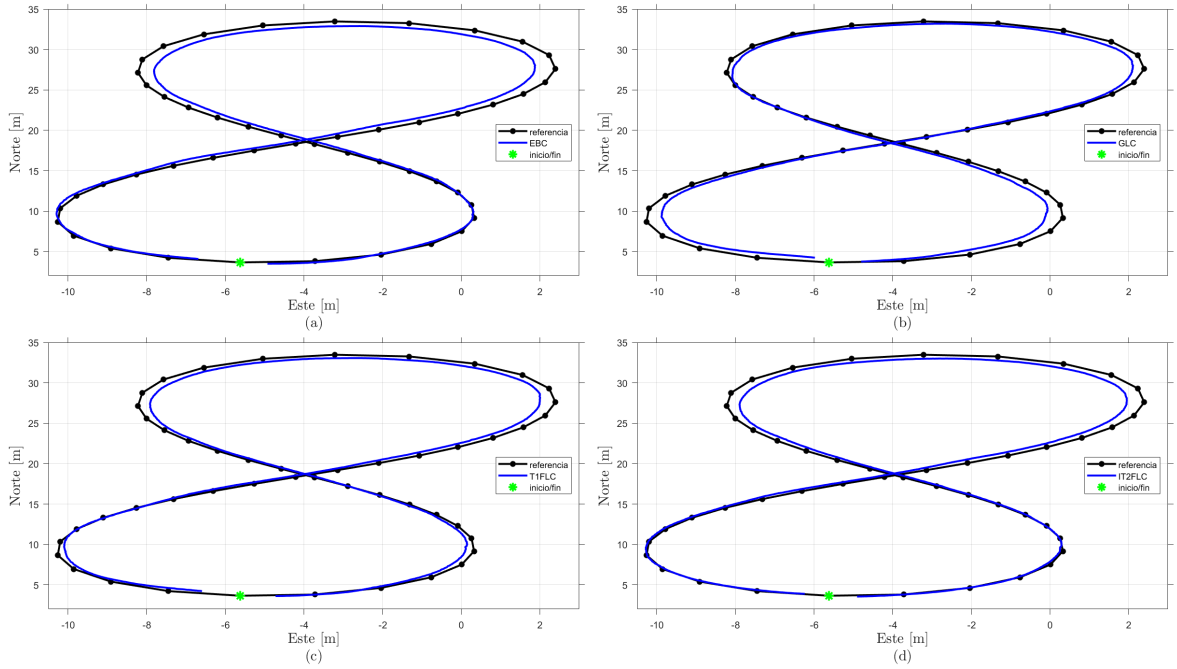


Figura 59: Comparación del desempeño de diferentes controladores para el seguimiento de una lemniscata.

### 4.3. Prueba de estabilidad

La arquitectura del sistema de control lateral implementado para el TR se detalla en la Figura 61. Este esquema presenta una estructura de control en cascada: el lazo externo consiste en el IT2-FLC (con OSTR), el cual procesa el error de orientación  $e_\theta(k)$  y su retardo  $e_\theta(k-1)$  para producir una referencia de dirección  $\delta_{ref}(k)$ . El lazo interno emplea un controlador PI para regular la posición del volante  $\delta(k)$  a través de la señal de control  $u(k)$ . Para el subsistema de dirección, se plantea un modelo de primer orden en tiempo continuo:

$$\frac{\Delta(s)}{U(s)} = \frac{K_m}{\tau s + 1}, \quad (63)$$

donde  $K_m$  es la ganancia del sistema y  $\tau$  es la constante de tiempo del mismo. Al aplicar la transformada inversa de Laplace con condiciones iniciales nulas, se obtiene la ecuación diferencial que rige la dinámica del actuador:

$$\tau \dot{\delta}(t) + \delta(t) = K_m u(t). \quad (64)$$

Para su implementación digital, se considera el periodo de muestreo  $T_s = 20$  ms y se aplica la aproximación de la derivada discreta de Euler:

$$\dot{\delta}(t) \approx \frac{\delta(k+1) - \delta(k)}{T_s}. \quad (65)$$

Sustituyendo la Ec. (65) en Ec. (64), resulta en la expresión discretizada:

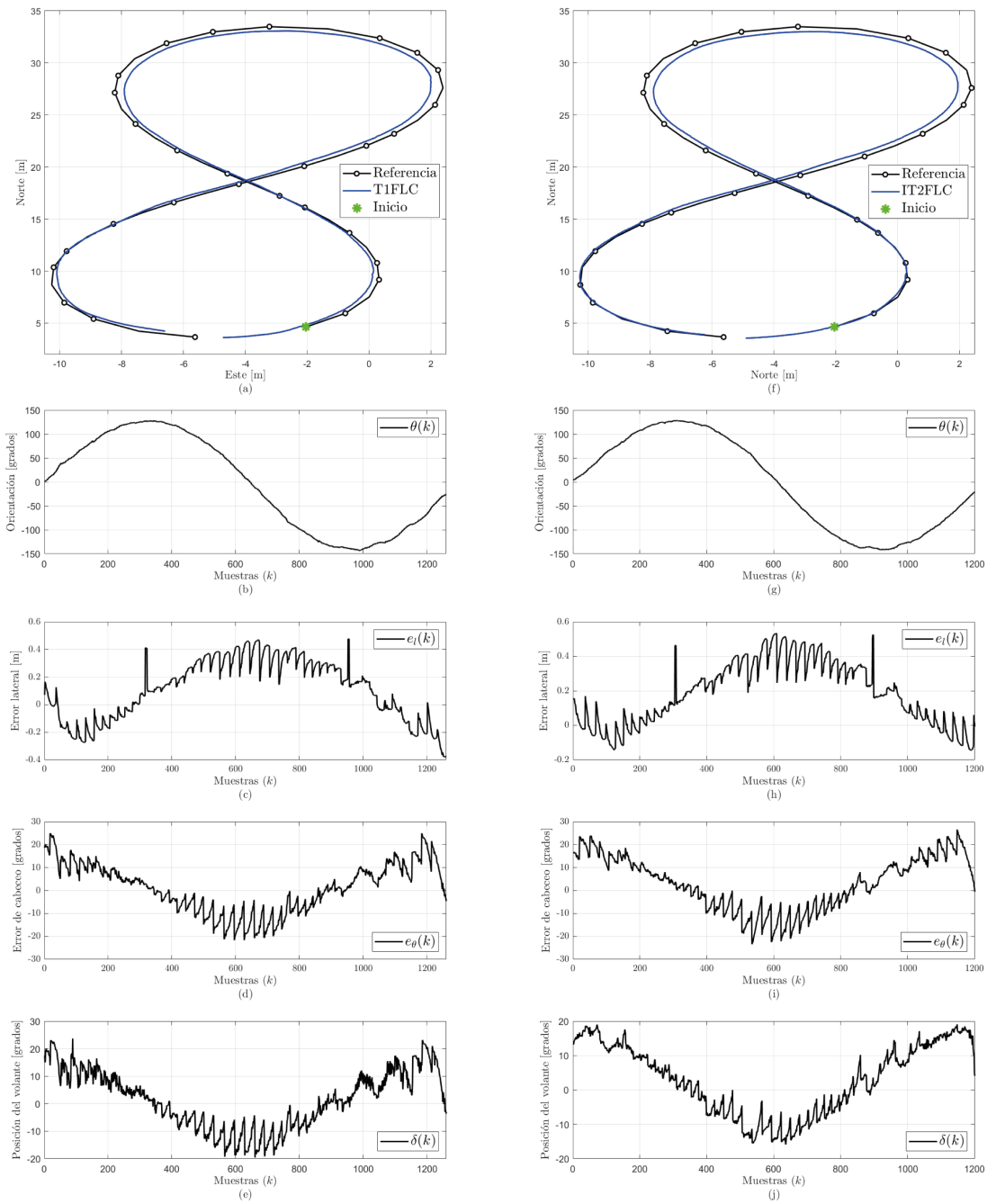


Figura 60: Desempeño del seguimiento de trayectorias en una lemniscata: comparativa entre el controlador T1FLC (columna izquierda) y el IT2FLC (columna derecha). (a) y (f) Seguimiento de la ruta de referencia en el plano Este-Norte; (b) y (g) Evolución de la orientación  $\theta(k)$ ; (c) y (h) Evolución del error lateral  $e_l(k)$ ; (d) y (i) Evolución del error de cabeceo  $e_\theta(k)$ ; (e) y (j) Señal de control de dirección  $\delta(k)$  aplicada al actuador.

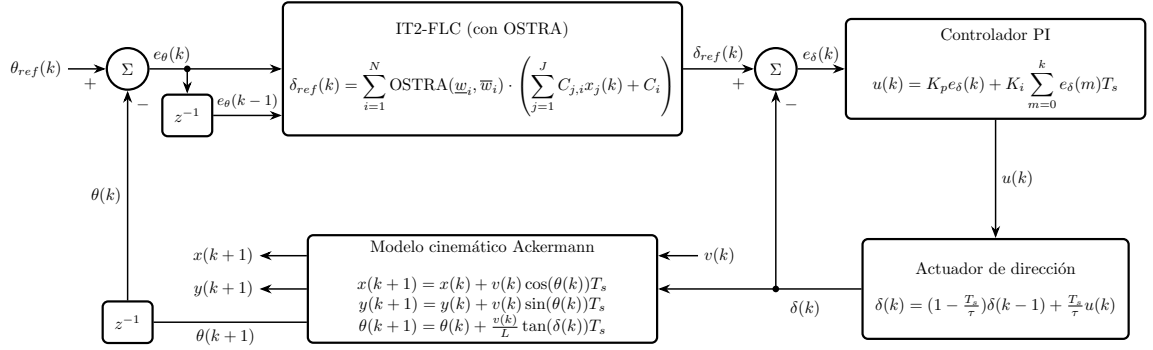


Figura 61: Representación matemática en diagrama a bloques del sistema de control lateral en lazo cerrado.

$$\tau \left( \frac{\delta(k+1) - \delta(k)}{T_s} \right) + \delta(k) = K_m u(k). \quad (66)$$

Al despejar el estado futuro del ángulo de dirección  $\delta(k+1)$  se obtiene:

$$\tau \delta(k+1) + (T_s - \tau) \delta(k) = T_s K_m u(k). \quad (67)$$

La ecuación en diferencias que rige el comportamiento del actuador en el diagrama de la Figura 61 es:

$$\delta(k+1) = \left[ \frac{\tau - T_s}{\tau} \right] \delta(k) + \left[ \frac{T_s K_m}{\tau} \right] u(k). \quad (68)$$

Finalmente, el desplazamiento del vehículo se rige por su modelo cinemático, cuyas ecuaciones se detallan en la Ec. (43).

Con base en la arquitectura anterior, se plantea realizar un análisis de estabilidad del sistema utilizando la desigualdad cuadrática de Lyapunov para sistemas difusos T-S. Para simplificar esta tarea, se asume que el lazo interno de control posee una respuesta lo suficientemente rápida para considerar que el ángulo real de las llantas sigue fielmente la referencia ( $\delta(k) \approx \delta_{ref}(k)$ ). Además, se emplea la linealización del modelo cinemático bajo la premisa de ángulos pequeños. A continuación, se muestra el desarrollo realizado para llevar el sistema a una representación en espacio de estados discreto, partiendo de la cinemática en tiempo continuo.

El comportamiento cinemático del TR obedece a las ecuaciones matemáticas del modelo de Ackermann Ec. (42), donde la velocidad angular  $\dot{\theta}$  en tiempo continuo se calcula como se muestra a continuación:

$$\dot{\theta}(t) = \frac{v}{L} \tan(\delta(t)). \quad (69)$$

Para la tarea de seguimiento de trayectorias enfocada estrictamente en la dinámica lateral, el error de cabeceo se define como la diferencia entre una referencia  $\theta_r$  y la orientación actual  $\theta$ :

$$e_\theta(t) = \theta_r(t) - \theta(t). \quad (70)$$

Considerando que en tramos referenciados rectos u homogéneos  $\theta_r(t)$  permanece fija localmente, su derivada es nula ( $\dot{\theta}_r(t) = 0$ ). Derivando la expresión del error se obtiene  $\dot{e}_\theta(t) = -\dot{\theta}(t)$ . Sustituyendo la ecuación cinemática Ec. (69) en la ecuación anterior resulta en:

$$\dot{e}_\theta(t) = -\frac{v}{L} \tan(\delta(t)). \quad (71)$$

Considerando la aproximación de ángulos pequeños en el diseño del control para maniobras operativas, se asume que la función tangente se aproxima linealmente a su propio argumento:  $\tan(\delta) \approx \delta$ . Esto permite llevar la dinámica a su forma afín lineal representativa:

$$\dot{e}_\theta(t) = -\frac{v}{L} \delta(t). \quad (72)$$

El algoritmo de control fue embebido en microcontroladores utilizando tiempo discreto. Aplicando la aproximación de la derivada discreta de Euler con un tiempo de muestreo de  $T_s = 20 \text{ ms} = 0.02 \text{ s}$ :

$$\dot{e}_\theta(t) \approx \frac{e_\theta(k+1) - e_\theta(k)}{T_s}. \quad (73)$$

Sustituyendo la Ec. (73) en la Ec. (72), se obtiene la ecuación en tiempo discreto:

$$\frac{e_\theta(k+1) - e_\theta(k)}{T_s} = -\frac{v}{L} \delta(k). \quad (74)$$

Despejando el error  $e_\theta(k+1)$  de la ecuación anterior se obtiene:

$$e_\theta(k+1) = e_\theta(k) - T_s \left( \frac{v}{L} \right) \delta(k). \quad (75)$$

Para simplificar la formulación, definimos la constante de ganancia geométrica invariable del TR como  $K = v/L$ . El IT2FLC propuesto cuenta con consecuentes lineales de tipo T-S. Específicamente, para cada regla local denotada por  $i$ , el sistema emite la corrección requerida para el actuador del volante  $\delta_i(k)$  a través de la ecuación analítica polinomial evaluada sobre las variables de entrada del controlador (error de cabeceo y derivada del error de cabeceo):

$$\delta_i(k) = C_{i,1} e_\theta(k) + C_{i,2} \left[ \frac{e_\theta(k) - e_\theta(k-1)}{T_s} \right] + C_{i,3}. \quad (76)$$

Esta salida analítica representa el consecuente matemático de la inferencia de T-S. Estructuralmente, al vincular las variables de entrada con sus respectivos IT2FS ( $\tilde{A}_{1,k}$  y  $\tilde{A}_{2,k}$ ), la forma condicional "SI-ENTONCES" que rige esta  $i$ -ésima regla se expresa como:

$$R_i: \text{ SI } e_\theta(k) \text{ está en } A_{1,k} \text{ y } \left[ \frac{e_\theta(k) - e_\theta(k-1)}{T_s} \right] \text{ está en } A_{2,k}. \quad (77)$$

$$\text{ENTONCES } \delta_i(k) = C_{i,1}e_\theta(k) + C_{i,2} \left[ \frac{e_\theta(k) - e_\theta(k-1)}{T_s} \right] + C_{i,3}.$$

Para analizar la estabilidad asintótica en el punto de equilibrio ( $e = 0$ ), se ignora el término independiente  $C_{i,3}$ . Esto permite aislar la dinámica del sistema y obtener la matriz característica homogénea, definiendo:

$$\delta_i(k) = C_{i,1}e_\theta(k) + C_{i,2} \left[ \frac{e_\theta(k) - e_\theta(k-1)}{T_s} \right]. \quad (78)$$

Al sustituir la ley de control de la Ec. (78) en la Ec. (75) del error, se obtiene la dinámica en lazo cerrado:

$$e_\theta(k+1) = e_\theta(k) - T_s K \left( C_{i,1}e_\theta(k) + \frac{C_{i,2}}{T_s} (e_\theta(k) - e_\theta(k-1)) \right), \quad (79)$$

al multiplicar y simplificar el tiempo de muestreo  $T_s$ , la expresión se reduce a:

$$e_\theta(k+1) = e_\theta(k) - T_s K C_{i,1} e_\theta(k) - K C_{i,2} e_\theta(k) + K C_{i,2} e_\theta(k-1). \quad (80)$$

Al agrupar términos comunes según su instante de tiempo ( $k$  y  $k-1$ ), se obtiene:

$$e_\theta(k+1) = [K C_{i,2}] e_\theta(k-1) + [1 - K(T_s C_{i,1} + C_{i,2})] e_\theta(k). \quad (81)$$

Para representar el sistema en un espacio de estados, se utiliza la forma canónica discreta común en sistemas con control PD. Así, se definen las dos variables de estado: el error del ciclo anterior  $x_1(k)$  y el error del ciclo actual  $x_2(k)$ :

$$x_1(k) = e_\theta(k-1), \quad x_2(k) = e_\theta(k). \quad (82)$$

Dado que el estado futuro del primer componente es el valor presente del segundo,  $x_1(k+1) = e_\theta(k) = x_2(k)$ , sustituimos esta relación en la ecuación diferencial (81). De este modo, el sistema se reescribe exclusivamente en función de sus variables de estado:

$$\begin{bmatrix} x_1(k+1) \\ x_2(k+1) \end{bmatrix} = \underbrace{\begin{bmatrix} 0 & 1 \\ K C_{i,2} & 1 - K(T_s C_{i,1} + C_{i,2}) \end{bmatrix}}_{\mathbf{A}_i} \begin{bmatrix} x_1(k) \\ x_2(k) \end{bmatrix}. \quad (83)$$

De este modo,  $\mathbf{A}_i$  constituye el modelo lineal invariante en el tiempo en lazo cerrado para cada submodelo local. Esta representación es válida cuando el grado de activación de una regla es total, lo cual facilita el análisis de estabilidad individual.

Al integrar la planta cinemática dependiente de la ganancia  $K$  y el tiempo de muestreo  $T_s$ , se demuestra formalmente cómo los parámetros físicos del tractor intervienen en la convergencia del error. Para comprobar numéricamente dicha convergencia en lazo cerrado, se procede a evaluar las matrices  $\mathbf{A}_i$  del sistema. Se utiliza el valor medio de velocidad  $v = 0.453$  m/s

como punto de operación nominal, asumiendo variaciones despreciables durante la ejecución de la trayectoria en línea utilizando el IT2FLC mostrado en la Figura 52. Considerando que el TR tiene distancia entre ejes de  $L = 1.3$  m, se obtiene una ganancia cinemática de  $K = 0.34892$ . Los parámetros del IT2FLC empleados en la experimentación, derivados del modelado mediante IT2-ANFIS OSTRÁ, se presentan a continuación en la Tabla 33.

Tabla 33: Parámetros óptimos del IT2FLC empleado en la experimentación

FS	MF	Parámetros premisa (MF)					Parámetros consecuentes			
		$\underline{a}$	$\bar{a}$	$b$	$c$	$h$	$C_{i,j}$	$C_{i,1}$	$C_{i,2}$	$C_{i,3}$
$\tilde{A}_1$	$\underline{A}_1$	17.8226	-	1.3713	-56.7445	0.1	Regla 1	0.1855	0.7994	-26.9896
	$\bar{A}_1$	-	21.4092			1				
$\tilde{A}_2$	$\underline{A}_2$	17.8583	-	1.1598	32.4727	0.1	Regla 2	1.7428	-2.5663	12.9292
	$\bar{A}_2$	-	21.4133			1				
$\tilde{A}_3$	$\underline{A}_3$	5.4435	-	0.8997	-11.6998	1	Regla 3	1.0942	-1.9010	21.0211
	$\bar{A}_3$	-	6.5314			1				
$\tilde{A}_4$	$\underline{A}_4$	5.4035	-	1.1279	15.4173	0.1	Regla 4	0.1346	1.4644	-15.5252
	$\bar{A}_4$	-	6.4781			1				

Sustituyendo los parámetros consecuentes del IT2FLC mostrados en la Tabla 33 en la expresión general de  $\mathbf{A}_i$  mostrada en la Ec. (83), se obtienen las cuatro matrices que definen la dinámica del sistema en lazo cerrado:

$$\mathbf{A}_1 = \begin{bmatrix} 0 & 1 \\ 0.2789 & 0.7198 \end{bmatrix}, \quad \mathbf{A}_2 = \begin{bmatrix} 0 & 1 \\ -0.8954 & 1.8833 \end{bmatrix}, \quad (84)$$

$$\mathbf{A}_3 = \begin{bmatrix} 0 & 1 \\ -0.6633 & 1.6556 \end{bmatrix}, \quad \mathbf{A}_4 = \begin{bmatrix} 0 & 1 \\ 0.5110 & 0.4881 \end{bmatrix}.$$

Para verificar la estabilidad asintótica local de cada regla, se analizan las raíces de su polinomio característico. Dado que todas las matrices de submodelo comparten la estructura canónica  $\mathbf{A}_i = \begin{bmatrix} 0 & 1 \\ a_{21} & a_{22} \end{bmatrix}$ , el polinomio se obtiene mediante la solución de  $\det(\lambda \mathbf{I} - \mathbf{A}_i) = 0$ , lo que resulta en la ecuación cuadrática:

$$\lambda^2 - a_{22}\lambda - a_{21} = 0. \quad (85)$$

Para obtener los autovalores de forma directa, se aplica la solución general para ecuaciones de segundo grado, definida como:

$$\lambda_{1,2} = \frac{a_{22} \pm \sqrt{(a_{22})^2 + 4a_{21}}}{2}, \quad (86)$$

donde al evaluar esta expresión con los coeficientes de la segunda fila de cada matriz  $\mathbf{A}_i$ , se obtienen los siguientes autovalores:

- **Para  $\mathbf{A}_1$ :**  $\lambda_{1,2} = 0.9990, -0.2792$  ( $|\lambda| < 1$ ).
- **Para  $\mathbf{A}_2$ :**  $\lambda_{1,2} = 0.9416 \pm i0.0936$  ( $|\lambda| = \sqrt{0.9416^2 + 0.0936^2} = 0.9463 < 1$ ).
- **Para  $\mathbf{A}_3$ :**  $\lambda_{1,2} = 0.9762, 0.6795$  ( $|\lambda| < 1$ ).
- **Para  $\mathbf{A}_4$ :**  $\lambda_{1,2} = 0.9994, -0.5113$  ( $|\lambda| < 1$ ).

Todos los submodelos en espacio de estados, configurados con los parámetros del IT2FLC obtenidos mediante IT2-ANFIS OSTRÁ, son asintóticamente estables. Al situarse todas las raíces de los polinomios característicos dentro del círculo unitario discreto, se asegura que el error de orientación convergerá a cero. Sin embargo, la estabilidad individual no garantiza matemáticamente el comportamiento del sistema cuando las cuatro reglas interactúan dinámicamente. Por lo tanto, para validar la estabilidad del modelo no lineal completo, se debe demostrar la existencia de una función de Lyapunov común mediante el Teorema 1 ([91]) que se describe a continuación:

**Teorema 1.** *El equilibrio del sistema difuso discreto  $x(k+1) = \sum_{i=1}^4 \mu_i(e_\theta, \dot{e}_\theta) \mathbf{A}_i x(k)$  es globalmente asintóticamente estable si existe una matriz  $\mathbf{P}$  común, simétrica y definida positiva tal que:*

$$\mathbf{A}_i^T \mathbf{P} \mathbf{A}_i - \mathbf{P} < 0, \quad \forall i = 1, \dots, 4,$$

*es decir, debe existir una matriz  $\mathbf{P}$  común que satisfaga la condición de estabilidad para todos los submodelos locales simultáneamente.*

Cabe precisar que, aunque el sistema propuesto emplea un IT2FLC, el criterio de estabilidad del Teorema 1 es plenamente aplicable. Esto se debe a que los consecuentes de las reglas locales son funciones lineales T1, lo que garantiza que las matrices de estado  $\mathbf{A}_i$  sean determinísticas. En esta configuración, la naturaleza IT2 del controlador incide únicamente en el cálculo de los pesos de activación  $\mu_i$ , pero no altera la estructura de las LMI requeridas para hallar la matriz  $\mathbf{P}$  común.

Para verificar analíticamente la estabilidad global del sistema, se sustituye cada una de las matrices de submodelo  $\mathbf{A}_i$  en la expresión del Teorema 1. Recordando que la condición exige el cumplimiento de  $\mathbf{A}_i^T \mathbf{P} \mathbf{A}_i - \mathbf{P} < 0$ , se despliega el siguiente sistema explícito de cuatro LMI:

Para el submodelo 1:

$$\begin{bmatrix} 0 & 0.2789 \\ 1 & 0.7198 \end{bmatrix} \begin{bmatrix} P_1 & P_2 \\ P_2 & P_3 \end{bmatrix} \begin{bmatrix} 0 & 1 \\ 0.2789 & 0.7198 \end{bmatrix} - \begin{bmatrix} P_1 & P_2 \\ P_2 & P_3 \end{bmatrix} < 0. \quad (87)$$

Para el submodelo 2 (dinámica subamortiguada):

$$\begin{bmatrix} 0 & -0.8954 \\ 1 & 1.8833 \end{bmatrix} \begin{bmatrix} P_1 & P_2 \\ P_2 & P_3 \end{bmatrix} \begin{bmatrix} 0 & 1 \\ -0.8954 & 1.8833 \end{bmatrix} - \begin{bmatrix} P_1 & P_2 \\ P_2 & P_3 \end{bmatrix} < 0. \quad (88)$$

Para el submodelo 3:

$$\begin{bmatrix} 0 & -0.6633 \\ 1 & 1.6556 \end{bmatrix} \begin{bmatrix} P_1 & P_2 \\ P_2 & P_3 \end{bmatrix} \begin{bmatrix} 0 & 1 \\ -0.6633 & 1.6556 \end{bmatrix} - \begin{bmatrix} P_1 & P_2 \\ P_2 & P_3 \end{bmatrix} < 0. \quad (89)$$

Para el submodelo 4:

$$\begin{bmatrix} 0 & 0.5110 \\ 1 & 0.4881 \end{bmatrix} \begin{bmatrix} P_1 & P_2 \\ P_2 & P_3 \end{bmatrix} \begin{bmatrix} 0 & 1 \\ 0.5110 & 0.4881 \end{bmatrix} - \begin{bmatrix} P_1 & P_2 \\ P_2 & P_3 \end{bmatrix} < 0. \quad (90)$$

Este sistema de inecuaciones fue resuelto computacionalmente mediante algoritmos de optimización convexa. Para garantizar un correcto acondicionamiento numérico y evitar soluciones triviales en el límite de la tolerancia computacional de la máquina, se incorporó al solucionador la restricción de que la matriz buscada sea estrictamente mayor a la identidad ( $\mathbf{P} \geq \mathbf{I}$ ). Bajo esta condición, el algoritmo halló la siguiente matriz  $\mathbf{P}$  común que satisface simultáneamente las cuatro condiciones de estabilidad:

$$\mathbf{P} = \begin{bmatrix} 2740.7 & -2734.8 \\ -2734.8 & 2769.9 \end{bmatrix}. \quad (91)$$

Para verificar analíticamente que la solución encontrada es estrictamente definida positiva ( $\mathbf{P} > 0$ ), se aplica el criterio de Sylvester sobre sus menores principales:

- **Primer menor principal:**  $M_1 = P_{11} = 2740.7 > 0$ .
- **Segundo menor principal (Determinante):**

$$\det(\mathbf{P}) = (2740.7)(2769.9) - (-2734.8)^2 = 1.1247 \times 10^5 > 0. \quad (92)$$

Al comprobarse la existencia de una matriz  $\mathbf{P}$  común, simétrica y definida positiva, se demuestra analíticamente mediante el Teorema 1 de Lyapunov para sistemas Takagi-Sugeno en tiempo discreto [91], que el origen del sistema en lazo cerrado es globalmente asintóticamente estable; la conclusión es de carácter global por las funciones de membresía son convexas para todo el universo de discurso. Esto garantiza matemáticamente que el error de orientación  $e_\theta(k)$  convergerá a cero operando dentro del universo de discurso de los parámetros premisa y consecuentes modelados utilizando IT2-ANFIS OSTR.

Este resultado analítico se corrobora de manera consistente en la práctica. Tal como se demostró experimentalmente en la Sección 4.2, al implementar el IT2FLC en el TR físico, las señales de error de orientación y error lateral convergieron hacia la referencia deseada en tiempo real bajo diferentes trayectorias (rectas, curvas y lemniscata). El controlador demostró superar las perturbaciones físicas y dinámicas no modeladas del vehículo, validando en el plano físico el postulado matemático.

### 4.3.1. Prueba de estabilidad ante variaciones de velocidad

En la sección anterior se demostró la existencia de una matriz  $\mathbf{P}$  común que garantiza la estabilidad asintótica global del sistema operando a la velocidad nominal de  $v = 0.453$  m/s. Para expandir este análisis y evaluar el comportamiento del controlador IT2FLC frente a diferentes escenarios de operación, se realizó un barrido paramétrico de la velocidad longitudinal  $v$ . El objetivo de esta prueba es determinar el rango de velocidades en el cual el sistema mantiene su estabilidad global sin necesidad de reajustar los parámetros consecuentes del controlador. Se evaluó un rango de velocidades desde 0.1 m/s hasta 2.0 m/s, con incrementos de 0.1 m/s. Para cada valor de velocidad  $v$ , se recalculó la ganancia cinemática del vehículo  $K$  mediante la relación:

$$K = \frac{v}{L}, \quad (93)$$

donde  $L = 1.3$  m es la distancia entre ejes del TR.

Una vez obtenido el nuevo valor de  $K$ , se actualizaron las cuatro matrices de estado en lazo cerrado  $\mathbf{A}_i$  (para  $i = 1, \dots, 4$ ) correspondientes a cada regla difusa. Posteriormente, se resolvió el sistema de LMI planteado en el Teorema 1 utilizando algoritmos de optimización convexa, manteniendo la restricción estricta de que  $\mathbf{P} \geq \mathbf{I}$  para evadir soluciones numéricas triviales. Los resultados de este análisis iterativo se condensan en la Tabla 34. Para cada velocidad, se indica si el solucionador encontró una matriz factible y se reportan los valores de los elementos de dicha matriz, así como su determinante.

Tabla 34: Resultados de la prueba LMI para estabilidad global variando la velocidad  $v$

$v$ (m/s)	Ganancia $K$	LMI Factible	$P_{11}$	$P_{12}$	$P_{22}$	$\det(\mathbf{P})$
0.1	0.0769	Sí	2007.2	-2002.3	2115.2	$2.37 \times 10^5$
0.2	0.1538	Sí	369826.1	-368829.9	379009.2	$4.13 \times 10^9$
0.3	0.2308	Sí	2222.2	-2216.9	2257.7	$1.02 \times 10^5$
0.4	0.3077	Sí	3730.1	-3720.4	3772.0	$2.28 \times 10^5$
0.5	0.3846	Sí	6950.6	-6930.9	7006.7	$6.63 \times 10^5$
0.6	0.4615	No (Falla)	-	-	-	-
0.7	0.5385	No (Falla)	-	-	-	-
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$
2.0	1.5385	No (Falla)	-	-	-	-

De acuerdo con los datos presentados en la Tabla 34, el algoritmo encontró una matriz simétrica común  $\mathbf{P}$  para velocidades desde 0.1 m/s hasta 0.5 m/s. En este intervalo, las matrices halladas cumplen estrictamente con el criterio de Sylvester para ser definidas positivas ( $\mathbf{P} > 0$ ). Específicamente, tanto el primer menor principal ( $P_{11} > 0$ ) como el segundo menor principal

o determinante ( $\det(\mathbf{P}) > 0$ ) arrojan valores positivos de manera consistente para cada evaluación.

Por otro lado, a partir de la velocidad de 0.6 m/s y hasta el límite evaluado de 2.0 m/s, el algoritmo computacional reporta infactibilidad para el sistema de inecuaciones. Desde la perspectiva de la teoría de control, esta ausencia de solución matemática implica que no existe una función de Lyapunov cuadrática común, por lo que la estabilidad global del sistema ya no está garantizada. Analíticamente, esta pérdida de convergencia se puede comprobar revisando los autovalores de los submodelos individuales en esta región operativa. Por ejemplo, al evaluar el sistema a  $v = 0.6$  m/s ( $K = 0.4615$ ), la matriz en lazo cerrado para la segunda regla se define como:

$$\mathbf{A}_2 = \begin{bmatrix} 0 & 1 \\ -1.1843 & 2.1683 \end{bmatrix}. \quad (94)$$

Calculando las raíces de su polinomio característico  $\det(\lambda \mathbf{I} - \mathbf{A}_2) = 0$ , se obtienen los polos  $\lambda = 1.084 \pm i0.098$ . La magnitud de estos polos es  $|\lambda| = 1.088$ , la cual es estrictamente mayor a 1. Al existir un submodelo localmente inestable con polos fuera del círculo unitario discreto, no se satisface una condición necesaria para la existencia de una matriz definida positiva  $\mathbf{P}$  común. En consecuencia, la demostración de la estabilidad asintótica global mediante este teorema cuadrático se vuelve matemáticamente inalcanzable. Esto indica que los parámetros del controlador, sintonizados para 0.453 m/s, mantienen la estabilidad del sistema únicamente hasta un umbral máximo de 0.5 m/s.

#### 4.3.2. No linealidad del actuador e invarianza dimensional

El desarrollo analítico de la dinámica del error presentado en la Ec. (72) emplea la aproximación trigonométrica de ángulos pequeños,  $\tan(\delta) \approx \delta$ . Dado que el actuador de dirección del TR opera con un límite físico de  $\pm 30^\circ$  ( $\approx \pm 0.5236$  rad), es pertinente evaluar analíticamente la discrepancia que esta simplificación introduce en los extremos operativos. Para demostrar matemáticamente este efecto, se parte de la ecuación cinemática de la dinámica continua sin aproximar:

$$\dot{\theta}(t) = -\frac{v}{L} \tan(\delta(t)). \quad (95)$$

Multiplicando y dividiendo el término de la dirección por  $\delta(t)$ , se aísla algebraicamente la función no lineal:

$$\dot{\theta}(t) = -\frac{v}{L} \left( \frac{\tan(\delta(t))}{\delta(t)} \right) \delta(t). \quad (96)$$

Este procedimiento analítico permite tratar la no linealidad trigonométrica como un factor numérico acotado que multiplica a la variable lineal  $\delta(t)$ , siguiendo la metodología de construcción y acotamiento de modelos de Tanaka y Wang [91]. Al evaluar el factor aislado en el

límite máximo de operación del volante ( $\delta_{max} = 0.5236$  rad), se obtiene la proporción máxima de incremento:

$$\frac{\tan(0.5236)}{0.5236} \approx 1.1027. \quad (97)$$

Sustituyendo este valor máximo en la dinámica del sistema, el factor numérico de 1.1027 afecta directamente a la ganancia cinemática de la planta. Al agrupar de forma proporcional este factor con la velocidad nominal  $v = 0.453$  m/s, la exigencia dinámica equivalente para el modelo linealizado se define como:

$$v_{eq} = 0.453 \text{ m/s} \times 1.1027 \approx 0.4995 \text{ m/s}. \quad (98)$$

Retomando los resultados de la prueba LMI en la Tabla 34, el algoritmo solucionador reportó factibilidad para velocidades de hasta 0.5 m/s. Dado que la demanda dinámica máxima generada por el límite físico del volante (0.4995 m/s) es estrictamente menor que el valor máximo comprobado numéricamente (0.5 m/s), se garantiza formalmente que el controlador IT2FLC conserva la convergencia asintótica global en todo el rango físico de operación de la dirección.

Adicional a los límites físicos del actuador, es pertinente establecer la relación de homogeneidad entre las unidades de medida utilizadas durante el modelado IT2-ANFIS OSTRA y las requeridas por el análisis analítico. El modelo cinemático del TR exige el uso de radianes; no obstante, los datos de entrada y salida utilizados para sintonizar el IT2FLC fueron adquiridos en grados. La ley de control difusa, omitiendo el término de ajuste  $C_{i,3}$  para el análisis de estabilidad en el origen, se expresa como:

$$\delta_{grados}(k) = C_{i,1}e_{\theta,grados}(k) + C_{i,2} \left[ \frac{e_{\theta,grados}(k) - e_{\theta,grados}(k-1)}{T_s} \right]. \quad (99)$$

La conversión entre sistemas de medición angular es un factor de escala constante ( $180/\pi$ ). Si se sustituye esta relación en la Ec. (99) para evaluar el sistema en radianes, se obtiene:

$$\delta_{rad}(k) \left( \frac{180}{\pi} \right) = \left( \frac{180}{\pi} \right) \left( C_{i,1}e_{\theta,rad}(k) + C_{i,2} \left[ \frac{e_{\theta,rad}(k) - e_{\theta,rad}(k-1)}{T_s} \right] \right). \quad (100)$$

Dado que el factor escalar multiplica de manera global a ambos lados de la ecuación diferencial, se cancela algebraicamente. Este procedimiento demuestra que los coeficientes consecuentes  $C_{i,1}$  y  $C_{i,2}$  son parámetros invariantes respecto a la escala de las unidades angulares, siempre que las variables de entrada y salida mantengan correspondencia dimensional. Por consiguiente, las matrices  $\mathbf{A}_i$  construidas a partir de estos parámetros son válidas para el análisis en el espacio de estados en radianes, respaldando la legitimidad de la prueba LMI obtenida.

#### 4.4. Ambiente de simulación

En esta sección se describe el ambiente de simulación desarrollado para evaluar el comportamiento del sistema. Dicho entorno se fundamenta en el modelo cinemático de bicicleta bajo

la geometría de Ackermann descrito matemáticamente en la Sección 2.7. Es importante recalcar que no se incorporaron modelos dinámicos complejos porque la premisa medular de esta tesis es resolver el seguimiento de trayectorias mediante enfoques libres de modelo dinámico, donde la red neurodifusa IT2-ANFIS OSTRa compensa estas incertidumbres físicas no modeladas a partir del aprendizaje puro de los datos de E/S. Bajo este enfoque cinemático, el ambiente de simulación fue concebido como una herramienta analítica complementaria para modelar y evaluar el control longitudinal acoplado a la dinámica lateral. Esto permitió expandir el alcance de la investigación comprobando el control sobre la propia velocidad, y a su vez sortear temporalmente las limitaciones mecatrónicas de actuación autónoma sobre el acelerador del TR físico. El simulador permite además evaluar el desempeño frente a la inyección intencional de ruido en los sensores virtuales a lo largo de diferentes perfiles de trayectoria. A continuación, se presentan los elementos clave que componen el ambiente y los resultados obtenidos.

Para obtener más información sobre el código desarrollado, librerías y requisitos necesarios para el ambiente de simulación, puede consultarse el repositorio de GitHub disponible en [92]. La Figura 62 muestra la interfaz del simulador, destacando los diferentes componentes que permiten la configuración de las trayectorias, la selección de los algoritmos de control y la visualización de los resultados.

1. **Selección de Trayectoria:** Este menú desplegable permite al usuario elegir entre varias rutas predefinidas que seguirá el vehículo durante la simulación. Las rutas representan diferentes escenarios para probar el rendimiento de los algoritmos de control. Las rutas disponibles incluyen:
  - (a) **Ruta 1:** Una trayectoria en línea recta simple, ideal para pruebas de control básicas y para garantizar que el sistema pueda mantener una trayectoria recta.
  - (b) **Ruta 2:** Tres surcos con giros de radio de 2 m entre las filas, diseñado para representar un entorno agrícola típico con giros estrechos.
  - (c) **Ruta 3:** Tres surcos con curvas de radio de 3 m entre las filas y tramos rectos más largos. Una variación del recorrido anterior con curvas más amplias y tramos rectos extendidos, lo que proporciona un escenario más desafiante que combina el control en línea recta y seguimiento de curvas.
  - (d) **Ruta 4:** Lemniscata (trayectoria en forma de ocho) trayectoria que se utiliza para probar la capacidad del vehículo para manejar curvas continuas con radios cambiantes, lo que agrega complejidad a la tarea de control.
2. **Arquitectura de Control (Lateral):** Este elemento permite al usuario seleccionar, bajo el rubro de arquitectura de control, entre varios métodos para gestionar la dirección del vehículo. Cada método aplica diferentes estrategias de control para mantener el vehículo en la trayectoria deseada. El ángulo de dirección está limitado entre  $\pm$

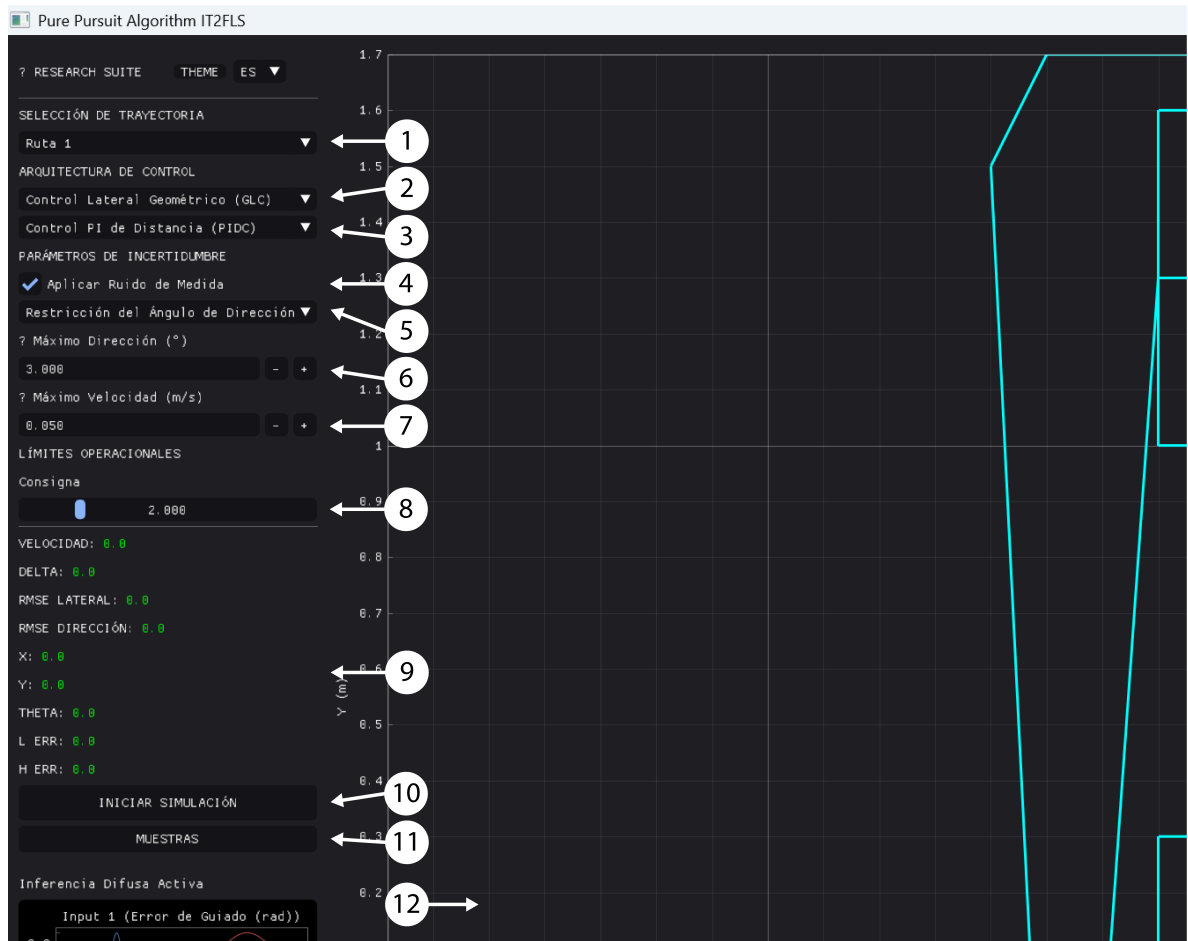


Figura 62: Interfaz del simulador desarrollado para el TR, mostrando los componentes que la integran.

0.698132 radianes (o  $\pm 40$  grados) para simular limitaciones realistas del vehículo. Los controles disponibles se describen a continuación:

- (a) **Control Lateral Geométrico (GLC):** Este método calcula el ángulo de dirección considerando la relación geométrica entre la posición actual del vehículo y un punto de referencia en la ruta. La distancia de anticipación  $l_a$  es una medida fija que determina el objetivo del vehículo, asegurando que siempre esté a una distancia mayor que esta constante, lo que permite al controlador anticipar la trayectoria de aproximación. Al ajustar el ángulo de dirección para alinear el rumbo del vehículo con este punto, el GLC minimiza las desviaciones laterales de la trayectoria deseada, proporcionando un comportamiento de seguimiento de trayectoria suave y estable. Este enfoque se utiliza para trayectorias simples y requiere recursos computacionales mínimos, lo que facilita su aplicación para el control en tiempo real de vehículos. El ángulo de dirección  $\delta$  viene dado por:

$$\delta(k) = \tan^{-1} \left( \frac{2L \sin(e_\theta(k))}{l_d(k)} \right), \quad (101)$$

donde:

- $L$  es la distancia entre ejes de las ruedas,
- $e_\theta(k)$  es el error de cabeceo entre la dirección del vehículo y el punto objetivo,
- $l_d(k)$  es la distancia actual entre el vehículo y el punto de referencia en la ruta.

- (b) **Controlador Basado en el Error (EBC):** En este método, el ángulo de dirección se establece directamente en función del error de cabeceo  $\theta_e$  sin ninguna ganancia o modificación. Este enfoque es sencillo y se basa únicamente en la diferencia angular entre el cabeceo del vehículo y la trayectoria deseada. La ecuación es:

$$\delta(k) = e_\theta(k) \quad (102)$$

- (c) **T1FLC con 2MF:** Este controlador de lógica difusa de tipo 1 (T1FLC, por sus siglas en inglés) utiliza dos MF en forma de campana generalizada para mapear el error de cabeceo y su derivada a un ángulo de dirección. El controlador se obtiene automáticamente utilizando ANFIS de MatLab [9] con un RMSE de 0,0386 radianes y 100 épocas de entrenamiento. Los datos de entrenamiento se obtuvieron de una simulación sin ruido en Ruta 2, utilizando el GLC para calcular  $\delta$  y el control de distancia PI (PIDC, por sus siglas en inglés) para calcular la velocidad  $v$ . El sistema de control utiliza conjuntos difusos y las reglas del tipo T-S se basan en:

$$\delta(k) = \sum_{i=1}^N \tilde{w}_i \cdot \left( \left( \sum_{j=1}^J C_{i,j} \cdot x_j(k) \right) + C_i \right), \quad (103)$$

donde:

- $\tilde{w}_i$  es el  $i$ -ésimo nivel de disparo normalizado,
- $C_{ij}$  son los parámetros consecuentes  $i$ -ésima regla,
- $x_j(k)$  son las entradas (ej., error de cabeceo y su derivada).

- (d) **IT2FLC con 2MF:** Los parámetros y el conjunto de datos utilizados para entrenar el T1FLC son los mismos que para IT2FLC; sin embargo, la principal distinción es que se utiliza IT2-ANFIS OSTRÁ para el modelado. IT2FLC está diseñado para manejar niveles más altos de incertidumbre en el sistema mediante el uso de conjuntos difusos con un rango de posibles valores de pertenencia, en lugar de valores fijos como en T1. Esta flexibilidad permite una respuesta más robusta a

las variaciones en los datos de entrada.

Este modelo utiliza dos IT2FS para cada entrada, dando como resultado una base de reglas simple y, en consecuencia, un bajo costo computacional. OSTRa se emplea para garantizar que las salidas difusas de IT2 se transformen con precisión en un ángulo de dirección utilizable. Esto diferencia al enfoque IT2FLC del tradicional T1FLC, dotando al algoritmo de capacidades mejoradas para lidiar con perturbaciones e incertidumbres dinámicas en el control direccional. El ángulo de dirección del controlador se calcula utilizando:

$$\delta(k) = \sum_{i=1}^N \text{OSTRA}(\underline{w}_i, \bar{w}_i) \cdot \left( \left( \sum_{j=1}^J C_{i,j} \cdot x_j(k) \right) + C_i \right), \quad (104)$$

donde:

- $\text{OSTRA}(\cdot, \cdot)$  representa la función de reducción de tipo que mapea el intervalo difuso a un valor discreto de T1,
- $\underline{w}_i$  y  $\bar{w}_i$  representan la fuerza de disparo inferior y superior de la  $i$ -ésima regla, respectivamente.

3. **Arquitectura de Control (Longitudinal):** Esta función proporciona, como segundo parámetro de la arquitectura de control, tres algoritmos para ajustar la velocidad del vehículo durante la simulación: Control proporcional integral (PI) de Distancia (PIDC), Control PI de Velocidad (PIVC) e IT2FLC con 2MF. Cada método se detalla a continuación.

(a) **Control PI de Distancia (PIDC):** En este controlador PI en su forma integral discreta [93], la velocidad del vehículo se ajusta en función del error entre la distancia al objetivo y el valor de anticipación. La idea principal es reducir la velocidad a medida que el vehículo se acerca al objetivo para mejorar la precisión del seguimiento. La ecuación para el PIDC es:

$$v(k) = K_p (l_d(k) - l_a) + K_i \sum_{m=0}^k (l_d(m) - l_a) T_s, \quad (105)$$

donde:

- $v(k)$  es la consigna de velocidad del vehículo en la iteración discreta actual  $k$ ,
- $K_p$  y  $K_i$  son las ganancias proporcional e integral, respectivamente,
- $l_d(k)$  y  $l_a$  representan la distancia actual hacia la ruta y la distancia de anticipación constante. Su diferencia  $(l_d(k) - l_a)$  corresponde empíricamente al error de control de distancia posicional del vehículo;

- $m$  es el índice temporal utilizado para la sumatoria integral, que captura los valores históricos del error desde el inicio de la simulación ( $m = 0$ ) hasta el instante actual ( $k$ ),
- $T_s$  es el tiempo de muestreo, que indica el intervalo de tiempo discreto entre actualizaciones de la ley de control (0.1 s).

(b) **Control PI de Velocidad (PIVC):** El control de velocidad proporcional, integral discreto (PIVC, por sus siglas en inglés) [93] se basa en una velocidad objetivo y utiliza la diferencia entre la velocidad deseada y la velocidad actual para ajustar la aceleración del vehículo. Este método es útil para mantener una velocidad constante o garantizar una aceleración/desaceleración suave. La ecuación de este controlador es:

$$v(k) = K_p (v_t - v(k-1)) + K_i \sum_{m=0}^k (v_t - v(m-1)) T_s, \quad (106)$$

donde:

- $v(k)$  es la velocidad del vehículo en la iteración discreta actual  $k$ ,
- $v(k-1)$  es la velocidad que el vehículo tenía en el instante de tiempo inmediatamente anterior,
- $m$  representa el índice temporal histórico (donde  $m = 0, 1, \dots, k$ ), indicando que la sumatoria considera todos y cada uno de los eventos desde el inicio de la simulación  $m = 0$  hasta la iteración actual  $k$ ,
- $v_t$  es la velocidad objetivo final (*set point*) fijada por el usuario en la interfaz.

Las ganancias establecidas para los controladores de velocidad PIDC y PIVC fueron establecidas empíricamente como  $K_p = 1.0$  y  $K_i = 0.1$ . Esto se debió a que el enfoque principal de esta investigación radica en el control de la dirección mediante lógica difusa (IT2FLC), por lo que las ganancias longitudinales se ajustaron únicamente para proveer una aceleración amortiguada que mantuviera el sistema numéricamente estable durante las pruebas.

(c) **IT2FLC con 2MF:** Este método de control es análogo al modelo IT2FLC descrito anteriormente para el ángulo de dirección, pero emplea un conjunto de parámetros entrenados específicamente para el control de velocidad. Los datos de entrenamiento se obtuvieron de una simulación inicial sin ruido geométrico sobre la Ruta 2, utilizando GLC para calcular el ángulo de dirección  $\delta$  y el PIDC para calcular la velocidad  $v(k)$ . Las variables de entrada difusa  $x_j$  para este modelo IT2FLC son el error de distancia posicional ( $l_d(k) - l_a$ ) y su derivada, con el objetivo de inferir la velocidad de avance del vehículo como salida resultante. La métrica matemática del controlador se sustenta sobre la Ec. (104).

4. **Aplicar Ruido de Medida:** Esta casilla de verificación habilita el marco de incertidumbre de la simulación. El realismo del simulador no reside únicamente en añadir variaciones estadísticas a las señales, sino en la sinergia entre las incertidumbres extrínsecas

(sensores) y las restricciones intrínsecas (dinámica del vehículo). Cuando está habilitada, se agrega ruido aleatorio ( $\pm 2$  cm) a la posición  $x, y$  del vehículo para replicar las imprecisiones típicas de un sistema RTK-GNSS. Sin embargo, para profundizar en el realismo físico, el simulador activa el modelado de las limitaciones mecánicas del TR real, tales como los límites de saturación en el torque de dirección y las restricciones en la tasa de cambio de velocidad (aceleración limitada). Al combinar el ruido sensorial con estos modelos de respuesta de actuadores reales, el simulador ofrece un entorno que permite validar la capacidad de recuperación de los algoritmos de control ante las perturbaciones.

5. **Parámetros de Incertidumbre (Modelo de dirección):** Este menú desplegable permite al usuario elegir la representación del modelo de dirección al realizar simulaciones con ruido. Hay tres opciones disponibles: restricción del ángulo de dirección, modelo en función de transferencia y un T1FLS. Cada uno de estos modelos ajusta el comportamiento del volante en la simulación para reflejar una respuesta más realista.

(a) **Restricción del Ángulo de Dirección:** Cuando se selecciona esta opción, se habilita un deslizador en la interfaz de usuario (*slider*) denominado: 7. “Delta Máx. de Dirección ( $^{\circ}$ )”, el cual permite ajustar el cambio máximo del ángulo de dirección ( $\Delta\delta$ ) en grados por cada iteración. Este modelo de saturación lineal limita el incremento de la dirección en cada paso de tiempo, asegurando que el seguimiento automático de la trayectoria se mantenga dentro de las capacidades cinemáticas reales del vehículo.

(b) **Modelo en Función de Transferencia:** Esta opción utiliza una función de transferencia de primer orden derivada de los datos recopilados del sistema de dirección del TR real. Se calcula un error entre el nuevo ángulo de referencia y el ángulo de dirección actual. El error se multiplica por la ganancia  $P K_p=41.3006$ , y la señal de control resultante  $u(k)$  limitada a  $[-255, 255]$ , se utiliza para actualizar el nuevo ángulo de dirección en función de la siguiente función de transferencia:

$$\frac{\Delta(z)}{U(z)} = \frac{0.001911}{1 - 0.9989z^{-1}} \quad (107)$$

Aplicando la transformada  $z$  inversa y despejando la Ec. (107), obtenemos la siguiente ecuación que nos permite calcular la referencia para el nuevo ángulo de dirección:

$$\delta(k) = 0.9989 \cdot \delta(k-1) + 0.001911 \cdot u(k) \quad (108)$$

Esta función se actualiza cada  $20\text{ ms}$ , que es el tiempo de muestreo establecido en el TR real. Debido a que la dinámica de la dirección posee una constante de tiempo menor, el bucle de control principal que opera con un período de  $100\text{ ms}$ ,

llama a esta función de actualización cinco veces en cada iteración; esto permite simular con mayor precisión la respuesta temporal y las latencias propias del servomecanismo de dirección.

(c) **Modelo T1FLS:** Este modelo se deriva de igual forma de datos reales recopilados del volante del TR. Utilizando ANFIS de MatLab [9], se modelaron los datos de entrada y salida, donde las entradas son  $\delta(k)$  y  $u(k)$ , y la salida es el nuevo ángulo de dirección  $\delta(k+1)$ . De manera similar al modelo de función de transferencia, se utiliza un controlador PD para calcular la señal de control. La señal de control  $u(k)$  se limita al rango  $[-1, 1]$ , y luego se asigna a un rango de  $[-255, 255]$  para que coincida con el rango de entrada del modelo difuso. Luego, las entradas pasan a través del sistema difuso para obtener el nuevo ángulo de dirección. La ecuación de actualización se repite cinco veces dentro del ciclo principal de 100 ms, y cada iteración ocurre cada 20 ms con el objetivo de simular el tiempo de muestreo real del hardware del TR.

6. **Delta Máx. de Dirección (grados):** Este campo de entrada permite al usuario ajustar el cambio máximo permitido en el ángulo de dirección (en grados) para cada paso de simulación. Solo se habilita cuando se selecciona “Restricción del Ángulo de Dirección” en la opción 5. “Parámetros de Incertidumbre”. Al ajustar este valor, los usuarios pueden observar cómo responde el vehículo a cambios de dirección más suaves o más abruptos, lo que proporciona información sobre el comportamiento de los algoritmos de control de dirección bajo diferentes dinámicas de dirección.
7. **Delta Máx. de Velocidad (m/s):** Este campo de entrada permite al usuario definir el cambio máximo permitido en la velocidad (en metros por segundo) para cada paso de simulación. Se activa cuando se habilita la simulación con ruido, lo que permite al usuario simular restricciones realistas de aceleración y desaceleración durante el movimiento del vehículo.
8. **Límites Operacionales (Consigna):** Un deslizador en la interfaz (*slider*) que permite al usuario ajustar la consigna de velocidad máxima del vehículo durante la simulación. El rango de velocidad se puede ajustar de 0.1 m/s hasta 10 m/s, lo que proporciona flexibilidad para evaluar el rendimiento del seguimiento de trayectorias bajo diversos escenarios, desde maniobras lentas hasta desplazamientos de mayor velocidad. Esta función permite observar la respuesta del vehículo ante cambios en la dinámica longitudinal, asegurando un comportamiento consistente en el rango operacional definido.
9. **Indicadores:** Esta sección proporciona información en tiempo real sobre el estado del vehículo durante la simulación. Muestra parámetros esenciales como la posición actual ( $x, y$ ), la orientación ( $\theta$ ), la velocidad ( $v$ ), el ángulo de dirección ( $\delta$ ), el error lateral ( $e_l$ ), el error de cabeceo ( $e_\theta$ ), así como el RMSE tanto para el error lateral como para el error de cabeceo. Estos indicadores ofrecen información sobre el rendimiento del vehículo y la precisión a la hora de seguir la ruta deseada.

10. **Iniciar Simulación:** Este botón inicia la simulación utilizando los parámetros y configuraciones establecidos por el usuario. Activa el comportamiento de seguimiento de la trayectoria del vehículo en función de la ruta seleccionada, el control de la dirección, el control de velocidad y cualquier configuración adicional, como el ruido, lo que permite al usuario observar cómo se comporta el vehículo en las condiciones especificadas.
11. **Muestras:** Esta opción ejecuta todas las combinaciones con ruido existentes de manera consecutiva para evaluar el rendimiento y la solidez de cada método de control en las distintas rutas. Proporciona información sobre el comportamiento y la precisión del seguimiento de la trayectoria del vehículo mediante la generación de múltiples registros, lo que permite el análisis estadístico de los errores (laterales y de cabeceo), lo que resulta útil cuando se habilita el ruido.

De manera automática al presionar el botón, se ejecutan una por una todas las combinaciones disponibles en el simulador, generando una carpeta llamada *multi\_results* donde se almacenarán los resultados. Los resultados se organizan primeramente por el número de ruta, posteriormente en carpetas que contienen las combinaciones de esa ruta por modelo del volante y control de velocidad utilizado, para finalmente encontrar imágenes y archivos de los resultados por control del volante.

12. **Área de simulación:** Esta área representa visualmente la ruta seleccionada del vehículo y su trayectoria en tiempo real durante toda la simulación. Se actualiza dinámicamente para mostrar los movimientos del vehículo, los ajustes de dirección y los cambios de velocidad a medida que avanza por la ruta. El gráfico permite a los usuarios monitorear la precisión con la que el vehículo sigue la trayectoria predefinida en el entorno simulado, lo que facilita el análisis comparativo entre los distintos métodos de control evaluados.

A continuación, se presentarán los resultados obtenidos de las diversas pruebas realizadas con el simulador para validar la efectividad del método IT2-ANFIS OSTRÁ en el seguimiento de trayectorias del TR. Dado que el número total de pruebas realizadas es considerablemente alto, alcanzando un total de 144 combinaciones entre diferentes rutas, controles del volante, controles de velocidad, y modelos del volante, es necesario optimizar la manera en que se presentan estos resultados. Para abordar este desafío, los resultados de las simulaciones con ruido se muestran a través de tablas comparativas que permiten visualizar las diferencias en desempeño entre los diferentes algoritmos. De este modo, se busca proporcionar una visión clara y comprensible de los resultados sin abrumar con un exceso de información. Los resultados obtenidos a partir de las simulaciones se presentan en cuatro tablas que corresponden a las rutas evaluadas: línea recta (Tabla 35), tres surcos con radio de 2 m (Tabla 36), tres surcos con radio de 3 m (Tabla 37) y lemniscata (Tabla 38). Cada tabla muestra los valores del RMSE del error lateral y de cabeceo, comparando diferentes modelos del volante y métodos de control.

Tabla 35: RMSE del error lateral y de cabeceo en la Ruta 1 bajo condiciones de ruido

Modelo del volante	Control de velocidad	Control del volante			
		GLC	EBC	T1FLC 2MF	IT2FLC 2MF
Restricción del ángulo de dirección	PIDC	0.624 m 3.2°	0.574 m 2.25°	0.548 m 1.8°	0.539 m 1.75°
	PIVC	– –	0.707 m 3.17°	0.678 m 3.19°	0.663 m 2.46°
	IT2FLC 2MF	0.480 m 2.69°	0.436 m 1.35°	0.412 m 1.07°	0.400 m 1.04°
Modelo en función de transferencia	PIDC	0.624 m 2.29°	0.566 m 2.29°	0.539 m 1.79°	0.529 m 1.69°
	PIVC	0.762 m 3.25°	0.693 m 2.95°	0.663 m 2.39°	0.648 m 2.22°
	IT2FLC 2MF	0.480 m 1.36°	0.424 m 1.32°	0.400 m 1.02°	0.387 m 0.97°
T1FLS	PIDC	0.469 m 0.96°	0.436 m 1.14°	0.424 m 0.85°	0.412 m 0.84°
	PIVC	0.574 m 1.33°	0.539 m 1.38°	0.520 m 1.12°	0.510 m 1.11°
	IT2FLC 2MF	0.346 m 0.56°	0.332 m 0.66°	0.332 m 0.5°	0.316 m 0.51°

Tabla 36: RMSE del error lateral y de cabeceo en la Ruta 2 bajo condiciones de ruido

Modelo del volante	Control de velocidad	Control del volante			
		GLC	EBC	T1FLC 2MF	IT2FLC 2MF
Restricción del ángulo de dirección	PIDC	0.387 m 3.28°	0.387 m 4.21°	– –	0.332 m 2.91°
	PIVC	– –	0.400 m 4.48°	– –	0.361 m 3.1°
	IT2FLC 2MF	0.346 m 3.32°	0.346 m 4.54°	– –	0.300 m 3.1°
Modelo en función de transferencia	PIDC	0.387 m 2.91°	0.374 m 4.2°	– –	0.332 m 2.91°
	PIVC	0.400 m 3.08°	0.400 m 4.44°	– –	0.346 m 3.08°
	IT2FLC 2MF	0.346 m 2.98°	0.346 m 4.61°	– –	0.300 m 3.17°
T1FLS	PIDC	0.300 m 1.33°	0.300 m 1.97°	0.283 m 1.15°	0.283 m 1.23°
	PIVC	0.316 m 1.37°	0.300 m 2.01°	0.283 m 1.21°	0.300 m 1.25°
	IT2FLC 2MF	0.300 m 1.35°	0.283 m 2.03°	0.265 m 1.273°	0.265 m 1.271°

Tabla 37: RMSE del error lateral y de cabeceo en la Ruta 3 bajo condiciones de ruido

Modelo del volante	Control de velocidad	Control del volante			
		GLC	EBC	T1FLC 2MF	IT2FLC 2MF
Restricción del ángulo de dirección	PIDC	0.346 m 2.64°	0.316 m 2.9°	0.283 m 2.03°	0.283 m 1.93°
	PIVC	– –	0.332 m 3.01°	– –	0.283 m 2.01°
	IT2FLC 2MF	0.316 m 2.77°	0.300 m 3.13°	0.265 m 2.96°	0.265 m 2.06°
Modelo en función de transferencia	PIDC	0.332 m 2.02°	0.316 m 2.87°	– –	0.283 m 1.93°
	PIVC	0.332 m 2.08°	0.316 m 2.95°	– –	0.283 m 1.96°
	IT2FLC 2MF	0.300 m 2.09°	0.300 m 3.17°	– –	0.245 m 2.05°
T1FLS	PIDC	0.265 m 0.94°	0.265 m 1.35°	0.245 m 0.81°	0.245 m 0.83°
	PIVC	0.265 m 0.95°	0.265 m 1.37°	0.245 m 0.8°	0.245 m 0.84°
	IT2FLC 2MF	0.245 m 1.01°	0.245 m 1.46°	0.224 m 0.87°	0.224 m 0.89°

Tabla 38: RMSE del error lateral y de cabeceo en la Ruta 4 bajo condiciones de ruido

Modelo del volante	Control de velocidad	Control del volante			
		GLC	EBC	T1FLC 2MF	IT2FLC 2MF
Restricción del ángulo de dirección	PIDC	– –	0.775 m 4.43°	– –	0.748 m 3.35°
	PIVC	– –	– –	– –	– –
	IT2FLC 2MF	– –	0.735 m 4.03°	– –	0.721 m 3.57°
Modelo en función de transferencia	PIDC	0.787 m 3.13°	0.742 m 2.91°	– –	0.735 m 2.09°
	PIVC	0.794 m 3.29°	0.755 m 3°	– –	0.742 m 2.17°
	IT2FLC 2MF	0.748 m 2.96°	0.714 m 2.78°	– –	0.700 m 1.96°
T1FLS	PIDC	0.781 m 2.1°	0.755 m 1.75°	– –	0.748 m 1.53°
	PIVC	0.787 m 2.11°	0.762 m 1.79°	– –	0.762 m 1.54°
	IT2FLC 2MF	0.748 m 1.97°	0.721 m 1.66°	– –	0.721 m 1.44°

En todas las tablas se puede observar que, en la mayoría de los casos, el método IT2FLC aplicado al control del volante logra los mejores resultados frente a los demás métodos evaluados. Además, el menor error para cada ruta, considerando todos los métodos, se obtuvo utilizando IT2FLC tanto para el control del volante como para el control de velocidad, junto con el modelo T1FLS en el volante. Sin embargo, en la Ruta 3, si bien el menor RMSE del error lateral es igual para ambos, el método IT2FLC en el control del volante supera a T1FLC por 0.02 grados en el error de cabeceo. Estas diferencias reflejan el desempeño del método IT2FLC en el manejo de la incertidumbre y la capacidad para minimizar el error durante el seguimiento de trayectorias, destacando su superioridad frente a otros métodos. También, cabe mencionar que a pesar de que los controladores difusos fueron entrenados con datos que no presentaban ruido en sus mediciones y por lo tanto, obtenían resultados muy precisos, no representó una limitante para mantener el seguimiento de trayectoria ante las perturbaciones contenidas en los datos de las simulaciones. Con este simulador se proporciona una herramienta de código abierto que permite simular vehículos en diferentes rutas, facilitando así el desarrollo y validación de estrategias de control automático.

## 5. Conclusiones

La validación experimental documentada en esta tesis demuestra que el marco neurodifuso IT2-ANFIS OSTRa constituye una arquitectura robusta para la generación algorítmica y el control direccional de sistemas mecatrónicos operando bajo condiciones físicas restrictivas e incertidumbre no modelada.

Durante la experimentación directa sobre la plataforma TR, se comprobó la viabilidad del enfoque de aprendizaje a partir de señales de entrada/salida. El IT2FLC generado, tras ser entrenado con datos del propio algoritmo de referencia (GLC), logró abstraer la dinámica no lineal del vehículo. Los resultados muestran que el modelo inferido fue capaz de superar el desempeño de su algoritmo generador en pruebas de seguimiento de trayectoria, evidenciando capacidad de extrapolación al ser sometido a geometrías de ruta ausentes en su etapa de entrenamiento, como las trayectorias de lemniscata y surcos extendidos. De forma concurrente, el IT2FLC redujo consistentemente las métricas de error lateral y de cabeceo (evaluadas mediante RMSE), logrando igualar o superar el desempeño tanto del GLC como del T1FLC bajo el mismo esquema de validación experimental.

Un hallazgo analítico fundamental de este trabajo fue la validación del controlador frente a las perturbaciones físicas propias de una plataforma de bajo costo. El TR utilizado presentaba una deriva mecánica constante derivada de la leve desalineación inherente en sus neumáticos, una falla estructural que generaba sesgos laterales y requería correcciones ininterrumpidas incluso cuando el actuador general comandaba cero grados directivos. A diferencia de las metodologías como el T1FLC o los algoritmos de control netamente trigonométricos (como EBC y GLC), el modelo propuesto exhibió una superioridad evidente para absorber en sus consecuentes y compensar continuamente esta deformación física.

Adicionalmente, el desempeño observado experimentalmente en campo fue sustentado mediante rigor teórico. Al proyectar los submodelos locales de la inferencia difusa mediante aproximaciones en espacio de estados discreto derivadas del modelo de Ackermann, se calculó la optimización convexa por desigualdad lineal matricial. El resultado demostró formalmente que la ley de control interpolada por la red IT2-ANFIS OSTRa es global y asintóticamente estable. Esto ampara y valida formalmente el comportamiento convergente del controlador experimental.

A nivel algorítmico, otra de las aportaciones centrales de esta tesis es la estructuración matemática y validación informática de la plataforma IT2-ANFIS OSTRa en sí misma y de su TRA. El algoritmo OSTRa facilitó aprovechar la FOU característica de los IT2FLS, disminuyendo el costo computacional iterativo tradicional asociado a TRA de este orden. Esta eficiencia algorítmica consolida su viabilidad temporal para inferencias rápidas en plataformas embebidas.

Adicionalmente, el ciclo de desarrollo se complementó con la creación de un ambiente de simulación para cinemática variable. Este simulador sirvió como un banco de pruebas analítico para sortear las limitaciones de actuación longitudinal del TR físico, permitiendo la exploración y validación de algoritmos de control de velocidad acoplados a la dinámica lateral. Dicha herramienta facilitó la evaluación del rendimiento de los controladores frente a inyección de ruido sintético en escenarios operativos que excedían el alcance del *hardware* implementado.

Con base en la validación lograda, como recomendaciones de trabajo analítico futuro, se propone la integración de nuevas formas de funciones de pertenencia a las capacidades estructurales de IT2-ANFIS OSTRÁ y el estudio de su desempeño frente a perturbaciones dinámicas con múltiples variables longitudinales combinadas. Finalmente, se recomienda la experimentación de los controladores derivados de IT2-ANFIS OSTRÁ en otras arquitecturas móviles a escala, tales como sistemas cuadrirrotores (drones) o cinemática omnidireccional mecanoide, evaluando su capacidad adaptativa en entornos de prueba de variables controladas, fortaleciendo así el campo del control convergente guiado por datos experimentales de E/S.

## Referencias bibliográficas

- [1] N. N. Karnik, J. M. Mendel, and Q. Liang, “Type-2 fuzzy logic systems,” *IEEE transactions on Fuzzy Systems*, vol. 7, no. 6, pp. 643–658, 1999.
- [2] K. Mittal, A. Jain, K. S. Vaisla, O. Castillo, and J. Kacprzyk, “A comprehensive review on type 2 fuzzy logic applications: Past, present and future,” *Engineering Applications of Artificial Intelligence*, vol. 95, p. 103916, 2020.
- [3] CHC Navigation, “Advanced Auto Steering Technology Transforming Agricultural Landscape,” <https://www.unmannedsystemstechnology.com/feature/advanced-auto-steering-technology-transforming-agricultural-landscape/>, 2025, accesado 16 de abril de 2025.
- [4] FJDynamics, “FJD AT1 Autosteering Kit,” <https://www.fjdynamics.com/product/autosteeringkit>, 2025, accesado 16 de abril de 2025.
- [5] C. Sun, J. Sun, S. Ding, Q. Li, and L. Ma, “Path tracking control in autonomous agricultural vehicles: A systematic survey of models, methods, and challenges,” *Agriculture*, vol. 15, no. 23, p. 2522, 2025.
- [6] Z. Wenyu, W. Jin, Z. Zhigang, H. Jie, H. Lian, and L. Xiwen, “Self-calibrating variable structure kalman filter for tractor navigation during bds outages,” *Nongye Jixie Xuebao/Transactions of the Chinese Society of Agricultural Machinery*, vol. 51, no. 3, 2020.
- [7] L. Emmi, R. Fernández, and P. González-de Santos, “An efficient guiding manager for ground mobile robots in agriculture,” *Robotics*, vol. 13, no. 1, p. 6, 2023.
- [8] L. A. Zadeh, “Fuzzy sets,” *Information and control*, vol. 8, no. 3, pp. 338–353, 1965.
- [9] MathWorks, “Neuro-Adaptive Learning and ANFIS,” <https://www.mathworks.com/help/fuzzy/neuro-adaptive-learning-and-anfis.html>, 2022, accesado 25 de mayo de 2022.
- [10] N. N. Karnik and J. M. Mendel, “Centroid of a type-2 fuzzy set,” *information SCIences*, vol. 132, no. 1-4, pp. 195–220, 2001.
- [11] D. Wu and J. M. Mendel, “Enhanced Karnik–Mendel algorithms,” *IEEE transactions on fuzzy systems*, vol. 17, no. 4, pp. 923–934, 2008.
- [12] D. Wu and M. Nie, “Comparison and practical implementation of type-reduction algorithms for type-2 fuzzy sets and systems,” in *2011 IEEE International Conference on Fuzzy Systems (FUZZ-IEEE 2011)*. IEEE, 2011, pp. 2131–2138.
- [13] Q. Liang and J. M. Mendel, “Interval type-2 fuzzy logic systems: theory and design,” *IEEE Transactions on Fuzzy systems*, vol. 8, no. 5, pp. 535–550, 2000.

- [14] A. Agrawal and R. K. Bhogal, “Edge detection techniques in dental radiographs (Sobel, T1FLS & IT2FLS),” in *Communication, Networks and Computing: First International Conference, CNC 2018, Gwalior, India, March 22-24, 2018, Revised Selected Papers 1*. Springer, 2019, pp. 411–421.
- [15] L. Amador-Angulo and O. Castillo, “Stabilization of a DC Motor Speed Controller Using Type-1 Fuzzy Logic Systems Designed with the Chicken Search Optimization Algorithm,” in *Intelligent and Fuzzy Techniques for Emerging Conditions and Digital Transformation: Proceedings of the INFUS 2021 Conference, held August 24-26, 2021, Volume 1*. Springer, 2022, pp. 492–499.
- [16] A. Wafa’H, A. H. Alalawin, M. Mahfouf, and O. H. Abdallah, “A dynamic type-1 fuzzy logic system for the development of a new warehouse assessment scheme,” *IEEE Access*, vol. 9, pp. 43 611–43 619, 2021.
- [17] R. Antão, *Type-2 fuzzy logic: uncertain systems’ modeling and control*. Springer, 2017.
- [18] J. M. Mendel, “Uncertainty, fuzzy logic, and signal processing,” *Signal Processing*, vol. 80, no. 6, pp. 913–933, 2000.
- [19] A. D. Torshizi, M. H. F. Zarandi, and H. Zakeri, “On type-reduction of type-2 fuzzy sets: A review,” *Applied Soft Computing*, vol. 27, pp. 614–627, 2015.
- [20] K. Duran and M. Bernal, Hector packing, “Improved iterative algorithm for computing the generalized centroid of an interval type-2 fuzzy set,” in *NAFIPS 2008-2008 Annual Meeting of the North American Fuzzy Information Processing Society*. IEEE, 2008, pp. 1–5.
- [21] J.-S. Jang, “ANFIS: adaptive-network-based fuzzy inference system,” *IEEE transactions on systems, man, and cybernetics*, vol. 23, no. 3, pp. 665–685, 1993.
- [22] D. Karaboga and E. Kaya, “Adaptive network based fuzzy inference system (ANFIS) training approaches: a comprehensive survey,” *Artificial Intelligence Review*, vol. 52, pp. 2263–2293, 2019.
- [23] S. Zhu, M. K. Tan, K. G. Lim, R. K. Y. Chin, B. L. Chua, and K. T. K. Teo, “Engine misfire fault diagnosis based on SC-ANFIS,” *Journal of Intelligent & Fuzzy Systems*, vol. Preprint, pp. 1–22, 2023.
- [24] J. Li, G. Yan, L. H. Abbud, T. Alkhalifah, F. Alturise, M. A. Khadimallah, and R. Marzouki, “Predicting the shear strength of concrete beam through ANFIS-GA-PSO hybrid modeling,” *Advances in Engineering Software*, vol. 181, p. 103475, 2023.
- [25] K. K. Dash, A. Sundarsingh, G. BhagyaRaj, V. K. Pandey, B. Kovács, and S. A. Mukarram, “Modelling of ultrasonic assisted osmotic dehydration of cape gooseberry using adaptive neuro-fuzzy inference system (ANFIS),” *Ultrasonics Sonochemistry*, vol. 96, p. 106425, 2023.

- [26] H. Moayedi and B. Le Van, “The applicability of biogeography-based optimization and earthworm optimization algorithm hybridized with ANFIS as Reliable Solutions in Estimation of Cooling Load in Buildings,” *Energies*, vol. 15, no. 19, p. 7323, 2022.
- [27] T. Takagi and M. Sugeno, “Fuzzy identification of systems and its applications to modeling and control,” *IEEE transactions on systems, man, and cybernetics*, vol. 1, pp. 116–132, 1985.
- [28] G. M. Méndez and M. De Los Angeles Hernández, “Interval type-2 ANFIS,” *Innovations in Hybrid Intelligent Systems*, pp. 64–71, 2007.
- [29] H. MonirVaghefi, M. R. Sandgani, and M. A. Shoorehdeli, “Interval type-2 adaptive network-based fuzzy inference system (ANFIS) with type-2 non-singleton fuzzification,” in *2013 13th Iranian Conference on Fuzzy Systems (IFSC)*. IEEE, 2013, pp. 1–6.
- [30] J. Tavoosi, S. Agreement, A. Amir, and M. B. Menhaj, “Stable ANFIS2 for nonlinear system identification,” *Neurocomputing*, vol. 182, pp. 235–246, 2016.
- [31] J. Tavoosi and F. Mohammadi, “A new type-II fuzzy system for flexible-joint robot arm control,” in *2019 6th International Conference on Control, Instrumentation and Automation (ICCIA)*. IEEE, 2019, pp. 1–4.
- [32] C. Chen, R. John, J. Twycross, and J. M. Garibaldi, “An extended ANFIS architecture and its learning properties for type-1 and interval type-2 models,” in *2016 IEEE International Conference on Fuzzy Systems (FUZZ-IEEE)*. IEEE, 2016, pp. 602–609.
- [33] C. Chen, R. John, J. Twycross, and J. Garibaldi, “Type-1 and interval type-2 ANFIS: A comparison,” in *2017 IEEE International Conference on Fuzzy Systems (FUZZ-IEEE)*. IEEE, 2017, pp. 1–6.
- [34] C. Chen, “A novel framework for the implementation and evaluation of type-1 and interval type-2 ANFIS,” Ph.D. dissertation, University of Nottingham, 2018.
- [35] M. Öztürk and İ. Özkol, “Comparison of modified Karnik-Mendel algorithm-based interval type-2 ANFIS and type-1 ANFIS,” *Aircraft Engineering and Aerospace Technology*, vol. 93, no. 10, pp. 1526–1532, 2021.
- [36] A. Bechar and C. Vigneault, “Agricultural robots for field operations: Concepts and components,” *Biosystems Engineering*, vol. 149, pp. 94–111, 2016.
- [37] A. Alberto-Rodríguez, M. Neri-Muñoz, J. C. Ramos-Fernández, M. A. Márquez-Vera, L. E. Ramos-Velasco, O. Díaz-Parra, and E. Hernández-Huerta, “Review of control on agricultural robot tractors,” *International Journal of Combinatorial Optimization Problems and Informatics*, vol. 11, no. 3, p. 9, 2020.

- [38] J. Deere, “Autonomous Tractor,” <https://www.deere.com/en/autonomous/>, 2024, accessed 9 de octubre de 2024.
- [39] YANMAR, “Self-Driving Robot Tractor: Yanmar Technology for Sustainable Agriculture,” [https://www.yanmar.com/global/about/technology/technical\\_review/2019/0403\\_1.html](https://www.yanmar.com/global/about/technology/technical_review/2019/0403_1.html), 2024, accessed 9 de octubre de 2024.
- [40] Case IH, “Automation,” <https://www.caseih.com/en-us/unitedstates/company/innovations/automation>, 2024, accessed 9 de octubre de 2024.
- [41] A. Stentz, C. Dima, C. Wellington, H. Herman, and D. Stager, “A system for semi-autonomous tractor operations,” *Autonomous Robots*, vol. 13, pp. 87–104, 2002.
- [42] O. C. Barawid Jr, A. Mizushima, K. Ishii, and N. Noguchi, “Development of an autonomous navigation system using a two-dimensional laser scanner in an orchard application,” *Biosystems Engineering*, vol. 96, no. 2, pp. 139–149, 2007.
- [43] R. Takai, O. Barawid Jr, and N. Noguchi, “Autonomous navigation system of crawler-type robot tractor,” *IFAC Proceedings Volumes*, vol. 44, no. 1, pp. 14 165–14 169, 2011.
- [44] J. Qu, Q. Z. Zhang, Zhe party, K. Guo, and D. Li, “Applications of Autonomous Navigation Technologies for Unmanned Agricultural Tractors: A Review,” *Machines*, vol. 12, no. 4, p. 218, 2024.
- [45] R. Galati, G. Mantriota, and G. Reina, “Robonav: An affordable yet highly accurate navigation system for autonomous agricultural robots,” *Robotics*, vol. 11, no. 5, p. 99, 2022.
- [46] E. Kayacan, E. Kayacan, H. Ramon, O. Kaynak, and W. Saeys, “Towards agrobots: Trajectory control of an autonomous tractor using type-2 fuzzy logic controllers,” *IEEE/ASME transactions on mechatronics*, vol. 20, no. 1, pp. 287–298, 2014.
- [47] E. H. Mamdani, “Application of fuzzy logic to approximate reasoning using linguistic synthesis,” *IEEE transactions on computers*, vol. 26, no. 12, pp. 1182–1191, 1977.
- [48] T. Takagi and M. Sugeno, “Fuzzy identification of systems and its applications to modeling and control,” *IEEE transactions on systems, man, and cybernetics*, vol. 1, pp. 116–132, 1985.
- [49] R. K. Pattanaik, S. K. Mohapatra, M. N. Mohanty, and B. K. Pattanayak, “System identification using neuro fuzzy approach for iot application,” *Measurement: Sensors*, vol. 24, p. 100485, 2022.
- [50] H. Jiang, F. Sabetzadeh, and C. Zhang, “An intelligent adaptive neuro-fuzzy inference system for modeling time-series customer satisfaction in product design,” *Systems*, vol. 12, no. 6, p. 224, 2024.

- [51] G. K. Yadav, M. K. Kirar, S. Gupta, and J. Rajender, “Integrating ANN and ANFIS for effective fault detection and location in modern power grid,” *Science and Technology for Energy Transition*, vol. 80, p. 34, 2025.
- [52] E. Cuevas, P. Díaz, O. Avalos, D. Zaldívar, and M. Pérez-Cisneros, “Nonlinear system identification based on ANFIS-Hammerstein model using Gravitational search algorithm,” *Applied Intelligence*, vol. 48, pp. 182–203, 2018.
- [53] S. Mahapatra, R. Daniel, D. N. Dey, and S. K. Nayak, “Induction motor control using PSO-ANFIS,” *Procedia Computer Science*, vol. 48, pp. 753–768, 2015.
- [54] F. Khoshbin, H. Bonakdari, S. H. Ashraf Talesh, I. Ebtehaj, A. H. Zaji, and H. Azimi, “Adaptive neuro-fuzzy inference system multi-objective optimization using the genetic algorithm/singular value decomposition method for modelling the discharge coefficient in rectangular sharp-crested side weirs,” *Engineering Optimization*, vol. 48, no. 6, pp. 933–948, 2016.
- [55] D. KARABOĞA and E. Kaya, “Training ANFIS by using the artificial bee colony algorithm,” *Turkish Journal of Electrical Engineering & Computer Sciences*, vol. 25, no. 3, pp. 1669–1679, 2017.
- [56] L. A. Zadeh, “The concept of a linguistic variable and its application to approximate reasoning—i,” *Information sciences*, vol. 8, no. 3, pp. 199–249, 1975.
- [57] N. N. Karnik and J. M. Mendel, “Type-2 fuzzy logic systems: type-reduction,” in *SMC’98 Conference Proceedings. 1998 IEEE International Conference on Systems, Man, and Cybernetics (Cat. No. 98CH36218)*, vol. 2. Ieee, 1998, pp. 2046–2051.
- [58] M. Nie and W. W. Tan, “Towards an efficient type-reduction method for interval type-2 fuzzy logic systems,” in *2008 IEEE International Conference on Fuzzy Systems (IEEE World Congress on Computational Intelligence)*. IEEE, 2008, pp. 1425–1432.
- [59] R. I. John and C. Czarnecki, “A type 2 adaptive fuzzy inferencing system,” in *SMC’98 Conference Proceedings. 1998 IEEE International Conference on Systems, Man, and Cybernetics (Cat. No. 98CH36218)*, vol. 2. IEEE, 1998, pp. 2068–2073.
- [60] R. John and C. Czarnecki, “An adaptive type-2 fuzzy system for learning linguistic membership grades,” in *FUZZ-IEEE’99. 1999 IEEE International Fuzzy Systems. Conference Proceedings (Cat. No. 99CH36315)*, vol. 3. IEEE, 1999, pp. 1552–1556.
- [61] G. M. Méndez and O. Castillo, “Interval type-2 TSK fuzzy logic systems using hybrid learning algorithm,” in *The 14th IEEE International Conference on Fuzzy Systems, 2005. FUZZ’05*. IEEE, 2005, pp. 230–235.
- [62] G. M. Méndez and I. Juárez, “First-order interval type-2 TSK fuzzy logic systems using a hybrid learning algorithm,” *WSEAS Transactions on Computers*, vol. 4, no. 4, pp. 378–384, 2005.

- [63] M. A. Khanesar, E. Kayacan, M. Teshnehlab, and O. Kaynak, "Extended Kalman filter based learning algorithm for type-2 fuzzy logic systems and its experimental evaluation," *IEEE Transactions on Industrial Electronics*, vol. 59, no. 11, pp. 4443–4455, 2011.
- [64] J. M. Mendel, "On the importance of interval sets in type-2 fuzzy logic systems," in *Proceedings Joint 9th IFSA World Congress and 20th NAFIPS International Conference (Cat. No. 01TH8569)*, vol. 3. IEEE, 2001, pp. 1647–1652.
- [65] H. Hong-Gui, C. Zhi-Yuan, L. Hong-Xu, and Q. Jun-Fei, "A self-organizing interval Type-2 fuzzy-neural-network for modeling nonlinear systems," *Neurocomputing*, vol. 290, pp. 196–207, 2018.
- [66] T. Kumbasar and H. Hagraş, "A gradient descent based online tuning mechanism for pi type single input interval type-2 fuzzy logic controllers," in *2015 IEEE International Conference on Fuzzy Systems (FUZZ-IEEE)*. IEEE, 2015, pp. 1–6.
- [67] M. de-los Angeles-Hernández, P. Melin, G. M. Méndez, O. Castillo, and I. López-Juárez, "A hybrid learning method composed by the orthogonal least-squares and the back-propagation learning algorithms for interval A2-C1 type-1 non-singleton type-2 TSK fuzzy logic systems," *Soft Computing*, vol. 19, no. 3, pp. 661–678, 2015.
- [68] T. Kumbasar, I. Eksin, M. Guzelkaya, and E. Yesil, "Interval type-2 fuzzy inverse controller design in nonlinear IMC structure," *Engineering Applications of Artificial Intelligence*, vol. 24, no. 6, pp. 996–1005, 2011.
- [69] T. Kumbasar, I. Eksin, M. Guzelkaya, and E. Yesil, "An inverse controller design method for interval type-2 fuzzy models," *Soft Computing*, vol. 21, pp. 2665–2686, 2017.
- [70] S. Assilian, "Artificial intelligence in control of real dynamic systems." Ph.D. dissertation, Queen Mary University of London, 1974.
- [71] W. J. Kickert and H. V. N. Lemke, "Application of a fuzzy controller in a warm water plant," *Automatica*, vol. 12, no. 4, pp. 301–308, 1976.
- [72] P. I. Corke, W. Jachimeczyk, and R. Pillat, *Robotics, vision and control: fundamental algorithms in MATLAB*. Springer, 2011, vol. 73.
- [73] R. S. Wallace, A. Stentz, C. E. Thorpe, H. P. Moravec, W. Whittaker, and T. Kanade, "First Results in Robot Road-Following." in *IJCAI*, vol. 2, 1985, pp. 1089–1095.
- [74] O. Amidi and C. E. Thorpe, "Integrated mobile robot control," in *Mobile Robots V*, vol. 1388. SPIE, 1991, pp. 504–523.
- [75] R. C. Coulter *et al.*, *Implementation of the pure pursuit path tracking algorithm*. Carnegie Mellon University, The Robotics Institute, 1992.

- [76] J. D. Padre-Ñonthe, “Diseño y desarrollo de los servomecanismos para la automatización de un tractor agrícola,” Master’s thesis, Universidad Politécnica de Pachuca, Febrero 2018.
- [77] R. Cortes-Gutiérrez, “Control difuso en la dirección de un tractor autónomo con detección y aislamiento de fallas,” Master’s thesis, Universidad Politécnica de Pachuca, Septiembre 2018.
- [78] J. G. Duran-Candelaria, “Detección de filas de cultivo para la agricultura de precisión utilizando visión artificial y técnicas neurodifusas,” Master’s thesis, Universidad Politécnica de Pachuca, Febrero 2019.
- [79] O. Alonso-Hernández, “Identificación y modelado difuso con sistemas embebidos para suministro de energía en vehículos autónomos,” Master’s thesis, Universidad Politécnica de Pachuca, Febrero 2020.
- [80] A. Alberto-Rodríguez, “Instrumentación y control inteligente de la dirección de un tractor robot,” Master’s thesis, Universidad Politécnica de Pachuca, Noviembre 2020.
- [81] M. Neri-Muñoz, J.-C. Ramos-Fernández, J. Hernández-Salazar, and M. Márquez-Vera, “Identificación de un modelo y sintonización de un controlador difuso: Aplicado a la dirección de un tractor robot,” vol. 12, pp. 1405–1410, 10 2020.
- [82] J. Power, “Implementation of ANFIS using the pyTorch framework,” <https://github.com/jfpower/anfis-pytorch>, 2021, accesado 25 de octubre de 2021.
- [83] M. Balcilar, “Adaptive neuro fuzzy inference system from scratch. all codes inside, no dependency,” <https://github.com/balcilar/Anfis-from-scratch>, 2021, accesado 25 de octubre de 2021.
- [84] UCI Machine Learning Repository, “Istanbul stock exchange,” <https://archive.ics.uci.edu/dataset/247/istanbul+stock+exchange>, 2021, accesado 25 de octubre de 2021.
- [85] UCI Machine Learning Repository, “Energy efficiency,” <https://archive.ics.uci.edu/dataset/242/energy+efficiency>, 2021, accesado 25 de octubre de 2021.
- [86] U. Machine Learning Repository, “Yacht Hydrodynamics,” <https://archive.ics.uci.edu/dataset/243/yacht+hydrodynamics>, 2021, accesado 25 de octubre de 2021.
- [87] R. H. Abiyev, O. Kaynak, and E. Kayacan, “A type-2 fuzzy wavelet neural network for system identification and control,” *Journal of the Franklin Institute*, vol. 350, no. 7, pp. 1658–1685, 2013.
- [88] J. Tavoosi and M. A. Badamchizadeh, “A class of type-2 fuzzy neural networks for nonlinear dynamical system identification,” *Neural computing and applications*, vol. 23, pp. 707–717, 2013.

- [89] M. Wakchaure, B. Patle, and A. Mahindrakar, “Application of AI techniques and robotics in agriculture: A review,” *Artificial Intelligence in the Life Sciences*, vol. 3, p. 100057, 2023.
- [90] E. Kayacan, W. Saeys, E. Kayacan, H. Ramon, and O. Kaynak, “Intelligent control of a tractor-implement system using type-2 fuzzy neural networks,” in *2012 IEEE international conference on fuzzy systems*. IEEE, 2012, pp. 1–8.
- [91] K. Tanaka and H. O. Wang, *Fuzzy control systems design and analysis: a linear matrix inequality approach*. John Wiley & Sons, 2004.
- [92] A. Alberto-Rodriguez, “Pure-Pursuit-Algorithm-IT2FLS,” <https://github.com/aar98cu/Pure-Pursuit-Algorithm-IT2FLS.git>, 2024, accesado 9 de octubre de 2024.
- [93] K. Ogata, *Discrete-time control systems*. Prentice Hall Englewood Cliffs, NJ, 1995, vol. 2.
- [94] J. C. Ramos-Fernández, V. López-Morales, M. A. Márquez-Vera, J. M. X. Pérez, and J. Suarez-Cansino, “Neuro-fuzzy modelling and stable pd controller for angular position in steering systems,” *International Journal of Automotive Technology*, vol. 22, no. 6, pp. 1495–1503, 2021.
- [95] Emlid, “Reach RS/RS+,” <https://docs.emlid.com/reachrs/>, 2024, accesado 9 de octubre de 2024.
- [96] Swift Navigation, “Piksi multi & piksi multi inertial,” <https://support.swiftnav.com/support/solutions/folders/44001200455>, 2024, accesado 9 de octubre de 2024.
- [97] S. Navigation, “Swift Navigation Binary Protocol Specification,” <https://www.swiftnav.com/resource-files/Swift%20Navigation%20Binary%20Protocol/v4.1.1/Specification/Swift%20Navigation%20Binary%20Protocol%20Specification%20v4.1.1.pdf>, 2024, accesado 9 de octubre de 2024.
- [98] I. Santos-Velázquez, “Detección de obstáculos con LIDAR mediante la técnica de background,” Master’s thesis, Universidad de Sevilla, Julio 2013.
- [99] E. H. C. Harik, “Tractor-robot cooperation: A heterogeneous leader-follower approach,” *Robotics*, vol. 12, no. 2, p. 57, 2023.

## A. Modificaciones al TR

En este apéndice se detallan las modificaciones realizadas al TR para dotarlo de la capacidad de seguir trayectorias de forma automática y cumplir con los objetivos establecidos en esta investigación. Estas modificaciones abarcan tanto el hardware como el software del sistema, e incluyen la actualización de componentes clave, la implementación de una arquitectura modular y la integración de nuevos sensores y sistemas de comunicación. El propósito principal de estas mejoras es optimizar el rendimiento del TR, aumentar su precisión y fiabilidad, y establecer una base sólida para futuros desarrollos y aplicaciones en el campo de la automatización agrícola.

### A.1. Justificación de las modificaciones

A pesar de los avances significativos alcanzados en el desarrollo previo del TR, se identificaron limitaciones que impedían alcanzar plenamente los objetivos de seguimiento de trayectorias en entornos diversos. Las configuraciones de hardware y software existentes presentaban desafíos en términos de capacidad de procesamiento, flexibilidad para integrar nuevas tecnologías y comunicación eficiente entre los componentes del sistema. Una de las principales limitaciones detectadas fue la escalabilidad y replicabilidad del proyecto. Además, la integración de todos los sistemas en un único microprocesador, no permitía una modularización y correcta distribución de los procesos que realiza cada actuador. La precisión en la percepción del entorno y en el posicionamiento también se veía comprometida por sensores con limitaciones en su exactitud y resolución.

Con el objetivo de superar estas limitaciones y potenciar las capacidades del TR, se decidió realizar una remodelación integral del sistema. Esta decisión se fundamenta en la necesidad de establecer una arquitectura modular y abierta que facilite la incorporación de tecnologías emergentes y permita una adaptación más ágil a diferentes escenarios y requerimientos. Las modificaciones implementadas se justifican por los siguientes motivos:

- **Optimización del rendimiento:** Al integrar microcontroladores de bajo costo y consumo energético, se mejora la eficiencia del sistema, permitiendo un procesamiento más rápido y una mayor autonomía operativa.
- **Flexibilidad y escalabilidad:** La adopción de una arquitectura modular facilita la integración de nuevos sensores, actuadores y algoritmos de control, promoviendo la adaptabilidad del TR a distintas condiciones de trabajo y avances tecnológicos futuros.
- **Mejora en la comunicación:** La implementación de sistemas de comunicación más robustos y eficientes, como el Bus de red de área del controlador (CAN, por sus siglas en inglés), reduce las latencias y aumenta la fiabilidad en la transmisión de datos entre los componentes, esencial para la sincronización en tiempo real.

- **Precisión en la navegación:** La incorporación de sensores de alta precisión, como el RTK-GNSS actualizado, mejora significativamente la exactitud en el posicionamiento y percepción del entorno, elemento crítico para el seguimiento de trayectorias en entornos agrícolas.
- **Reducción de costos:** Al utilizar componentes de hardware más accesibles y de código abierto, se disminuyen los costos de desarrollo y mantenimiento, facilitando la replicación del proyecto en otros contextos y promoviendo su adopción.
- **Facilidad de mantenimiento y actualización:** Una nueva arquitectura bien documentada y estandarizada simplifica el proceso de mantenimiento y futuras actualizaciones, asegurando la sostenibilidad del proyecto a largo plazo.

Estas modificaciones son fundamentales para alcanzar los objetivos propuestos en esta investigación, ya que proporcionan una plataforma sólida y versátil para la implementación de algoritmos avanzados de modelado y control basados en inteligencia computacional. Al abordar las limitaciones del sistema anterior y mejorar sus capacidades, se sientan las bases para desarrollar un TR más eficiente, preciso y adaptable, contribuyendo significativamente al avance de la automatización en el sector agrícola. A continuación, se describen en detalle las modificaciones implementadas, organizadas en las siguientes subsecciones:

- **A.1.1 Actualización del sistema de control**
- **A.1.2 Implementación de una arquitectura modular**
- **A.1.3 Integración de sistemas de comunicación avanzados**
- **A.1.4 Incorporación de sensores de alta precisión**

Estas modificaciones son fundamentales para garantizar que el TR pueda completar de forma automática trayectorias, manejando la incertidumbre y complejidad inherente al entorno. Además, la nueva configuración permite una mayor facilidad de mantenimiento y actualización, lo que es esencial para la sostenibilidad del proyecto a largo plazo. En la Figura 63 se observa el estado actual del TR y los diferentes componentes que lo integran.

#### **A.1.1. Actualización del sistema de control**

El sistema de control del TR ha experimentado varias actualizaciones a lo largo de su desarrollo, con el objetivo de mejorar su rendimiento, modularidad y capacidad para integrar nuevas tecnologías. A continuación, se describen las etapas principales de esta evolución y se presenta una comparación de los microcontroladores utilizados en cada fase, justificando las decisiones tomadas en cada cambio.

#### **Configuración inicial con LaunchPad TM4C1294**

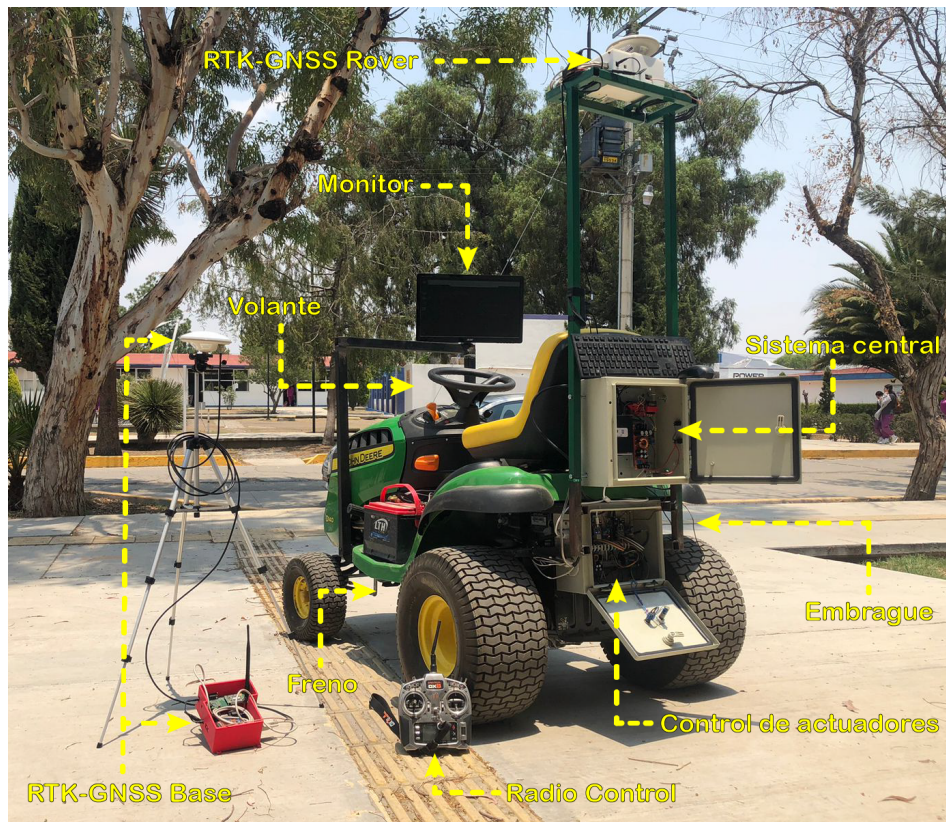


Figura 63: Configuración del sistema de seguimiento de trayectorias para un TR: integración de sensores y actuadores

En la configuración inicial, el sistema de control del TR se basaba en una LaunchPad TM4C1294 de la serie Tiva C [76]. Este microcontrolador de Texas Instruments cuenta con un procesador ARM Cortex-M4F a 120 MHz, 1 MB de memoria flash y 256 KB de SRAM. La Figura 64 muestra la placa utilizada y su configuración. Esta placa integraba todos los subsistemas del TR, incluyendo el control del volante, frenado, embrague y comunicación con la computadora personal (PC, por sus siglas en inglés). Aunque ofrecía una alta capacidad de procesamiento y múltiples interfaces de comunicación, concentrar todas las funciones en un solo microcontrolador generaba limitaciones en la modularidad y escalabilidad del sistema. Además, la complejidad del código y la carga de trabajo en un solo dispositivo dificultaban el mantenimiento y actualización del software.

### Primera modificación con microcontroladores Arduino

Para mejorar la modularidad y distribución de tareas, se realizó una primera modificación en la que se utilizaron dos microcontroladores Arduino Uno y un Arduino Mega [94], como se observa en la Figura 65. La distribución de funciones con esta modificación fue la siguiente:

- **Arduino Uno frenado y embrague:** Encargado del control del frenado y embrague.

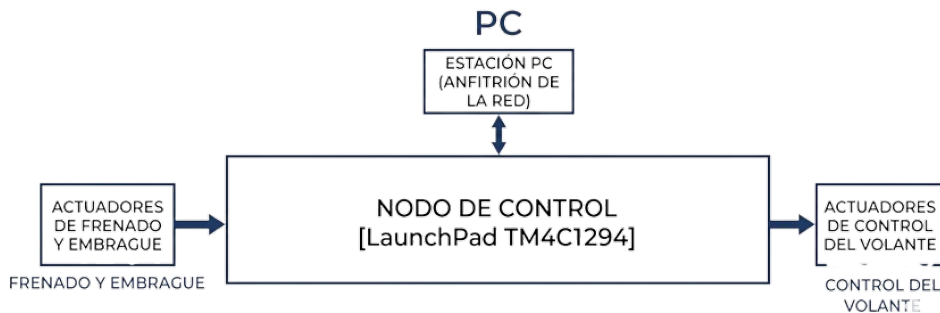


Figura 64: Configuración del microcontrolador LaunchPad TM4C1294 de la serie Tiva C.

Gestionaba los actuadores asociados y leía los sensores correspondientes.

- **Arduino Uno control del volante:** Dedicado al control del volante. Controlaba el servomecanismo de dirección y leía mediante un encoder óptico la posición del volante.
- **Arduino Mega concentrador:** Actuaba como concentrador, recopilando información de los dos Arduino Uno y comunicándose con la PC a través de comunicación serial.

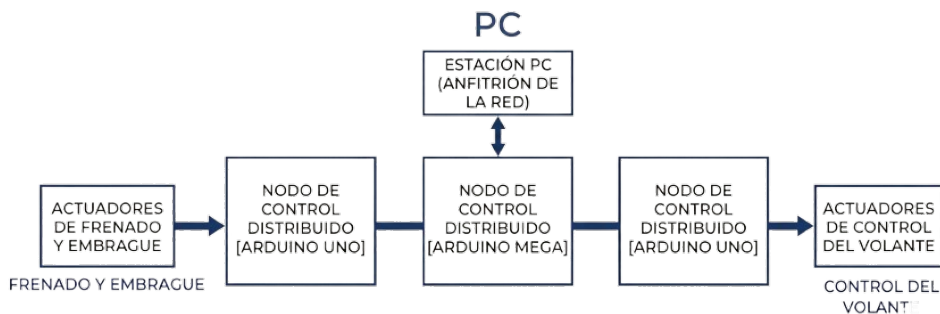


Figura 65: Configuración de los microcontroladores Arduino Uno y Mega.

La utilización de los Arduinos permitió distribuir las tareas, facilitando el desarrollo y mantenimiento del código para cada subsistema. Sin embargo, esta configuración presentaba limitaciones en cuanto a:

- **Capacidad de procesamiento:** Los microcontroladores Arduino Uno y Mega tienen frecuencias de reloj más bajas (16 MHz) y menos memoria, lo que restringe la implementación de algoritmos más complejos.
- **Falta de conectividad avanzada:** Los Arduinos carecen de interfaces de comunicación modernas como Wi-Fi o Bluetooth integrados, lo que limitaba la flexibilidad para futuras expansiones.

## Implementación actual con microcontroladores ESP32

Buscando superar las limitaciones anteriores, en la presente investigación se optó por integrar tres microcontroladores ESP32 como se muestra en la Figura 66. La distribución actual de funciones es la siguiente:

- **ESP32 frenado y embrague:** Gestiona los actuadores y sensores asociados a estos subsistemas.
- **ESP32 control del volante:** Controla el servomecanismo de dirección y proporciona retroalimentación de posición.
- **ESP32 concentrador:** Recopila información de los ESP32 anteriores y comparte información con la PC de control a través de comunicación serial. Además, ofrece posibilidades de comunicación inalámbrica si se requiere.

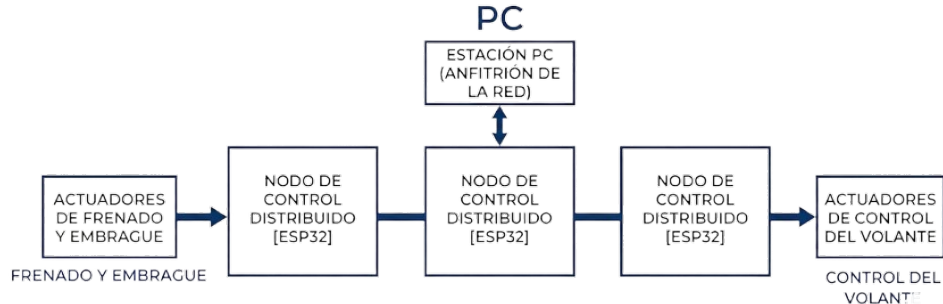


Figura 66: Configuración de los microcontroladores Arduino Uno y Mega.

Los ESP32 son microcontroladores potentes que cuentan con un procesador dual-core de 32 bits a 240 MHz, 520 KB de SRAM, Wi-Fi y Bluetooth integrados. Esto proporciona ventajas significativas en términos de capacidad de procesamiento, velocidad de comunicación y flexibilidad para integrar nuevas tecnologías.

## Comparativa de microcontroladores utilizados

Para justificar la transición entre los diferentes microcontroladores, se presenta la Tabla 39 que resume las principales características, ventajas y desventajas de cada uno.

Tabla 39: Comparativa de microcontroladores utilizados en el TR.

	TM4C1294	Arduino Uno	Arduino Mega	ESP32
Procesador	ARM Cortex-M4F	ATmega328P	ATmega2560	Xtensa Dual-Core
Frecuencia de Reloj	120MHz	16MHz	16MHz	240MHz
Memoria Flash	1MB	32KB	256KB	4MB
SRAM	256KB	2KB	8KB	520KB
Interfaces de comunicación	Ethernet, UART, I2C, SPI	UART, I2C, SPI	UART, I2C, SPI	Wi-Fi, Bluetooth, UART, I2C, SPI
Ventajas	Alto rendimiento, múltiples interfaces	Fácil programación, amplia comunidad	Fácil programación, más pines I/O	Alto rendimiento, conectividad inalámbrica, bajo costo
Desventajas	Complejidad de programación, costo	Limitado en recursos, baja velocidad	Limitado en recursos, baja velocidad	Mayor complejidad que Arduino, curva de aprendizaje

Por lo tanto la decisión de adoptar los microcontroladores ESP32 se basa en las siguientes razones:

- **Mayor capacidad de procesamiento:** Permite implementar algoritmos más complejos y procesar datos en tiempo real de manera eficiente.
- **Conectividad avanzada:** La integración de Wi-Fi y Bluetooth facilita la comunicación y posibilita futuras expansiones sin necesidad de hardware adicional.
- **Bajo costo y consumo energético:** Los ESP32 son económicos y eficientes en términos energéticos, lo que es ideal para aplicaciones móviles como el TR.
- **Modularidad y escalabilidad:** Al distribuir las tareas en tres ESP32, se mejora la modularidad del sistema, facilitando el mantenimiento y la actualización de componentes individuales.
- **Compatibilidad con Bus CAN:** Los ESP32 pueden integrarse con el Bus CAN mediante módulos adicionales, permitiendo una comunicación robusta y eficiente entre los distintos subsistemas.

### A.1.2. Implementación de una arquitectura modular

Con el objetivo de mejorar la modularidad, escalabilidad y mantenimiento del sistema de control del TR, se implementó una arquitectura modular que distribuye las funciones entre varios componentes especializados. Esta arquitectura facilita la integración de nuevos subsistemas y permite una gestión más eficiente de los recursos del sistema. La nueva arquitectura del TR se compone de tres microcontroladores ESP32, cada uno dedicado a funciones específicas, y un Arduino Nano encargado de procesar las señales del radio control (RC). Estos microcontroladores se comunican entre sí mediante un Bus CAN utilizando módulos MCP2515,

lo que garantiza una comunicación eficiente y robusta. La Figura 67 ilustra la configuración general de la arquitectura modular del sistema, cuyos módulos se describen a continuación.

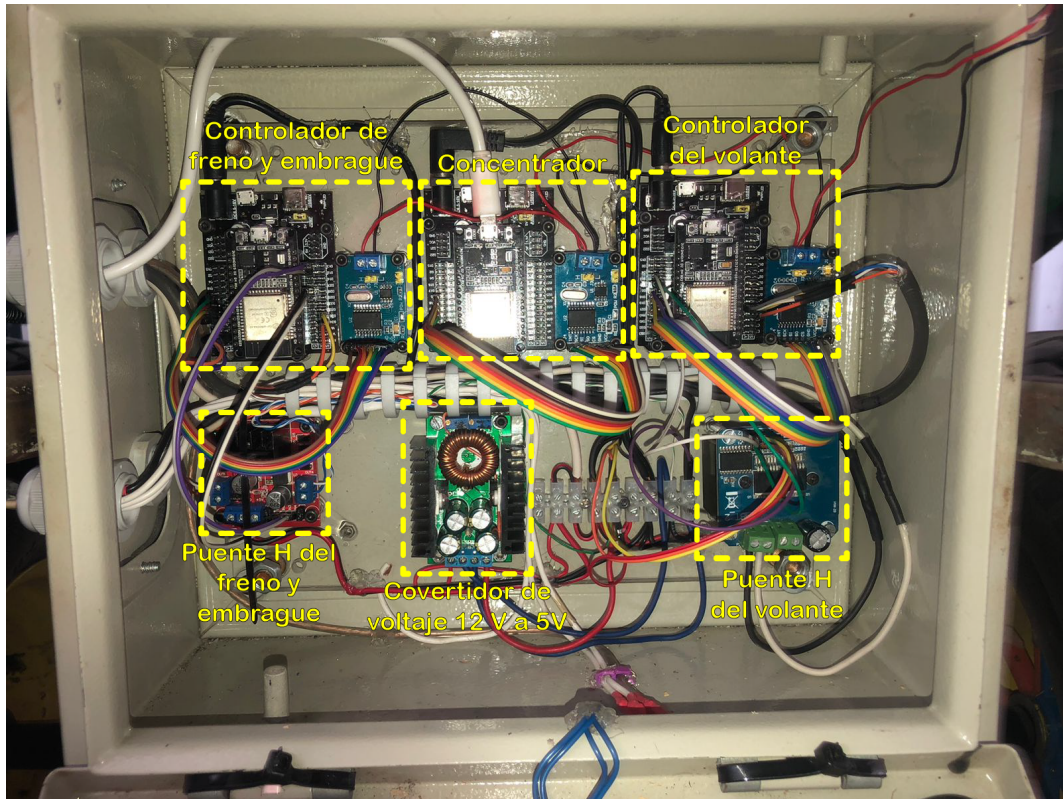


Figura 67: Arquitectura modular física del sistema de control del TR.

### Módulo ESP32 frenado y embrague

El primer ESP32, ubicado en el lado izquierdo de la Figura 67, está dedicado al control del frenado y embrague del TR. Este módulo gestiona dos actuadores lineales que operan el sistema de frenos y el embrague de velocidades. Las características principales de este módulo son:

- **Control de actuadores lineales:** Se utiliza un módulo puente H L298N para controlar la dirección de los actuadores lineales correspondientes al freno y al embrague. Este puente H permite manejar corrientes de hasta 2 A por canal, de acuerdo con los requerimientos de los actuadores utilizados.
- **Retroalimentación de posición:** Los actuadores lineales están equipados con potenciómetros de precisión que proporcionan retroalimentación de la posición del eslabón. Esta información es esencial para determinar las posiciones exactas de frenado (freno activado o desactivado) y las posiciones del embrague (Adelante, Neutral, Reversa).

- **Caracterización de posiciones:** Mediante la lectura de los potenciómetros, se realizó una caracterización de las posiciones críticas de los actuadores, estableciendo rangos de valores para cada estado operativo. Esto permite un control preciso y repetible del frenado y la selección de marchas.
- **Comunicación:** Este ESP32 se comunica con el resto del sistema a través del Bus CAN, recibiendo instrucciones y enviando información sobre el estado de los actuadores.

### Módulo ESP32 control del volante

El segundo ESP32, ubicado en el lado derecho de la Figura 67, está dedicado al control del volante del TR. Las funciones y características de este módulo incluyen:

- **Control del servomecanismo de dirección:** Utiliza un módulo puente H IBT-2 (también conocido como BTS7960) para manejar el motor eléctrico que acciona el volante. Este puente H es capaz de manejar corrientes de hasta 43 A, respondiendo a la demanda de corriente de este motor de mayor potencia.
- **Regulación de voltaje:** Se incluye un regulador de voltaje de 12 V a 5 V para alimentar pines del puente H que requieren 5 V, asegurando una alimentación estable y protegiendo los componentes sensibles. De esta forma el microcontrolador solo se encarga de enviar la señal de control.
- **Implementación de controlador proporcional, derivativo (PD):** Se implementó un controlador PD con ganancias  $K_p=0.8$ ,  $K_d=0.0001941$  y un filtro  $N=22.3799$ . Las ganancias fueron sintonizadas utilizando las herramientas de MatLab System Identification y PID Tuner. Mediante un encoder óptico incremental, se obtiene la posición actual del volante para calcular el error y aplicar la corrección necesaria a través del controlador PD.
- **Recepción de referencias:** La referencia de posición del volante es proporcionada por el ESP32 concentrador, que a su vez recibe las instrucciones desde la PC de control o el RC, dependiendo del modo de operación.
- **Retroalimentación de posición:** Como se mencionó anteriormente, el módulo obtiene la posición real del volante mediante un encoder incremental, para retroalimentar el control PD y además, enviar esta información al ESP32 concentrador a través del Bus CAN.

### Módulo ESP32 concentrador y comunicación con la PC

El tercer ESP32, ubicado en el centro de la Figura 67, actúa como concentrador y es el enlace principal entre los módulos de control de actuadores y la PC de control. Sus funciones principales son:

- **Comunicación serial con la PC:** Establece una comunicación serial con la PC de control, intercambiando información y/o comandos necesarios para el seguimiento de trayectorias.
- **Intercambio de información:**
  - **Recepción de instrucciones:** La PC envía instrucciones y referencias al ESP32 concentrador, incluyendo:
    - Referencia de posición del volante.
    - Comandos de frenado: 0 (sin freno) o 1 (frenado).
    - Selección de marcha: 0 (Neutral), 1 (Adelante), 2 (Reversa).
  - **Envío de datos a la PC:** El ESP32 concentrador envía a la PC información crítica para el control y monitoreo:
    - Modo de operación actual recibido desde el RC: 0 (Manual), 1 (Automático), 2 (Detener).
    - Referencia actual del volante.
    - Posición del embrague.
    - Posición del freno.
    - Posición real del volante.
- **Gestión de modos de operación:** El ESP32 concentrador determina el modo de operación del TR basándose en las señales recibidas del RC, y distribuye las instrucciones al módulo correspondiente.
- **Comunicación con el módulo actuador:** A través del Bus CAN, el ESP32 concentrador envía las referencias y comandos al módulo de control del volante, además de recibir información del RC y la posición del volante para retransmitirla a la PC mediante comunicación serial.

## Integración del RC y Arduino Nano

Para permitir el control manual y la selección de modos de operación, se integra un módulo receptor de RC SPEKTRUM AR8000, conectado a un Arduino Nano como se muestra en la Figura 68. Las funciones de este componente son:

- **Recepción de señales del RC:** El receptor AR8000 capta las señales inalámbricas del emisor SPEKTRUM DX8, que incluyen comandos para el control manual y la selección del modo de operación.
- **Mapeo de valores:** El Arduino Nano procesa las señales del receptor, mapeando los valores recibidos a comandos y parámetros entendibles por el sistema. Por ejemplo, los canales del RC se asignan a funciones específicas como control del volante y selección de modo.

- **Comunicación mediante Bus CAN:** El Arduino Nano envía los comandos y datos procesados al ESP32 a través del Bus CAN, utilizando un módulo MCP2515. Esto permite integrar las señales del RC en la arquitectura modular del sistema.

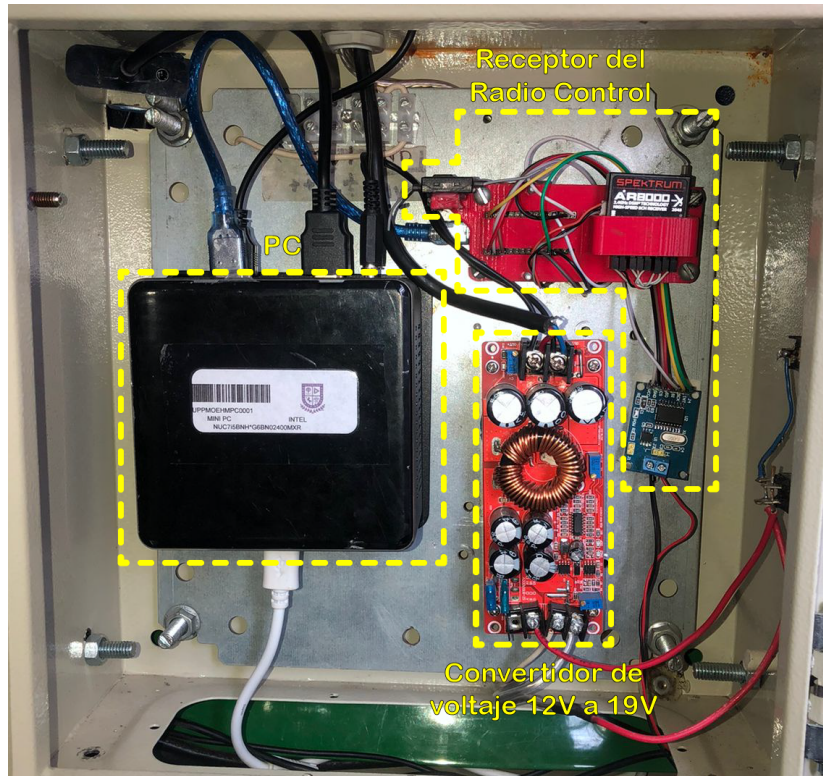


Figura 68: Arquitectura física del sistema de control del TR

La implementación de esta arquitectura modular en el TR ofrece las siguientes ventajas:

- **Flexibilidad y escalabilidad:** Facilita la incorporación de nuevos módulos y tecnologías, adaptándose a futuras necesidades y avances.
- **Mantenimiento simplificado:** Al aislar las funciones en módulos específicos, el diagnóstico y reparación de fallos es más sencillo y rápido.
- **Eficiencia operativa:** La distribución de tareas permite un uso más eficiente de los recursos del sistema, mejorando el rendimiento general.
- **Mejora en la comunicación:** El uso del Bus CAN y la estructuración modular optimiza la comunicación entre componentes, reduciendo latencias y mejorando la fiabilidad.
- **Personalización y adaptación:** La arquitectura modular facilita ajustar y personalizar los módulos según requerimientos específicos de diferentes aplicaciones o entornos de operación.

### A.1.3. Integración de sistemas de comunicación avanzados

La comunicación eficiente y fiable entre los diferentes componentes del TR es esencial para su correcto funcionamiento, especialmente en entornos agrícolas donde las condiciones pueden ser adversas. Para ello, se implementó un sistema de comunicación basado en el Bus CAN, utilizando módulos MCP2515 conectados a cada microcontrolador (los tres ESP32 y el Arduino Nano). Las ventajas de utilizar el Bus CAN incluyen:

- **Robustez y fiabilidad:** El Bus CAN es resistente al ruido y a las interferencias electromagnéticas, lo que es crucial en ambientes industriales y como este caso, agrícolas.
- **Comunicación eficiente:** Permite una comunicación rápida y priorizada entre múltiples nodos, con detección y corrección de errores.
- **Escalabilidad:** Facilita la adición de nuevos dispositivos al sistema sin cambios significativos en la infraestructura de comunicación.
- **Estandarización:** Es un estándar ampliamente utilizado en la industria automotriz y robótica, lo que facilita la integración con otros sistemas y dispositivos.
- **Sincronización Precisa:** La capacidad del Bus CAN para manejar comunicaciones en tiempo real es esencial para la coordinación precisa entre los módulos de control y los actuadores.

Se estableció una velocidad de comunicación de 125 kbps para el Bus CAN. Esta velocidad cumple con los parámetros del TR, ofreciendo una tasa constante de transmisión frente a interferencias, que son comunes en entornos agrícolas. Además, en el protocolo CAN, cada mensaje transmitido en el bus está asociado a un identificador (ID) que determina la prioridad del mensaje y permite a los nodos filtrar los mensajes que les corresponden. En el TR, se asignaron IDs específicos a cada módulo y tipo de mensaje, siguiendo una estructura organizada para facilitar la gestión y expansión futura del sistema, como se muestra en la Tabla 40.

Tabla 40: IDs asignados a los diferentes módulos del TR.

Módulo	Función	ID HEX
Arduino Nano	Comandos desde el RC	0x001
ESP32 concentrador	Distribución de comandos y referencias	0x002
ESP32 control del volante	Control y estado del volante	0x003

Los mensajes transmitidos a través del Bus CAN contienen datos específicos según el módulo y la función. A continuación se enlistan cada uno de los datos contenidos en el mensaje de cada módulo:

- **Datos Arduino Nano 0x001:**

Byte 1: Signo de referencia para el volante 0 (positivo) o 1 (negativo).

Byte 2: Valor absoluto de referencia para el volante [0:35].

Byte 3: Selección de marcha 0 (Neutral), 1 (Adelante) o 2 (Reversa).

Byte 4: Selección de frenado 0 (sin freno) o 1 (frenado).

Byte 5: Selección de modo 0 (Manual), 1 (Automático) o 2 (Detener).

- **Datos ESP32 concentrador 0x002:**

Byte 1: Signo de referencia para el volante 0 (positivo) o 1 (negativo).

Byte 2: Valor absoluto de referencia para el volante [0:35].

Byte 3: Valor decimal de referencia para el volante x100 [0:99].

Byte 4: Selección de marcha 0 (Neutral), 1 (Adelante) o 2 (Reversa).

Byte 5: Selección de frenado 0 (sin freno) o 1 (frenado).

Byte 6: Selección de modo 0 (Manual), 1 (Automático) o 2 (Detener).

- **Datos ESP32 controlador del volante 0x003:**

Byte 1: Signo de la posición del volante 0 (positivo) o 1 (negativo).

Byte 2: Valor absoluto de la posición del volante [0:35].

Byte 3: Valor decimal de la posición del volante x100 [0:99].

Byte 4: Signo de la señal de control 0 (positivo) o 1 (negativo).

Byte 5: Valor absoluto de la señal de control [0:1].

Byte 6: Valor decimal de la señal de control x100 [0:99].

Byte 7: Valor absoluto del error.

Byte 8: Valor decimal del error x100 [0:99].

## **Modos de operación y flujo de comunicación**

El TR puede operar en dos modos principales: Manual y Automático. A continuación, se describen los flujos de comunicación en cada modo de operación.

- **Modo Manual:** En este modo el operador controla el TR utilizando el RC, con la finalidad de recolectar datos para modelado. El flujo de comunicación es el siguiente:
  1. **Operador:** Usuario selecciona el modo de operación. El RC envía comandos mediante el emisor SPEKTRUM DX8.
  2. **Receptor AR8000 y Arduino Nano:** Reciben las señales del RC, procesan y mapean los valores a datos específicos.

3. **Arduino Nano:** Transmite los datos a través del Bus CAN con ID 0x001.
  4. **ESP32 frenado y embrague:** Recibe y filtra los datos del ID 0x001 relacionados con el freno y embrague.
  5. **ESP32 controlador del volante:** Recibe y filtra los datos del ID 0x001 relacionados con el volante. Además, transmite con el ID 0x003 la posición del volante, la señal de control y el error, después de aplicar el controlador PD.
  6. **ESP32 concentrador:** Recibe y concentra los datos de los IDs 0x001 y 0x003 para enviarlos mediante comunicación serial a la PC de control.
- **Modo Automático:** En el Modo Automático, la PC de control ejecuta los algoritmos de navegación y envía las instrucciones al ESP32 concentrador. El flujo de communication es el siguiente:
    1. **Operador:** Usuario selecciona el modo de operación. El RC envía comandos mediante el Emisor SPEKTRUM DX8.
    2. **Receptor AR8000 y Arduino Nano:** Reciben las señales del RC, procesan y mapean los valores a datos específicos.
    3. **Arduino Nano:** Transmite los datos a través del Bus CAN con ID 0x001.
    4. **PC de control:** Procesa los algoritmos de seguimiento de trayectorias calculando las referencias que guíen el TR hacia la ruta establecida, para enviar las referencias mediante comunicación serial al ESP32 concentrador.
    5. **ESP32 concentrador:** Recibe y filtra los datos del ID 0x001 determinando el modo de operación. Recibe mediante serial las referencias calculadas, concentrando la información para enviar los datos al Bus CAN con ID 0x002.
    6. **ESP32 controlador del volante:** Recibe y filtra los datos del ID 0x001 determinando el modo de operación. Recibe y filtra los datos del ID 0x002 relacionados con el volante. Además, transmite con el ID 0x003 la posición del volante, la señal de control y el error, después de aplicar el controlador PD.
    7. **ESP32 frenado y embrague:** Recibe y filtra los datos del ID 0x001 determinando el modo de operación. Recibe y filtra los datos del ID 0x002 relacionados con el freno y embrague.
    8. **ESP32 concentrador:** Recibe y concentra los datos del ID 0x003 para enviarlos mediante comunicación serial a la PC de control.

La implementación del Bus CAN como sistema de comunicación en el TR ha sido fundamental para mejorar la eficiencia, fiabilidad y flexibilidad del vehículo. Al detallar aspectos técnicos como los IDs asignados, la velocidad de comunicación y la estructura de los mensajes, se demuestra una planificación cuidadosa y orientada a cumplir con los objetivos de seguimiento de trayectorias.

#### A.1.4. Incorporación de sensores de alta precisión

La precisión en la percepción del entorno y en el posicionamiento es crítica para el seguimiento de trayectorias del TR, especialmente en entornos agrícolas donde se requiere un seguimiento de trayectorias con alta exactitud. Con el objetivo de mejorar el sistema de posicionamiento y superar las limitaciones del equipo anterior, se incorporó un nuevo sensor RTK-GNSS de alta precisión.

Anteriormente, el TR contaba con un receptor RTK-GNSS Emlid Reach RS+ [95], el cual proporcionaba datos de latitud, longitud y altura. Si bien este receptor ofrecía las coordenadas requeridas para ciertas aplicaciones, presentaba limitaciones al no proporcionar directamente las coordenadas cartesianas  $x$ ,  $y$ ,  $z$  relativas a la estación base. Esto añadía complejidad en el procesamiento de datos para el seguimiento de trayectorias, ya que era necesario realizar conversiones adicionales. Para mejorar la precisión y simplificar el procesamiento, se decidió reemplazar el receptor anterior por el RTK-GNSS Swift Navigation Piksi<sup>®</sup> Multi [96]. Las razones principales para la elección de este RTK-GNSS son:

- **Salida de coordenadas cartesianas:** El Piksi<sup>®</sup> Multi proporciona internamente las coordenadas  $x$ ,  $y$ ,  $z$  respecto a la base, facilitando el cálculo de distancias y trayectorias en el plano cartesiano.
- **Precisión centimétrica:** Ofrece precisión centimétrica corrigiendo errores ionosféricos y atmosféricos que afectan a los receptores GPS tradicionales. Esto permite una localización 100 veces más precisa que el GPS convencional, a un costo mucho más bajo que sistemas comparables.
- **Posicionamiento robusto:** Admite GPS L1/L2, GLONASS G1/G2, BeiDou B1/B2 y Galileo E1/E5b para mediciones y posicionamiento RTK, así como SBAS para posicionamiento submétrico en modo no RTK. El uso de múltiples constelaciones mejora el rendimiento en entornos de vista aérea desafiantes.

El módulo Piksi<sup>®</sup> Multi Rover se instaló en el TR conectándose a la PC de control mediante un puerto serial. Por otro lado, la estación base se coloca sobre un trípode en una ubicación fija marcada con la intención de mantener la misma referencia para cada una de las pruebas de navegación (Figura 63). Inicialmente, la comunicación rover-base comenzará en modo flotante (menos preciso) y pasará al modo fijo (más preciso) una vez que haya al menos 5 satélites en común entre rover y la base. Aunque el Piksi<sup>®</sup> Multi puede proporcionar grandes cantidades de información [97], como datos de magnetómetro, acelerómetro y giroscopio, los algoritmos de seguimiento de trayectorias implementados en esta investigación utilizan únicamente los datos que se enlistan adelante. Esta información permite realizar un seguimiento de la posición y movimiento del TR, facilitando el seguimiento de trayectorias.

- **Coordenadas Norte, Este y Abajo (NED):** Las coordenadas NED, son un sistema de coordenadas geográficas locales basado en el modelo elipsoidal WGS 84, que se

utiliza para la navegación de aeronaves y naves espaciales [98]. Las coordenadas obtenidas son las distancias  $x$  (E),  $y$  (N) y  $z$  (D), que representan la distancia hacia el norte, este y abajo con respecto al origen ( $x = 0$  m,  $y = 0$  m,  $z = 0$  m) que corresponde a la ubicación de la estación base. La unidad de medición en la que se obtienen estos datos es en milímetros, sin embargo, se convierten a metros para una mejor representación. En el Swift Navigation Binary Protocol [97] este mensaje tiene el código MSG\_BASELINE\_NED, 524 en decimal y 0x020c en hexadecimal.

- **Velocidades NED:** Las velocidades  $v_x$ ,  $v_y$  y  $v_z$  en mm/s, representan la relación entre el espacio recorrido y el tiempo empleado para ello en cada uno de los ejes. El código del mensaje es MSG\_VEL\_NED 526 en decimal y 0x020E en hexadecimal. Las unidades de cada medición se convierten a m/s y se calcula adicionalmente la velocidad lineal  $v_l$  utilizando la siguiente ecuación:

$$v_l = \sqrt{(v_x)^2 + (v_y)^2}. \quad (109)$$

- **Orientación:** Aunque el módulo Piksi<sup>®</sup> Multi integra una IMU de 9 grados de libertad, su uso para calcular la orientación del TR fue descartado tras las pruebas experimentales. Esta decisión responde a la acumulación de deriva y ruido en las mediciones. Por requerimientos de línea de visión satelital, el sensor se instaló sobre un mástil en la parte superior del chasis. Durante la operación, esta estructura amplifica mecánicamente las vibraciones originadas por el motor y el desplazamiento, saturando las lecturas del acelerómetro y el giroscopio. Paralelamente, la falta de un blindaje electromagnético en el encapsulado del sensor permite la entrada de interferencias en el magnetómetro. Al ser procesadas por los filtros de estimación, estas perturbaciones vibratorias y magnéticas se integran matemáticamente, generando un error angular acumulativo en la orientación.

La degradación de las mediciones inerciales debido a la naturaleza vibratoria de la maquinaria agrícola ha sido documentada previamente por [99]. Para resolver este problema, dichos autores descartaron la confiabilidad de la IMU e implementaron una configuración de doble antena RTK-GNSS para establecer un vector direccional. A diferencia de ese enfoque, y con el propósito de no depender de instrumentación suplementaria, en esta investigación se optó por prescindir de los datos inerciales y aplicar una solución cinemática.

La orientación del vehículo se determina numéricamente empleando de forma exclusiva el registro de coordenadas del receptor GNSS. Dado que durante el desplazamiento continuo del TR se dispone de las coordenadas planas actuales ( $x(k)$ ,  $y(k)$ ) y las del periodo de muestreo anterior ( $x(k-1)$ ,  $y(k-1)$ ), el ángulo de orientación  $\theta$  respecto al sistema de referencia absoluto se calcula directamente mediante el diferencial de estas variables espaciales, utilizando la siguiente expresión:

$$\theta(k) = \arctan2(y(k) - y(k-1), x(k) - x(k-1)), \quad (110)$$

donde  $\arctan2$  es la función arcotangente de dos parámetros, que calcula el ángulo considerando el signo del numerador y denominador para determinar el cuadrante correcto del ángulo resultante.

Este método permite calcular la orientación del TR basándose en su desplazamiento real sobre el plano cartesiano, evitando los problemas de deriva y acumulación de errores asociados con los sensores inerciales. Al utilizar únicamente las mediciones de posición proporcionadas por el Piksi<sup>®</sup> Multi, se simplifica el procesamiento y se mejora la precisión en la estimación de la orientación, lo cual es fundamental para el desempeño de los algoritmos de seguimiento de trayectorias implementados. La única desventaja de utilizar este método es que el vehículo debe estar en movimiento para que la orientación pueda ser calculada de forma correcta. Para abordar este problema, se agregaron condiciones a nivel de código que permiten únicamente calcular la orientación si el vehículo tiene una velocidad lineal mayor a 0.1 m/s, de lo contrario mantiene la última orientación calculada.

- **Tiempo:** La información de tiempo es fundamental para sincronizar datos y mantener un TM constante. El Piksi<sup>®</sup> Multi proporciona información de tiempo a través del mensaje MSG\_GPS\_TIME, 258 en decimal y 0x0102 en hexadecimal. Este mensaje representa el tiempo transcurrido desde el inicio de la época GPS el 6 de enero de 1980 a la medianoche UTC. El mensaje contiene la variable *tow* (Time of Week en inglés) que es el tiempo de la semana GPS redondeado al milisegundo más cercano. El valor de *tow* equivale a los milisegundos transcurridos desde el inicio de la semana GPS actual, que comienza en la transición de sábado a domingo y abarca 604800 segundos (60 segundos  $\times$  60 minutos  $\times$  24 horas  $\times$  7 días), reiniciándose al comienzo de cada semana GPS. Para calcular el tiempo transcurrido entre dos mediciones consecutivas, se utiliza la siguiente ecuación:

$$\Delta t = tow(k) - tow(k-1), \quad (111)$$

donde:

- $\Delta t$  es el tiempo transcurrido entre medición actual y la anterior en milisegundos.
  - $tow(k)$  es el tiempo de la semana GPS en la medición actual, en milisegundos.
  - $tow(k-1)$  es el tiempo de la semana GPS en la medición anterior, en milisegundos.
- **Bandera:** Además de las mediciones de posición, velocidad, orientación y tiempo, se considera el estado y la calidad de las soluciones GNSS proporcionadas por el receptor Piksi<sup>®</sup> Multi. Para ello, se utiliza la variable *flags* presente en el mensaje

MSG\_BASELINE\_NED 524 en decimal y 0x020c en hexadecimal. La variable *flags* es un campo de 8 bits que proporciona información detallada sobre el estado de la solución GNSS. La interpretación de *flags* es esencial para determinar la validez y confiabilidad de las mediciones. Cada uno de los bits de *flags* representan diferentes indicadores, como: 0-Inválida, 1-Reservada, 2-GNSS diferencial, 3-RTK flotante y 4-RTK fijo. Para la operación de forma automática del TR, se definió que la variable *flags* debe ser igual a 4, de lo contrario se detiene el movimiento hasta recuperar la calidad de la señal.

Para validar la precisión y estabilidad del receptor RTK-GNSS Piksi<sup>®</sup> Multi incorporado en el TR, se llevó a cabo una prueba experimental en la que se dejó el TR estacionario durante un período de tiempo, registrando las coordenadas *x* e *y* proporcionadas por el receptor. El objetivo de esta prueba fue analizar la variación en las mediciones cuando el vehículo está inmóvil, lo cual es fundamental para comprender el nivel de ruido y precisión que el sistema ofrece en condiciones estáticas. El TR se ubicó en una posición fija en un área abierta con buena visibilidad del cielo asegurando la recepción de señales, obteniendo un total de 216 muestras de coordenadas *x* e *y* como se muestra en la Figura 69. Se calculó la diferencia entre los valores máximos y mínimos de las coordenadas *x* e *y* para estimar la desviación máxima en cada eje, utilizando la Ec. (112).

$$\begin{aligned}\Delta x &= x_{\text{máx}} - x_{\text{mín}} \\ \Delta y &= y_{\text{máx}} - y_{\text{mín}}\end{aligned}\tag{112}$$

Para el eje *x* los valores de  $x_{\text{max}}$  y  $x_{\text{min}}$  son iguales a -2.502 m y -2.515 m respectivamente, dando como resultado un  $\Delta x = 0.013$  m. En el eje *y* las variables  $y_{\text{max}}$  e  $y_{\text{min}}$  son iguales a -2.715 m y -2.72 m, obteniendo un  $\Delta y = 0.005$  m. La incorporación del Piksi<sup>®</sup> Multi al sistema del TR ha sido fundamental para mejorar la precisión y fiabilidad del seguimiento de trayectorias. Las pruebas realizadas demuestran que el sensor proporciona mediciones de posición altamente precisas y estables, incluso en condiciones estáticas. Esto permite al TR operar con mayor exactitud en el seguimiento de trayectorias y ejecución de tareas en entornos agrícolas.

#### A.1.5. Implementación del IT2FLS modelado utilizando IT2-ANFIS OSTR

Con el propósito de validar la estructura IT2-ANFIS OSTR, el IT2FLS resultante se programó en lenguaje C/C++ para ser ejecutado y evaluado en sistemas embebidos. Este código demuestra que las operaciones de un IT2FLS son aptas para ser integradas en el flujo de procesamiento de un microcontrolador convencional. Para lograr lo anterior en el bucle principal, las operaciones matemáticas se estructuraron cuidadosamente en cuatro bloques secuenciales:

1. **Declaración de parámetros y matrices:** Todos los parámetros de las funciones de pertenencia y los consecuentes obtenidos durante la etapa de modelado IT2-ANFIS

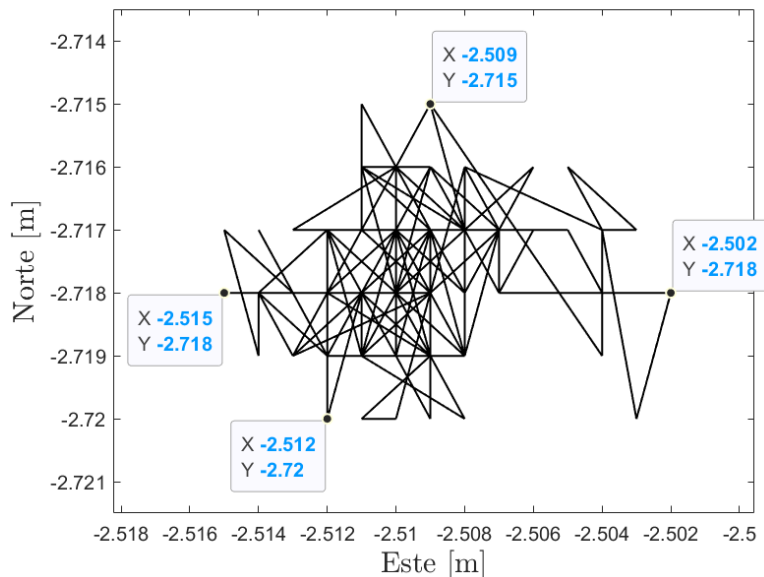


Figura 69: Nube de puntos del RTK-GNSS durante la prueba de inmovilidad del TR

OSTRA son transportados como matrices constantes en memoria (constexpr) permitiendo un cálculo inmediato sin consumo de memoria dinámica.

```
constexpr const float MFpa_i[4] = {8.6449, 8.6754, 23.9982, 23.9984};
constexpr const float MFpa_s[4] = {10.3708, 10.4218, 28.8013, 28.7957};
constexpr const float MFpb[4]   = {1.1764, 0.8790, 1.0303, 1.0851};
constexpr const float MFpc[4]   = {-21.4879, 21.8428, 4.0002, 124.0004};
constexpr const float MFpB[4]   = {0.1, 0.6583, 0.1189, 0.1};
```

```
constexpr const float CF[3][4] = {
    { 0.9846, 0.9861, 1.0076, 0.9850},
    { 0.0090, 0.0018, 0.0046, 0.0127},
    {-1.0204, -0.4040, -0.2355, -0.5921}
};
```

- Fuzificación (grados de membresía y FOU):** En cada interrupción del ciclo, las variables de entrada actualizadas (posición en grados y la señal de control  $U$ ) se sustituyen numéricamente en las GBSMF de IT2 para calcular sus límites superior e inferior. Luego, se emplean las T-normas correspondientes, alojando los resultados en matrices efímeras de disparo de regla ( $W_s$  para la UMF y  $W_i$  para la LMF).

```
float Xes[4] = {(posm-MFpc[0])/MFpa_s[0], (posm-MFpc[1])/MFpa_s[1],
               (U-MFpc[2])/MFpa_s[2], (U-MFpc[3])/MFpa_s[3]};
```

```

float Xei[4] = {(posm-MFpc[0])/MFpa_i[0], (posm-MFpc[1])/MFpa_i[1],
               (U-MFpc[2])/MFpa_i[2], (U-MFpc[3])/MFpa_i[3]};

float pfs[4] = {1/(1+pow(Xes[0]*Xes[0],MFpb[0])),
               1/(1+pow(Xes[1]*Xes[1],MFpb[1])),
               1/(1+pow(Xes[2]*Xes[2],MFpb[2])),
               1/(1+pow(Xes[3]*Xes[3],MFpb[3]))};

float pfi[4] = {MFpB[0]/(1+pow(Xei[0]*Xei[0],MFpb[0])),
               MFpB[1]/(1+pow(Xei[1]*Xei[1],MFpb[1])),
               MFpB[2]/(1+pow(Xei[2]*Xei[2],MFpb[2])),
               MFpB[3]/(1+pow(Xei[3]*Xei[3],MFpb[3]))};

float Ws[4] = {pfs[0]*pfs[2], pfs[0]*pfs[3], pfs[1]*pfs[2], pfs[1]*pfs[3]};
float Wi[4] = {pfi[0]*pfi[2], pfi[0]*pfi[3], pfi[1]*pfi[2], pfi[1]*pfi[3]};

```

3. **Reducción de tipo:** Aprovechando mecanismos afines al algoritmo implementado internamente por el modelo (como un reordenamiento de tipo *bubble sort* sobre los registros  $Ws$  y  $Wi$ ), el embebido procede a buscar interactivamente los puntos de cambio ( $l$  y  $r$ ) requeridos por OSTR. Finalmente, los pesos se vuelven a ordenar a un estado defuzificado de primer orden en un arreglo maestro  $Xoo$ .

```

for (int i=0; i<4; i++) { Wia[i] = Wi[i]; }

bubble_sort(Ws, 4);
bubble_sort(Wi, 4);

for (int i=0; i<4; i++) {
    for(int j=0; j<4;j++) { if(Wia[j]==Wi[i]) order[i] = j; }
    Wn[i] = (Wi[i]+Ws[i])/2;
}

Yi = (Wi[0]*Wn[0] + Wi[1]*Wn[1] + Wi[2]*Wn[2] + Wi[3]*Wn[3]) /
      (Wn[0] + Wn[1] + Wn[2] + Wn[3]);

Ys = (Ws[0]*Wn[0] + Ws[1]*Wn[1] + Ws[2]*Wn[2] + Ws[3]*Wn[3]) /
      (Wn[0] + Wn[1] + Wn[2] + Wn[3]);

for(int i=0; i<3; i++) {
    if(Wi[i]<=Yi && Yi<=Wi[i+1]) l=i;
    if(Ws[i]<=Ys && Ys<=Ws[i+1]) r=i;
}

```

```

for(int i=0; i<4; i++) {
  if (i<=l) { Xl[i] = Ws[i]; denominador_l += Ws[i]; }
  else      { Xl[i] = Wi[i]; denominador_l += Wi[i]; }

  if (i<=r) { Xr[i] = Wi[i]; denominador_r += Wi[i]; }
  else      { Xr[i] = Ws[i]; denominador_r += Ws[i]; }
}

for(int i=0; i<4; i++) {
  Xo[i] = ( (Xl[i]/denominador_l) + (Xr[i]/denominador_r) ) / 2;
}

for(int i=0; i<4; i++) { Xoo[i] = Xo[order[i]]; }

```

4. **Defuzificación y consecuentes tipo T-S:** En la etapa final de cada procesamiento iterativo, la salida del modelo  $f$  se consolida como un polinomio, multiplicando el arreglo de pesos previamente ajustado ( $X_{oo}$ ) por sus valores correspondientes extraídos de los consecuentes constantes CF.

```

for(int i=0; i<4; i++){
  f += Xoo[i]*(CF[0][i]*posm + CF[1][i]*U + CF[2][i]);
}

```

Mostrar el IT2FLS en su representación de código corrobora que el modelo obtenido utilizando la estructura IT2-ANFIS OSTRa es implementable; sus requerimientos de carga computacional no resultan prohibitivos y la ejecución natural de su matemática es totalmente compatible con arquitecturas de procesamiento móvil en escenarios reales.

## B. Contribuciones al estado del arte

Int. J. Fuzzy Syst.  
<https://doi.org/10.1007/s40815-024-01754-8>



### Novel Interval Type-2 ANFIS Modeling Based on One-Step Type Reducer Algorithm

Adrián Alberto-Rodríguez<sup>1</sup> · Virgilio López-Morales<sup>1</sup> · Julio Cesar Ramos-Fernández<sup>2</sup>

Received: 4 October 2023 / Revised: 1 March 2024 / Accepted: 4 April 2024  
© The Author(s) under exclusive licence to Taiwan Fuzzy Systems Association 2024

**Abstract** In this paper, a novel structure of Interval Type-2 Adaptive Network-Fuzzy Inference System (IT2-ANFIS) for modeling dynamic systems is proposed. Optimization algorithms are introduced to adjust the antecedent and consequent parameters of a fuzzy model. In order to avoid the classical iterative process commonly used in type reduction algorithms, a new one-step type reduction algorithm (OSTRA) is proposed, which in conjunction with IT2-ANFIS is tested on nonlinear dynamical system and one real system datasets. Furthermore, to validate the complete structure, experiments with numerical models are performed, in order to show the advantages of the proposed novel IT2-ANFIS structure, by obtaining better results than previously published T2-ANFIS structures. To illustrate a real application of the proposed modeling technique, a model obtained from a tractor steering wheel system was embedded in an electronic I/O board, to obtain a comparison of the on-line fuzzy model against the physical system, with satisfactory results in the approximation error.

**Keywords** Interval type-2 fuzzy logic system · ANFIS · One-step type reducer algorithm · Fuzzy modeling

#### 1 Introduction

Fuzzy logic systems (FLSs) have become a widely spread tool for modeling complex systems involving uncertainty and imprecision [1]. Type-1 FLS (T1FLS) has been widely used in applications for modeling, v.gr. social, economic, and dynamic systems with outstanding results. To include systems with greater complexity and uncertainty, Type-2 FLS (T2FLS) [2] can be used as a sound strategy to improve modeling results with respect to T1FLS systems. T2FLS provide a more accurate and complete representation of uncertainty, as the modeling is performed using fuzzy sets with a higher ability to handle uncertainty than Type-1 fuzzy sets (T1FS) [3]. A general problem of T2FLS is the high computational cost due to an iterative optimization process to calculate the antecedent and consequent parameters of the final model and the involved analytic computation. Furthermore, an Interval Type-2 fuzzy logic system (IT2FLS) takes into account (in iterations and computations) two membership functions per work interval where T1FLS takes only one. In addition, with a T2FLS, a type reduction step is iteratively performed, to obtain an output result from T2 to T1. Type reduction methods are an important part of T2FLS, since they are used to collect all the inherent data uncertainty to obtain the output of the T2FS within a space, known as the Footprint of Uncertainty (FOU) [4]. There are several types of reduction methods [4], where the choice of method depends on the specific application, and some methods are more accurate than others, but may also be slower to converge.

Adrián Alberto-Rodríguez, V. López-Morales and J. C. Ramos-Fernández have contributed equally to this work.

✉ Virgilio López-Morales  
virgilio@uaeh.edu.mx

Adrián Alberto-Rodríguez  
al296001@uaeh.edu.mx

Julio Cesar Ramos-Fernández  
jramos@upp.edu.mx

<sup>1</sup> Universidad Autónoma del Estado de Hidalgo, Carr. Tulancingo-Pachuca Km. 4.5, Pachuca de Soto 42184, HGO, México

<sup>2</sup> Universidad Politécnica de Pachuca, Carr. Pachuca-Cd. Sahagún Km. 20, Rancho Luna, Ex-Hacienda de Sta. Bárbara, Zempoala 43830, HGO, México

Published online: 12 June 2024

Springer

In MATLAB® Type-2 Fuzzy Inference Systems tool, four type of reduction algorithms are available: Karnik-Mendel algorithm (KMA) [5], Enhanced KMA (EKMA) [6], iterative algorithm with stop condition (IASC) [7], and the enhanced iterative algorithm with stop condition (EIASC) [8]. The KMA is the most common method to calculate the centroid of a T2FS by an iterative finding of optimal switching points. A switching point resides in the domain where the membership function (MF) of the T2FS changes from one value to another. Then, this algorithm iterates up to find the centroid of the T2FS. In our proposed architecture, a new type reduction algorithm is introduced, to obtain the centroid in a single step and even it facilitates the computation of the consequent parameters through optimization algorithms.

The T2FLS's optimization is still an argumentative issue, since many of the decisions, such as the choice of membership functions, the number of rules, operators, and defuzzification algorithms, are still made heuristically. However, improvements can be obtained through automatic learning, evolutionary algorithms, and bio-inspired optimization techniques [9]. ANFIS [10] is one of the most popular methods for FLS modeling which is based on a combination of neural networks and fuzzy logic systems to automatically optimize the antecedent and consequent parameters, based on the inputs–output of the system to be modeled. In [11], up to 52 different algorithms for the parameters optimization with T1-ANFIS are mentioned, in addition to many of them with real applications. Among the most popular and accurate optimization algorithms for T1FLS parameters are the following: BackPropagation algorithm (BPA) [12], particle swarm optimization [13], evolutive algorithms [14], and Biogeography-based optimization algorithm [15]. In this paper, the BPA is implemented for the optimization of the premise parameters of IT2, as well as the parameter that determines the amplitude of the FOU. Kalman filter (KF) is applied for the optimization of the consequent parameters of T1 Takagi–Sugeno (T–S) [16].

In [17], an IT2-ANFIS is proposed and called IT2 NS-1 ANFIS, referring to the fact that the antecedent parameters are from IT2 and the consequent parameters are from T1 non-singleton T–S. In [18], an IT2-ANFIS with non-singleton T–S fuzzification is introduced, and the optimization algorithms used are, for the antecedent parameters, the gradient descent or Levenberg–Marquardt algorithm, and for the consequent, recursive least squares. In [19], the equations of two IT2-ANFIS structures called ANFIS2 are described. First one uses IT2FS Gaussians in the antecedent part and T1FS Gaussians in the consequent part. Second one is based on linear T–S consequents. Both ANFIS2 structures use BPA based on the adaptive learning rate with guaranteed convergence for the parameter

optimization. Also, in [20], the ANFIS2 structure is used to model a robotic arm with flexible joints and tested in a control scheme, obtaining a better response with the ANFIS2 model against a T1-ANFIS. In addition, in [21–23], traditional T1-ANFIS structure is modified and tested to train IT2FLS. However, they state that results are not significantly better compared to T1-ANFIS results. In [24], an IT2-ANFIS using a type reduction algorithm (Modified-KMA) is used where only the consequent parameters are optimized to show the effects of the type reduction algorithm, while the parameters of the antecedent remain unchanged during training in each of the ANFIS structures. They claim that the results of IT2-ANFIS are equal to or better than those of T1-ANFIS.

In this paper, an IT2-ANFIS structure is proposed, which uses BPA for the optimization of the antecedent parameters of generalized bell-shaped membership function (GBSMF) of IT2, including the parameter  $h$  which is an uncertainty measure and determines the amplitude of the FOU. In addition, KF algorithm is used to optimize the consequent parameters T1 non-singleton T–S. Also, a new OSTRA is proposed to avoid numerous iterations during the mapping from T2 to T1. To show the effectiveness of the structure and the OSTRA, one nonlinear numerical system and one real dynamic system are modeled. Comparisons of the RMSE obtained with our structure against some already mentioned IT2-ANFIS structures and against the traditional T1-ANFIS are made, in order to illustrate the application and effectiveness of the proposed structure. Our algorithm generates IT2FLS with optimal antecedent and consequent parameters, where in the consequent parameters the traditional T–S structure is kept. IT2FLS obtained can be then used for controller design, fault diagnosis strategies, and prediction, among other applications.

The paper organization is as follows: In Sect. 2, a generalized bell-shaped MF of IT2 is shown. In Sect. 3, the type reduction algorithm is described. In Sect. 4, the IT2-ANFIS structure is introduced through eight layers and using the OSTRA. In Sect. 5, the results obtained from modeling with different datasets are illustrated and analyzed, comparing the RMSE of each structure. Finally, in Sect. 6, conclusions and future work are given.

## 2 Generalized Bell-Shaped Membership Function of IT2

An IT2FS is defined as  $\tilde{A}$  and composed of two Type-1 MFs, a lower (LMF) and an upper (UMF) [25]. The interval between both MFs  $[\underline{A}, \overline{A}]$  is defined as the FOU as shown in Fig. 1. This FS in an IT2FLS is characterized by

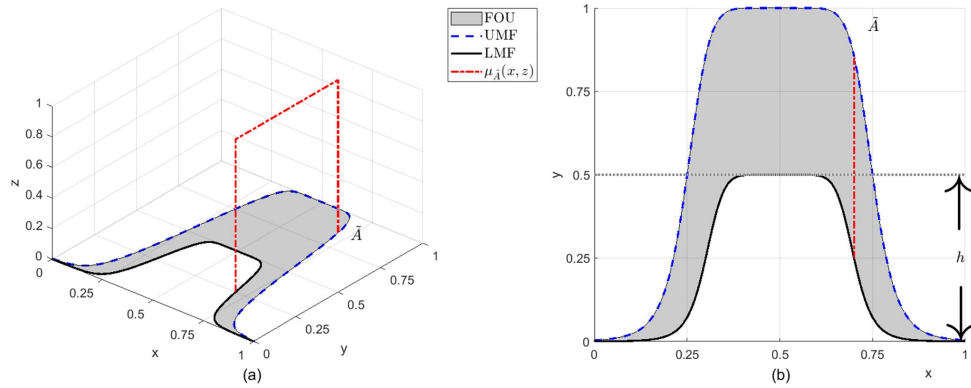


Fig. 1 3D (a) and 2D (b) view of an IT2 GBSMF with  $h$  parameter equals 0.5

having a MF  $[\bar{A}, \mu_{\bar{A}}(x, z)]$  in the  $z$  dimension, where all the degrees of membership are equal to 1, meaning that the information contained in the  $z$  dimension is equally weighted. The degree of membership to the LMF is defined as  $\mu_{\bar{A}}(x)$  and for the UMF as  $\mu_{\bar{A}}^-(x)$ . A known MF, as the GBSMF, can be used. Next, its definition is stated.

$$\mu_{\bar{A}_j}(x_j) = \left[ \mu_{\bar{A}_{j,k}}(x_j), \mu_{\bar{A}_{j,k}}^-(x_j) \right] = \left[ f(x_j; \underline{a}_{j,k}, b_{j,k}, c_{j,k}, h_{j,k}), f(x_j; \bar{a}_{j,k}, b_{j,k}, c_{j,k}) \right], \quad (1)$$

$$\left[ f(x_j; \underline{a}_{j,k}, b_{j,k}, c_{j,k}, h_{j,k}), f(x_j; \bar{a}_{j,k}, b_{j,k}, c_{j,k}) \right] = \left[ \frac{h_{j,k}}{1 + \left| \frac{x_j - c_{j,k}}{\underline{a}_{j,k}} \right|^{2b_{j,k}}}, \frac{1}{1 + \left| \frac{x_j - c_{j,k}}{\bar{a}_{j,k}} \right|^{2b_{j,k}}} \right]. \quad (2)$$

To calculate the degree of membership of a GBSMF of an IT2, the Eq. (2) is used, where  $\underline{a}_{j,k}$  and  $\bar{a}_{j,k}$  define the width of the LMF and UMF, respectively. A larger value creates a broader membership function. The parameter  $b_{j,k}$  defines the shape of the curve on either side of the central plateau, where a larger value creates a steeper transition, with  $c_{j,k}$  being the center of the GBSMF. Finally, the parameter  $h_{j,k} \in [0, 1]$  is the measure of the uncertainties; the closer to 0, the greater the FOU, and viceversa. This parameter only defines the boundary of the LMF and can be used for the different MF types. An important feature of our proposed IT2-ANFIS structure is the ability to optimize all the antecedent parameters and the parameter  $h_{j,k}$  described in Eq. (2) using BPA, by defining the shape of the FOU for each IT2FS during training.

### 3 Type Reducer

While the output of a T1FLS is crisp, for an IT2FLS and due to the uncertainties in the measurements and the shape of the MFs, it cannot be calculated directly, so a type reduction algorithm is implemented to obtain a crisp output. For instance in [24], an  $X$  matrix is used to estimate the consequent parameters. The proposed OSTRA is described below and compared with M-KMA.

#### 3.1 One-Step Type Reducer Algorithm [OSTRA]

- Sort the weights  $w_i$  and  $\bar{w}_i$  of Eq. (3) in increasing order, obtaining a new index  $n$ .

$$\begin{aligned} w_i &= \mu_{\bar{A}_{j,k}}(x_j) \mu_{\bar{A}_{j+1,k}}(x_{j+1}) \cdots \mu_{\bar{A}_{j,k}}(x_j), \\ \bar{w}_i &= \mu_{\bar{A}_{j,k}}^-(x_j) \mu_{\bar{A}_{j+1,k}}^-(x_{j+1}) \cdots \mu_{\bar{A}_{j,k}}^-(x_j). \end{aligned} \quad (3)$$

- Get  $w_n$  with the lower and upper weights ordered, as follows  $w_n = \frac{w_n + \bar{w}_n}{2}$ , calculate the reference points  $\underline{y}$  and  $\bar{y}$  as shown in Eq. (4).

$$\underline{y} = \frac{\sum_{n=1}^N w_n w_n}{\sum_{n=1}^N w_n}, \quad \bar{y} = \frac{\sum_{n=1}^N \bar{w}_n w_n}{\sum_{n=1}^N w_n}. \quad (4)$$

- Find the point of change “ $l$ ” for  $w_n < \underline{y} \leq w_{n+1}$  and “ $r$ ” for  $\bar{w}_n < \bar{y} \leq \bar{w}_{n+1}$
- Compute  $X$  by setting  $X = \frac{\underline{y} + \bar{y}}{2}$ , where:

$$\underline{X} = \frac{[\bar{w}_1, \dots, \bar{w}_l, \underline{w}_{l+1}, \dots, \underline{w}_N]}{\sum_{n=1}^l \bar{w}_n + \sum_{n=l+1}^N \underline{w}_n}, \quad (5)$$

$$\bar{X} = \frac{[\underline{w}_1, \dots, \underline{w}_r, \bar{w}_{r+1}, \dots, \bar{w}_N]}{\sum_{n=1}^r \underline{w}_n + \sum_{n=r+1}^N \bar{w}_n}.$$

5. The calculated  $X$  row vector is a sorted vector. However,  $X$  vector must be unsorted:

$$\text{Unsort}(X) = [w_i \quad w_{i+1} \quad \dots \quad w_N]. \quad (6)$$

6. Compute  $y$  with the unsorted  $X$  vector as shown in the following:

$$y = X \begin{bmatrix} f_i \\ f_{i+1} \\ \vdots \\ f_N \end{bmatrix}. \quad (7)$$

where  $f_i$  are the  $i$ -th rule output,  $i = 1, 2, \dots, N$  are the number of the fuzzy rules,  $C_{i,j}$ ,  $C_{i,j+1}, \dots$ , and  $C_{i,j}$  are the consequent parameter and  $C_{i,j+1}$  is the independent parameter, as shown in Eq. (8).

$$f_i = C_{i,j}x_j + C_{i,j+1}x_{j+1} + \dots + C_{i,j}x_j + C_{i,j+1}, \quad (8)$$

To illustrate the advantages of our proposed type reducer algorithm, a comparison of our OSTRA is made against the EIASC [8], KM [5], and EKM [6]. The algorithms are used to compute the output of three IT2FS whose FOUs are depicted in Fig. 2. The idea of the computational comparison is to obtain the average computing time that each algorithm spends in finding the output of a given FOU. The experimental procedure to carry out this comparison is described as follows:

1. Set the algorithm on the computing platform.
2. Set a FOU with a discretization of  $N$  points.

3. Measure the computing time demanded by  $M=100$  runs of the algorithm.
4. Record the value obtained for output in order to establish a numerical comparison.

Experiments are carried out over an Intel® Core i9-12900 H with 16GB of RAM and Python 3.11.5 computing environment. Tables 1, 2, 3 present the results relative to the computing time of the algorithm and outputs. The algorithms are equivalent in terms of arithmetic precision since they provide the same results for the outputs functions. The difference between the calculation times of each algorithm is similar; however, OSTRA has the shortest calculation time. The difference in computing times is related to the computational complexity of the algorithms more than to their mathematical properties. OSTRA has been proposed as alternative solution to compute the generalized centroid of an interval type-2 fuzzy set is obtained with only one run.

The differences between the OSTRA and the M-KMA are:

1. In OSTRA, ordering is done concerning the  $\underline{w}_i$  and  $\bar{w}_i$  weights of the  $i$ -th rule and not with the rule outputs  $f_i$ .
2. The computing of the IT2FLS centroid in OSTRA is obtained with only one run during each training epoch of the IT2-ANFIS structure. The M-KMA requires an unknown number of runs to find  $y_l$  and  $y_r$  for each training epoch, increasing the execution time of the algorithm.
3. The  $X$  matrix obtained through OSTRA does not contain the system inputs and is unsorted both in the ANFIS training and in the model validation. On the other hand, in M-KMA, the unsort step is only carried out when training the model with IT2-ANFIS.
4. OSTRA can be used in the IT2-ANFIS structure and with the fuzzy model, without the need to modify it. The M-KMA is only used during IT2-ANFIS training

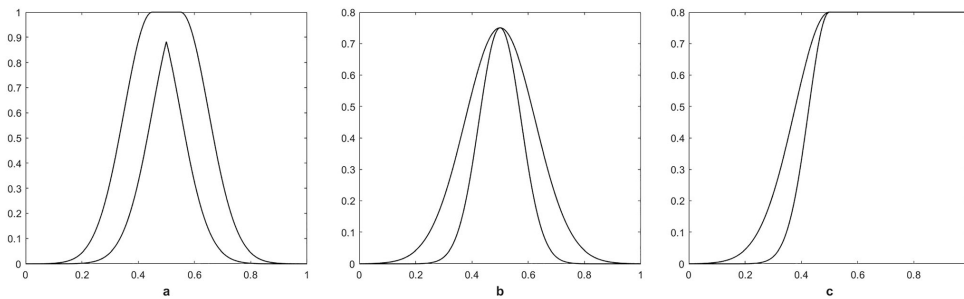


Fig. 2 FOUs used for the computational comparison between OSTRA, EIASC, KM and EKM algorithms. a Gaussian with uncertain mean, b Gaussian with uncertain standard deviation, and c Gaussian with uncertain mean (right-sided)

**Table 1** Results primary Gaussian IT2FS with uncertain mean

N	OSTRA		EIASC		KMA		EKMA	
	Output	Time	Output	Time	Output	Time	Output	Time
10	0.9021	0.001	0.9021	0.002	0.9021	0.005	0.9021	0.003
100	0.8903	0.006	0.8903	0.008	0.8903	0.019	0.8903	0.013
1000	0.8408	0.04	0.8408	0.05	0.8408	0.14	0.8408	0.16
10,000	0.7209	0.4	0.7209	0.5	0.7209	1.3	0.7209	1.4

**Table 2** Results primary Gaussian IT2FS with uncertain standard deviation

N	OSTRA		EIASC		KMA		EKMA	
	Output	Time	Output	Time	Output	Time	Output	Time
10	0.4877	0.001	0.4877	0.003	0.4877	0.005	0.4877	0.005
100	0.5701	0.005	0.5701	0.008	0.5701	0.02	0.5701	0.01
1000	1.0151	0.04	1.0151	0.06	1.0151	0.15	1.0151	0.17
10,000	1.0855	0.4	1.0855	0.5	1.0855	1.4	1.0855	1.5

**Table 3** Results right Gaussian IT2FS with uncertain standard deviation

N	OSTRA		EIASC		KMA		EKMA	
	Output	Time	Output	Time	Output	Time	Output	Time
10	0.8847	0.002	0.8847	0.003	0.8847	0.006	0.8847	0.004
100	0.8624	0.006	0.8624	0.007	0.8624	0.019	0.8624	0.018
1000	0.9069	0.03	0.9069	0.06	0.9069	0.12	0.9069	0.18
10000	1.2895	0.2	1.2895	0.6	1.2895	1.2	1.2895	1.6

and, for the fuzzy model, the KMA must be used in the type reduction step.

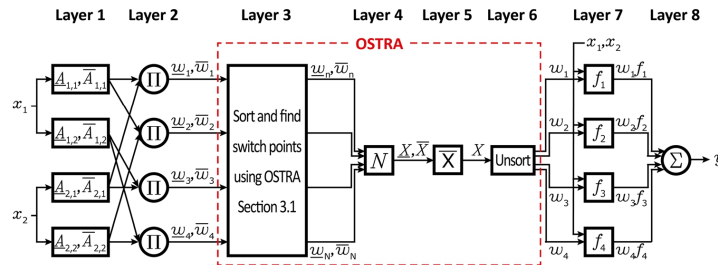
The proposed IT2-ANFIS structure is described below, which is used by OSTRA in type reduction step.

### 4 IT2-ANFIS

For the sake of clarity, the proposed IT2-ANFIS structure in Fig. 3 is shown for  $J = 2$  inputs, one output, and  $K = 2$  MFs for each input, with a number of rules or nodes

$N = K^J = 4$ . The IT2-ANFIS structure has 8 layers, which are described in the following:

*Layer 1:* Similar to T1-ANFIS, each node in this layer is a fuzzifier, so the output  $O_{(j-1)K+k}^1$  can be represented as the interval of degrees to membership of two T1FS  $\underline{A}_{j,k}$  and  $\bar{A}_{j,k}$ , which together represent an IT2FS. Thus,  $a_{j,k}$ ,  $\bar{a}_{j,k}$ ,  $b_{j,k}$ ,  $c_{j,k}$ , and  $h_{j,k}$  are the set of antecedent parameters of a GBSMF of IT2, where  $(a_{j,k}, b_{j,k}, c_{j,k}, \text{ and } h_{j,k})$  belong to the LMF and  $(\bar{a}_{j,k}, b_{j,k}, \text{ and } c_{j,k})$  to the UMF. The degrees of membership of each FS are calculated using antecedent parameters, as indicated in Eq. (2). The correct way to



**Fig. 3** IT2-ANFIS structure proposed, for a particular case of 2 inputs and 2 MF for each input

interpret the indexes  $j, k$  is as a for loop from  $j = 1$  to  $J$  and nest another for loop from  $k = 1$  to  $K$ .

$$O_{(j-1)K+k}^1 = [\mu_{A_{j,k}}(x_j), \mu_{\bar{A}_{j,k}}(x_j)]. \tag{9}$$

**Layer 2:** The interval of the lower and upper weights  $[\underline{w}_j, \bar{w}_j]$ , respectively, is calculated. The values of  $w_j$  and  $\bar{w}_j$  are obtained by multiplying the degrees of membership obtained in Layer 1 as follows:

$$O_i^2 = [\underline{w}_i, \bar{w}_i] = [\mu_{A_{j,k}}(x_j) \mu_{A_{j+1,k}}(x_{j+1}) \cdots \mu_{A_{j,k}}(x_j) \mu_{\bar{A}_{j+1,k}}(x_{j+1}) \cdots \mu_{\bar{A}_{j,k}}(x_j)]. \tag{10}$$

The interpretation for the indexes  $j, k$  of this layer is done by nesting for the loops from  $k = 1$  to  $K$ , equal to the number of inputs  $J$ . The index  $j$  is used as shown in the Eq. (10).

**Layer 3:** In this layer, weights obtained in the previous layer are ordered in ascending order to calculate the change points “ $l$ ” and “ $r$ ” using the OSTRAs described in Sect. 3.1, up to step number 3. The ordered weights and the change points calculated in the OSTRAs are obtained at the output of this layer.

$$O_n^3 = [\underline{w}_n, \bar{w}_n]. \tag{11}$$

**Layer 4:** To calculate the output of this layer (normalization layer), the ordered weights and the change points calculated in Layer 3 are used. The normalized value is the ratio between the weight of the  $i$ th ordered rule and the total of all ordered weights, generating two column vectors  $\underline{X}$  and  $\bar{X}$  as shown in Eqs. (12) and (13). The interval of these two vectors forms the output  $O^4$  as shown in Eq. (14).

$$\underline{X} = \frac{[\underline{w}_n, \dots, \bar{w}_l, \underline{w}_{l+1}, \dots, \underline{w}_N]}{\sum_{n=1}^l \bar{w}_n + \sum_{n=l+1}^N \underline{w}_n}, \tag{12}$$

$$\bar{X} = \frac{[\underline{w}_n, \dots, \underline{w}_r, \bar{w}_{r+1}, \dots, \bar{w}_N]}{\sum_{n=1}^r \underline{w}_n + \sum_{n=r+1}^N \bar{w}_n}, \tag{13}$$

$$O^4 = [\underline{X}, \bar{X}]. \tag{14}$$

**Layer 5:** Add the vectors  $\underline{X}$  and  $\bar{X}$  and get the interval average as represented in Eq. (15) to generate the normalized weights.

$$O^5 = X = \frac{\underline{X} + \bar{X}}{2}. \tag{15}$$

**Layer 6:** Reorder the weights in the  $X$  matrix according to the original order in  $w_i$  before reaching Layer 3.

$$O^6 = \text{Unsort}(X) = [w_i, w_{i+1}, \dots, w_N]. \tag{16}$$

**Layer 7:** This is a defuzzification layer. Through a first-order polynomial as observed in Eq. (17), the weighted values of the rules are calculated, using the consequent parameters, weights, and inputs of Eq. (17).

$$O_i^7 = w_i f_i = w_i (C_{i,j} x_j + C_{i,j+1} x_{j+1} + \dots + C_{i,J} x_J + C_{i,J+1}). \tag{17}$$

**Layer 8:** Sum of the outputs of each rule, as represented in Eq. (18).

$$O^8 = \sum_{i=1}^N O_i^7 = \sum_{i=1}^N w_i f_i. \tag{18}$$

In the following, our proposed scheme for fuzzy identification and modeling is applied.

### 5 Simulation Results

In this section, a comparison of our architecture is made against the T1-ANFIS developed by Jang [10] since, up to our knowledge, there is no an available platform like ANFIS-MATLAB [26] for IT2-ANFIS. A main feature of the IT2-ANFIS OSTRAs is the additional optimization of the  $h_{j,k}$  parameters (cf. Eq. 2) to define an optimal shape of the FOU for each IT2FS during the training phase. To carry out a robust comparison, two GBMSMFs are used for each input variable and the number of training epochs is set to 100 for each of the examples described below.

#### 5.1 Theoretical Data

**Example 1** Aiming a comparison with the modeled system provided in [19], we take their example which is the following second-order nonlinear time-varying plant:

$$y_r(k) = \frac{y_r(k-1)y_r(k-2)y_r(k-3)u(k-1)(y_r(k-3)-b(k))+c(k)u(k)}{a(k)+y_r(k-2)^2+y_r(k-3)^2}, \tag{19}$$

where  $y_r(k-1)$ ,  $y_r(k-2)$ , and  $y_r(k-3)$  are one-, two-, and three-order delayed outputs of the plant, and  $u(k)$  and  $u(k-1)$  are the current and the one-step-delayed input of the plant, respectively. The time-varying parameters  $a(k)$ ,  $b(k)$ , and  $c(k)$  are given by:

$$\begin{aligned}
 a(k) &= 1.2 - 0.2\cos\left(\frac{2\pi k}{T}\right); \\
 b(k) &= 1 - 0.4\sin\left(\frac{2\pi k}{T}\right); \\
 c(k) &= 1 + 0.4\sin\left(\frac{2\pi k}{T}\right);
 \end{aligned}
 \tag{20}$$

where  $T = 1000$ . The excitation signal  $u(k)$  is chosen as follows:

$$u(k) = \begin{cases} \sin\left(\frac{\pi k}{25}\right), & k < 250, \\ 1, & 250 \leq k < 500, \\ -1, & 500 \leq k < 750, \\ 0.3\sin\left(\frac{\pi k}{25}\right) + 0.1\sin\left(\frac{\pi k}{32}\right) + 0.6\sin\left(\frac{\pi k}{10}\right), & 750 \leq k < 1000. \end{cases}
 \tag{21}$$

In this example  $u(k)$ ,  $k = 1, \dots, 1000$  is fed to the non-linear dynamical system Eq. (19) and 1000 samples of  $y_r(k)$  are obtained. Figure 4 shows the identification results of the T1-ANFIS and IT2-ANFIS OSTRA structures; since the error is very similar, no difference can be seen. In Fig. 5, on the other hand, the error between the reference signal and the model of each structure is shown. It can be seen that IT2-ANFIS OSTRA has a smaller error than T1-ANFIS.

The results obtained with T1-ANFIS and IT2-ANFIS OSTRA were compared with three other numerically simulated platforms taken from [19], as shown in Table 4, where it can be seen that IT2-ANFIS OSTRA obtains a lower error with respect to the whole set of platforms.

### 5.2 Real Data Using Steering Wheel System

**Example 2** Autonomous navigation of mobile robots in crop fields is challenging due to the variable nature of the environment. Crop fields frequently have obstacles, such as

trees, posts, and fences, and the ground can be uneven and unstable [29]. Then, the obtainment of a robust model, despite the environment disturbances, is paramount. In this paper, a model of a steering wheel system of a robot tractor (RT) is proposed to test our proposed modeling scheme. Steering wheel modeling and control has been addressed by different authors [30–32] using T1FLS techniques. For instance, in [33, 34], the dynamic subsystems of the steering wheel and speed are modeled using fuzzy T2 neural networks, to achieve autonomous navigation of RTs.

For this example, an automatized RT prototype is used to obtain the I/O data from its steering wheel system. The RT has a servo mechanism for steering angle movement of the steering wheel. Specifically, the data are obtained by means of a LPD3806 600BM-G5-24C encoder sensor of 600 pulses per revolution, coupled to a rack-and-pinion mechanism where the steering angle is measured (Fig. 6).

A 24 V SHINANO KENSHI brand direct current motor is used for the movement of the steering wheel, with a torque of 1.5 Nm and a maximum of 300 revolutions per minute, as well as a 2x30 Sabertooth brand H-bridge. The input signal  $u(k)$  has values between 1 and 127, where 1 is a turn at maximum speed in the negative direction and 127 is in the positive direction, as shown in Fig. 7. The signal  $y_r(k)$  is a unit delay of the output  $y_r(k + 1)$  of the plant. The data acquisition system is embedded in a electronic I/O board with a sample time of 20 ms.

A dataset of 811 I/O samples was obtained to carry out the identification using T1-ANFIS and IT2-ANFIS OSTRA methods. Figure 8 shows the approximation results obtained for T1-ANFIS and IT2-ANFIS OSTRA. Similarly, Fig. 9 shows the error of each model with respect to the actual output of the plant. The results of IT2-ANFIS OSTRA are better compared to T1-ANFIS by thousandths of degrees of the position of the steering wheel, as observed in Table 5.

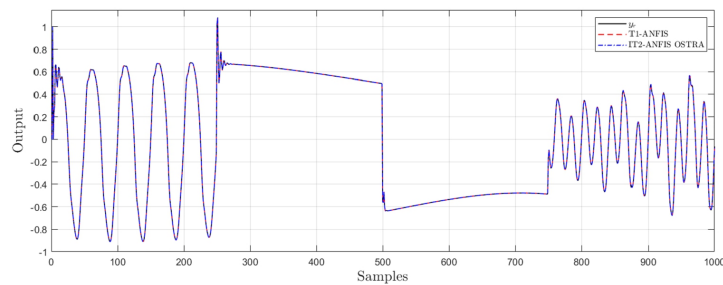


Fig. 4 T1-ANFIS vs IT2-ANFIS OSTRA approximation of the output signal (Example 1)

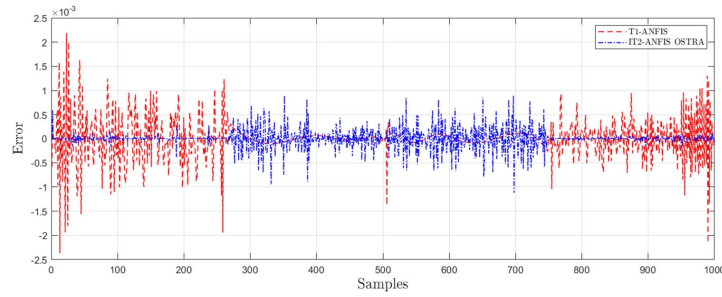


Fig. 5 T1-ANFIS vs IT2-ANFIS OSTR model error,  $(y_r - y_m)$  (Example 1)

Table 4 RMSE results of test T1-ANFIS vs IT2-ANFIS (Example 1)

Example	T1-ANFIS	IT2-ANFIS OSTR	Method [27]	Method [28]	ANFIS2 Fuzzy [19]
1	0.000396	0.000192	0.0253	0.0042	0.0009

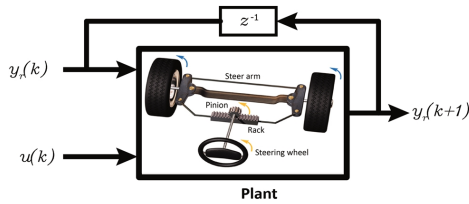


Fig. 6 The structure of steering wheel system of the RT, with 2 inputs and 1 output

By using the model obtained by IT2-ANFIS OSTR programmed in an electronic I/O board, with the optimal antecedent and consequent parameters shown in Table 6, we carried out two additional tests in real time to verify the effectiveness of the model. For both tests, the same input signal  $u(k)$  was used with a different dataset for a cross-validation, as shown in Fig. 10. The control signal was feeded to the plant and the model, while the output

obtained from the plant is fed back to the model. In addition, two different terrains were used so that the model was exposed to unknown disturbances. Test No. 1 was performed with the RT stopped and on a flat surface, as shown in Fig. 11. For test No. 2, the tractor was on a dirt surface and in motion, to generate a difference between the data of the flat surface, as shown in Fig. 12. The results of the test on flat and dirt terrain are shown in Figs. 11 and 12, respectively.

From both tests, note that the IT2-ANFIS OSTR model is reliable and robust even with unknown input data and when there are disturbances that were not found in the training dataset.

A comparison of the performance of the OSTR with three other algorithms: EIASC, KMA, and EKMA is presented in Table 7. Using the model parameters shown in Table 6, we seek to validate IT2FLS using the data from Test No. 1 (Fig. 11) and No. 2 (Fig. 12), but with each of

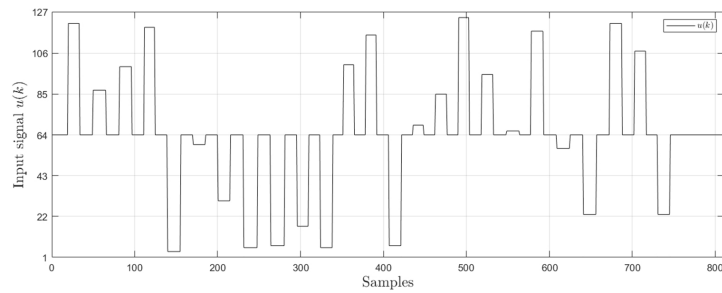


Fig. 7 Input signal  $u(k)$  for training

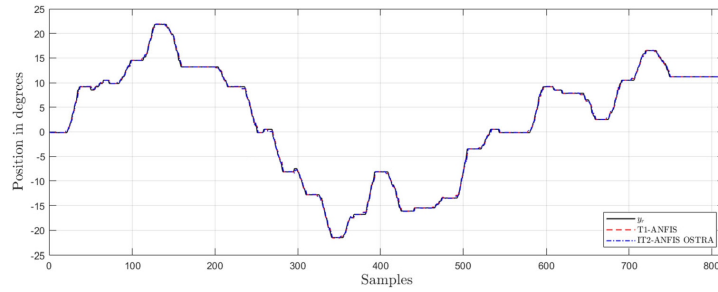


Fig. 8 T1-ANFIS vs IT2-ANFIS OSTR approximation of the output signal (Example 2)

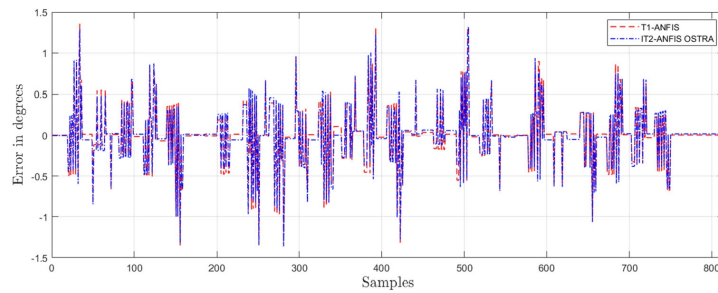


Fig. 9 T1-ANFIS vs IT2-ANFIS OSTR model error,  $(y_r - y_m)$  (Example 2)

Table 5 RMSE results of test T1-ANFIS vs IT2-ANFIS (Example 2)

Example	T1-ANFIS	IT2-ANFIS OSTR
2	0.319667	0.31769

the aforementioned algorithms in the type reduction step. The objective of the test is to evaluate the effectiveness of OSTR in reducing the RMSE error and computation time in IT2FLS. It can be observed that the OSTR algorithm has the lowest RMSE in both tests. In test No. 1, the shortest time is from OSTR and in test No. 2, it is from

Table 6 Antecedent and consequent parameters of the model obtained with IT2-ANFIS OSTR for Example 2

FS	MF	Antecedent parameters					Consequent parameters			
		$\underline{a}$	$\bar{a}$	$b$	$c$	$h$	$\tilde{C}_{i,j}$	$j = 1, \dots, J + 1$ ; Inputs		
$\tilde{A}_{1,1}$	$\underline{A}_{1,1}$	8.6449	–	1.1764	–21.4879	0.1	$i = 1, \dots, N$ ; Rules	0.9846	0.0090	–1.0204
	$\bar{A}_{1,1}$	–	10.3708	–	–	1				
$\tilde{A}_{1,2}$	$\underline{A}_{1,2}$	8.6754	–	0.8790	21.8428	0.6583		0.9861	0.0018	–0.4040
	$\bar{A}_{1,2}$	–	10.4218	–	–	1				
$\tilde{A}_{2,1}$	$\underline{A}_{2,1}$	23.9982	–	1.0303	4.0002	0.1189		1.0076	0.0046	–0.2355
	$\bar{A}_{2,1}$	–	28.8013	–	–	1				
$\tilde{A}_{2,2}$	$\underline{A}_{2,2}$	23.9984	–	1.0851	124.0004	0.1		0.9850	0.0127	–0.5921
	$\bar{A}_{2,2}$	–	28.7957	–	–	1				

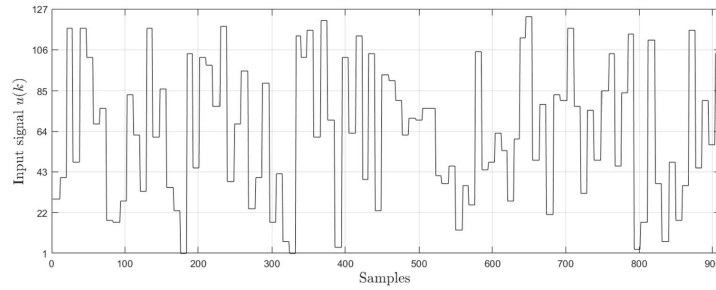


Fig. 10 Input signal  $u(k)$  for test

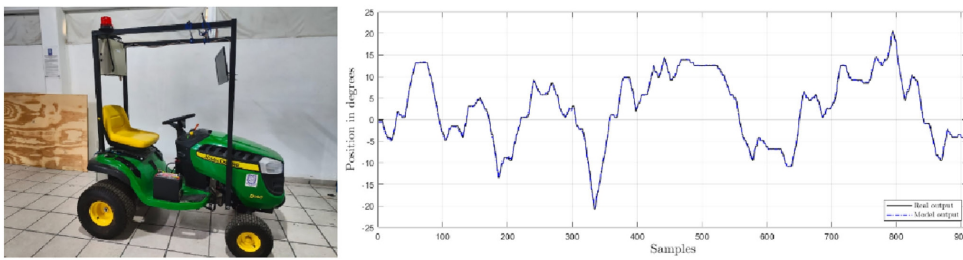


Fig. 11 Test No.1 the RT on a flat terrain, results of the model against the real system with RMSE = 0.3084

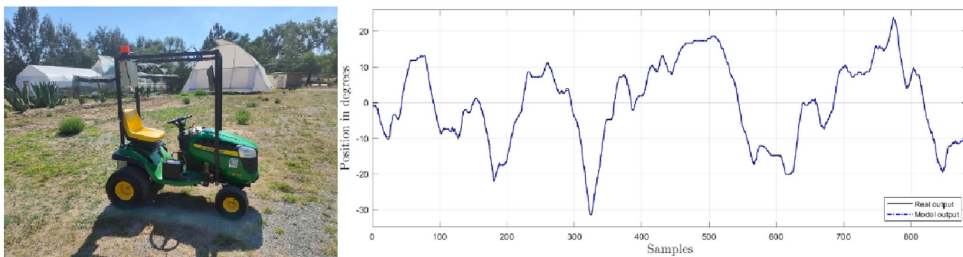


Fig. 12 Test No.2 the RT on a dirt road, results of the model against the real system with RMSE = 0.3088

Table 7 Comparison of RMSE and average computing time for type reduction algorithms of Example 2

Test	OSTRA		EIASC		KMA		EKMA	
	RMSE	Time	RMSE	Time	RMSE	Time	RMSE	Time
No.1	0.3084	0.05	0.4694	0.06	0.4694	0.1	0.4743	0.06
No.2	0.3088	0.05	0.6095	0.04	0.6095	0.09	0.6170	0.06

the EIASC algorithm. The experiments were carried out on an Intel® Core i9-12900H processor with 16GB of RAM and in the Python 3.11.5 programming language.

*Noise reduction test:* In order to test the inherent noise reduction property of IT2FLS, using a  $u(k)$  signal with

white Gaussian noise as shown in Fig. 13, a model is obtained using IT2-ANFIS OSTRA. The input signal  $y(k)$  is the same as that used to obtain the model described in Table 6. The RMSE of identification obtained is 0.3193, very similar to that obtained using an input signal  $u(k)$

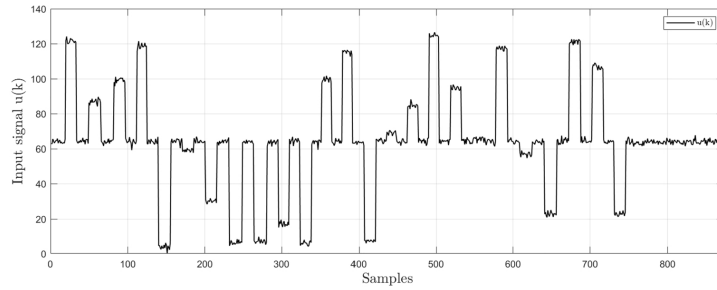


Fig. 13 Input signal  $u(k)$  with white Gaussian noise for test

without noise as shown in Table 9. In Fig. 14, we can see how the addition of noise to the input  $u(k)$  does not considerably affect the output signal  $y(k + 1)$ , and the difference in RMSE is insignificant. For this reason, and as discussed above with Test No.1 and No.2, the model obtained with IT2-ANFIS OSTR A is able to respond to disturbances not included in the training data and is also able to reduce the noise present in the input data.

## 6 Conclusions and Future Works

Obtained experimental results show that IT2-ANFIS OSTR A can achieve better performance than the traditional T1-ANFIS method and other IT2-ANFIS structures from the literature, in the sense of RMSE or real implementation case. The proposed IT2-ANFIS OSTR A is a promising approach for modeling dynamical systems represented with I/O datasets. IT2-ANFIS OSTR A optimizes the antecedent parameters of each MF of the IT2FLS, including the parameter  $h$  that represents the measure of uncertainty and determines the amplitude of the FOU. In addition, the obtained model with IT2-ANFIS OSTR A is capable of responding to different input data and disturbances not

present in the training data, showing that it is an useful tool for modeling uncertainty inherent in complex systems.

The proposed IT2-ANFIS OSTR A eliminates the iterative process in the type reduction step, to calculate the centroid of an T2FS, resulting in less training time and lower computational cost. Additionally, the model obtained with the IT2-ANFIS OSTR A can be programmed in a low-cost microcontroller, as demonstrated throughout the application in the RT steering wheel direction.

As mentioned, the autonomous navigation of an agricultural RT is a challenge due to the various conditions of the terrain. By modeling and controlling the linear velocity and steering wheel direction, an autonomous navigation in different terrains can then be developed. In future works, the design of controllers, based on the fuzzy model, obtained through the IT2-ANFIS OSTR A will be studied. Specifically, we are interested in modeling the RT previously mentioned, with multiple inputs and outputs variables to provide an autonomous navigation for precision agriculture.

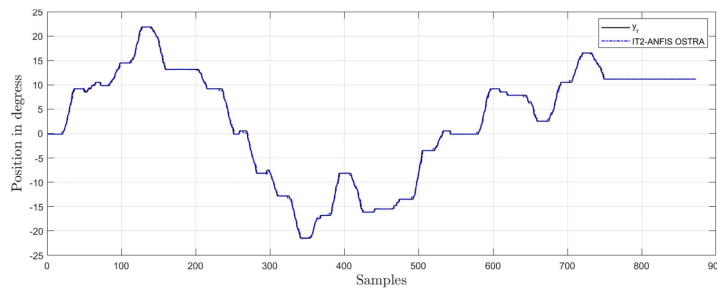


Fig. 14 IT2-ANFIS OSTR A approximation of the output signal (Example 2) with white Gaussian noise in  $u(k)$

**Appendix I**

The general structure of an IT2 TSK model is given as follows:

If  $x_j$  is  $\tilde{A}_{j,k}$  and  $x_{j+1}$  is  $\tilde{A}_{j+1,k}$  and ... and  $x_J$  is  $\tilde{A}_{J,k}$ .  
 Then  $f_i = C_{i,j}x_j + C_{i,j+1}x_{j+1} + \dots + C_{i,J}x_J + C_{i,J+1}$ , (22)

where  $i = 1, 2, \dots, N$  are the number of the fuzzy rules,  $j = 1, 2, \dots, J$  are the number of Inputs,  $k = 1, 2, \dots, K$  are the number of MFs,  $x_j, x_{j+1}, \dots, x_J$  are the inputs,  $\tilde{A}_{j,k}, \tilde{A}_{j+1,k}, \dots, \tilde{A}_{J,k}$  are the IT2FS,  $f_i$  are the  $i$ -th rule output,  $C_{i,j}, C_{i,j+1}, \dots, C_{i,J}$  are the consequent parameter, and  $C_{i,J+1}$  is the independent parameter.

In an IT2 TSK model, lower and upper weights of the  $i$ -th rule, i.e.,  $w_i$  and  $\bar{w}_i$ , are given by Eq. (3). To calculate the final output of the IT2 TSK, it is necessary to use a type reducer algorithm as mentioned above. The proposed OSTRA is described in detail below. Equation (4) can be rewritten as follows:

$$y = \frac{w_n \left( \frac{w_n + \bar{w}_n}{2} \right) + w_{n+1} \left( \frac{w_{n+1} + \bar{w}_{n+1}}{2} \right) + \dots + w_N \left( \frac{w_N + \bar{w}_N}{2} \right)}{\left( \frac{w_n + \bar{w}_n}{2} \right) + \left( \frac{w_{n+1} + \bar{w}_{n+1}}{2} \right) + \dots + \left( \frac{w_N + \bar{w}_N}{2} \right)}, \tag{23}$$

$$\bar{y} = \frac{\bar{w}_n \left( \frac{w_n + \bar{w}_n}{2} \right) + \bar{w}_{n+1} \left( \frac{w_{n+1} + \bar{w}_{n+1}}{2} \right) + \dots + \bar{w}_N \left( \frac{w_N + \bar{w}_N}{2} \right)}{\left( \frac{w_n + \bar{w}_n}{2} \right) + \left( \frac{w_{n+1} + \bar{w}_{n+1}}{2} \right) + \dots + \left( \frac{w_N + \bar{w}_N}{2} \right)} \tag{24}$$

the weights  $w_n, w_{n+1}, \dots, w_N$  and  $\bar{w}_n, \bar{w}_{n+1}, \dots, \bar{w}_N$  are sorted in incremental order, so the new index  $n$  instead of  $i$ . The Eqs. (23) and (24) can be factored as follows:

$$y = \frac{w_n(w_n + \bar{w}_n) + w_{n+1}(w_{n+1} + \bar{w}_{n+1}) + \dots + w_N(w_N + \bar{w}_N)}{(w_n + \bar{w}_n) + (w_{n+1} + \bar{w}_{n+1}) + \dots + (w_N + \bar{w}_N)}, \tag{25}$$

$$\bar{y} = \frac{\bar{w}_n(w_n + \bar{w}_n) + \bar{w}_{n+1}(w_{n+1} + \bar{w}_{n+1}) + \dots + \bar{w}_N(w_N + \bar{w}_N)}{(w_n + \bar{w}_n) + (w_{n+1} + \bar{w}_{n+1}) + \dots + (w_N + \bar{w}_N)}. \tag{26}$$

The reference points  $y$  and  $\bar{y}$ , unlike in type reduction algorithms with iterative processes to calculate the centroid, i.e., KM, in OSTRA it is only calculated once time. The main difference of OSTRA is to use the weights  $w_i$  and  $\bar{w}_i$  instead of the output of the rules  $f_i$ . To find the point of change  $1 \leq "l" \leq N-1$  and  $1 \leq "r" \leq N-1$ , these conditions  $w_n < y \leq w_{n+1}$  and  $\bar{w}_n < \bar{y} \leq \bar{w}_{n+1}$ , respectively, must be satisfied. For example, suppose the point of change "l" and "r" were found at middle, both of them. Then, the

average of  $\underline{X}$  and  $\bar{X}$  from Eq. (5), for example, when  $N=4$  is represented as follows:

$$X = \frac{\left[ \frac{w_n + \bar{w}_{n+1} + w_{n+2} + \bar{w}_N}{w_n + w_{n+1} + w_{n+2} + w_N} \right] + \left[ \frac{w_n + w_{n+1} + \bar{w}_{n+2} + \bar{w}_N}{w_n + w_{n+1} + \bar{w}_{n+2} + w_N} \right]}{2}. \tag{27}$$

When the vector  $X$  is calculated, it is necessary to do a reordering to return to its previous order; for this reason, the indices are changed from  $n$  to  $i$  as shown in Eq. (6). Finally to calculate the output of IT2 TSK model, the Eq. (7) can be rewritten as follows:

$$y = w_i(C_{i,j}x_j + C_{i,j+1}x_{j+1} + \dots + C_{i,J}x_J + C_{i,J+1}) + w_{i+1}(C_{i+1,j}x_j + C_{i+1,j+1}x_{j+1} + \dots + C_{i+1,J}x_J + C_{i+1,J+1}) + \dots + w_N(C_{N,j}x_j + C_{N,j+1}x_{j+1} + \dots + C_{N,J}x_J + C_{N,J+1}) \tag{28}$$

**Data availability** [https://drive.google.com/drive/folders/1T5KV7L\\_KZF9rAFHmLkrUvWx5EW\\_lqY-Q?usp=sharing](https://drive.google.com/drive/folders/1T5KV7L_KZF9rAFHmLkrUvWx5EW_lqY-Q?usp=sharing).

**References**

1. Zadeh, L.A.: Fuzzy sets. *Inf. Control* **8**(3), 338–353 (1965). [https://doi.org/10.1016/S0019-9958\(65\)90241-X](https://doi.org/10.1016/S0019-9958(65)90241-X)
2. Karnik, N.N., Mendel, J.M., Liang, Q.: Type-2 fuzzy logic systems. *IEEE Trans. Fuzzy Syst.* **7**(6), 643–658 (1999). <https://doi.org/10.1109/91.811231>
3. Antão, R.: Type-2 fuzzy logic: uncertain systems' modeling and control. Springer, Berlin (2017). <https://doi.org/10.1007/978-981-10-4633-9>
4. Torshizi, A.D., Zarandi, M.H.F., Zakeri, H.: On type-reduction of type-2 fuzzy sets: a review. *Appl. Soft Comput.* **27**, 614–627 (2015). <https://doi.org/10.1016/j.asoc.2014.04.031>
5. Karnik, N.N., Mendel, J.M.: Centroid of a type-2 fuzzy set. *Inf. Sci.* **132**(1–4), 195–220 (2001). [https://doi.org/10.1016/S0020-0255\(01\)00069-X](https://doi.org/10.1016/S0020-0255(01)00069-X)
6. Wu, D., Mendel, J.M.: Enhanced Karnik–Mendel algorithms. *IEEE Trans. Fuzzy Syst.* **17**(4), 923–934 (2008). <https://doi.org/10.1109/TFUZZ.2008.924329>
7. Duran, K., Bernal, H., Melgarejo, M.: Improved iterative algorithm for computing the generalized centroid of an interval type-2 fuzzy set. In: *NAFIPS 2008–2008 Annual Meeting of the North American Fuzzy Information Processing Society*. IEEE, pp. 1–5 (2008). <https://doi.org/10.1109/NAFIPS.2008.4531244>
8. Wu, D., Nie, M.: Comparison and practical implementation of type-reduction algorithms for type-2 fuzzy sets and systems. In: *2011 IEEE International Conference on Fuzzy Systems (FUZZ-IEEE 2011)*. IEEE, pp. 2131–2138 (2011). <https://doi.org/10.1109/FUZZY.2011.6007317>
9. Mittal, K., Jain, A., Vaisla, K.S., Castillo, O., Kacprzyk, J.: A comprehensive review on type 2 fuzzy logic applications: past, present and future. *Eng. Appl. Artif. Intell.* **95**, 103916 (2020). <https://doi.org/10.1016/j.engappai.2020.103916>
10. Jang, J.S.: ANFIS: adaptive-network-based fuzzy inference system. *IEEE Trans. Syst. Man Cybern.* **23**(3), 665–685 (1993). <https://doi.org/10.1109/21.256541>

11. Karaboga, D., Kaya, E.: Adaptive network based fuzzy inference system (ANFIS) training approaches: a comprehensive survey. *Artif. Intell. Rev.* **52**(4), 2263–2293 (2019). <https://doi.org/10.1007/s10462-017-9610-2>
12. Zhu, S., Tan, M.K., Lim, K.G., Chin, R.K.Y., Chua, B.L., Teo, K.T.K.: Engine misfire fault diagnosis based on SC-ANFIS. *J. Intell. Fuzzy Syst.* (2023). <https://doi.org/10.3233/JIFS-224059>. (Preprint)
13. Li, J., Yan, G., Abbud, L.H., Alkhalifah, T., Alturise, F., Khadimallah, M.A., Marzouki, R.: Predicting the shear strength of concrete beam through ANFIS-GA-PSO hybrid modeling. *Adv. Eng. Softw.* **181**, 103475 (2023). <https://doi.org/10.1016/j.advengsoft.2023.103475>
14. Dash, K.K., Sundarsingh, A., BhagyaRaj, G.V.S., Pandey, V.K., Kovács, B., Mukarram, S.A.: Modelling of ultrasonic assisted osmotic dehydration of cape gooseberry using adaptive neuro-fuzzy inference system (ANFIS). *Ultrason. Sonochem.* **96**, 106425 (2023). <https://doi.org/10.1016/j.ultrsonch.2023.106425>
15. Moayedi, H., Le Van, B.: The applicability of biogeography-based optimization and earthworm optimization algorithm hybridized with ANFIS as reliable solutions in estimation of cooling load in buildings. *Energies* **15**(19), 7323 (2022). <https://doi.org/10.3390/en15197323>
16. Takagi, T., Sugeno, M.: Fuzzy identification of systems and its applications to modeling and control. *IEEE Trans. Syst. Man Cybern.* **SMC-15**(1), 116–132 (1985). <https://doi.org/10.1109/TSMC.1985.6313399>
17. Mendez, G.M., De Los Angeles Hernandez, M.: Interval type-2 anfis. *Innov. Hybrid Intell. Syst.* **44**, 64–71 (2007). [https://doi.org/10.1007/978-3-540-74972-1\\_10](https://doi.org/10.1007/978-3-540-74972-1_10)
18. MonirVaghefi, H., Sandgani, M.R., Shoorehdeli, M.A.: Interval type-2 adaptive network-based fuzzy inference system (anfis) with type-2 non-singleton fuzzification. In: 2013 13th Iranian Conference on Fuzzy Systems (IFSC). IEEE, pp. 1–6 (2013). <https://doi.org/10.1109/IFSC.2013.6675612>
19. Tavooisi, J., Suratgar, A.A., Menhaj, M.B.: Stable ANFIS2 for nonlinear system identification. *Neurocomputing* **182**, 235–246 (2016). <https://doi.org/10.1016/j.neucom.2015.12.030>
20. Tavooisi, J., Mohammadi, F.: A new type-II fuzzy system for flexible-joint robot arm control. In: 2019 6th International Conference on Control, Instrumentation and Automation (ICCIA). IEEE, pp. 1–4 (2019). <https://doi.org/10.1109/ICCIA49288.2019.9030872>
21. Chen, C., John, R., Twycross, J., Garibaldi, J.M.: An extended ANFIS architecture and its learning properties for type-1 and interval type-2 models. In: 2016 IEEE International Conference on Fuzzy Systems (FUZZ-IEEE), pp. 602–609 (2016). <https://doi.org/10.1109/FUZZ-IEEE.2016.7737742>
22. Chen, C., John, R., Twycross, J., Garibaldi, J.M.: Type-1 and interval type-2 ANFIS: a comparison. In: 2017 IEEE International Conference on Fuzzy Systems (FUZZ-IEEE), pp. 1–6 (2017). <https://doi.org/10.1109/FUZZ-IEEE.2017.8015555>
23. Chen, C.: A novel framework for the implementation and evaluation of type-1 and interval type-2 ANFIS (Doctoral dissertation, University of Nottingham). (2018). <http://eprints.nottingham.ac.uk/id/eprint/49442>
24. Öztürk, M., Özkol, İ.: Comparison of modified Karnik–Mendel algorithm-based interval type-2 ANFIS and type-1 ANFIS. *Aircr. Eng. Aerosp. Technol.* **93**(10), 1526–1532 (2021). <https://doi.org/10.1108/AEAT-11-2020-0268>
25. Liang, Q., Mendel, J.M.: Interval type-2 fuzzy logic systems: theory and design. *IEEE Trans. Fuzzy Syst.* **8**(5), 535–550 (2000). <https://doi.org/10.1109/91.873577>
26. MatLab: Neuro-Adaptive Learning and ANFIS. MathWorks. <https://la.mathworks.com/help/fuzzy/neuro-adaptive-learning-and-anfis.html>, accessed 20 Jun 2023
27. Abiyev, R.H., Kaynak, O., Kayacan, E.: A type-2 fuzzy wavelet neural network for system identification and control. *J. Franklin Inst.* **350**(7), 1658–1685 (2013). <https://doi.org/10.1016/j.jfranklin.2013.04.020>
28. Tavooisi, J., Badamchizadeh, M.A.: A class of type-2 fuzzy neural networks for nonlinear dynamical system identification. *Neural Comput. Appl.* **23**, 707–717 (2013). <https://doi.org/10.1007/s00521-012-0981-7>
29. Wakchaure, M., Patle, B.K., Mahindrakar, A.K.: Application of AI techniques and robotics in agriculture: a review. *Artif. Intell. Life Sci.* **3**, 100057 (2023). <https://doi.org/10.1016/j.ailesci.2023.100057>
30. Borrero, G.H., Becker, M., Archila, J.F., Bonito, R.: Fuzzy control strategy for the adjustment of the front steering angle of a 4WSD agricultural mobile robot. In: 2012 7th Colombian Computing Congress (CCC) IEEE, pp. 1–6 (2012). <https://doi.org/10.1109/ColombianCC.2012.6398024>
31. Kannan, P., Natarajan, S.K., Dash, S.S.: Design and implementation of fuzzy logic controller for online computer controlled steering system for navigation of a teleoperated agricultural vehicle. *Math. Probl. Eng.* (2013). <https://doi.org/10.1155/2013/590861>
32. Barakat, M.H., Azar, A.T., Ammar, H.H.: Agricultural service mobile robot modeling and control using artificial fuzzy logic and machine vision. In: The International Conference on Advanced Machine Learning Technologies and Applications (AMLTA2019). Springer International Publishing **921**, 453–465 (2019). [https://doi.org/10.1007/978-3-030-14118-9\\_46](https://doi.org/10.1007/978-3-030-14118-9_46)
33. Kayacan, E., Kayacan, E., Ramon, H., Kaynak, O., Saeyns, W.: Towards agrobots: trajectory control of an autonomous tractor using type-2 fuzzy logic controllers. *IEEE/ASME Trans. Mechatron.* **20**(1), 287–298 (2014). <https://doi.org/10.1109/TMECH.2013.2291874>
34. Kayacan, E., Saeyns, W., Kayacan, E., Ramon, H., Kaynak, O.: Intelligent control of a tractor-implement system using type-2 fuzzy neural networks. In: 2012 IEEE international conference on fuzzy systems IEEE, pp. 1–8 (2012). <https://doi.org/10.1109/FUZZ-IEEE.2012.6250790>

Springer Nature or its licensor (e.g. a society or other partner) holds exclusive rights to this article under a publishing agreement with the author(s) or other rightsholder(s); author self-archiving of the accepted manuscript version of this article is solely governed by the terms of such publishing agreement and applicable law.



**Adrián Alberto-Rodríguez** is currently studying for a PhD in Computer Science at the Autonomous University of Hidalgo. He is currently working on the application of fuzzy interval type-2 modelling and control techniques for a tractor robot in agricultural applications. He is interested in studying fuzzy modelling and control in real systems.



**Virgilio López-Morales** works as a Researcher Professor of Computational Intelligence at the Information Tech. & Syst. Research Center for Computational Sciences at the Autonomous University of Hidalgo. He is currently working on Algorithms and Artificial Intelligence for a project: 'Decision-Making Support Systems.' His current research interest is on "Fuzzy Decision-Making," "Nonlinear Optimization applied on Decision-Making,"

and "Reliable Algorithms for Assessing Decision-Making Methods under uncertainty." Besides, some SW and HW applications can be found as a part of his applied systems.



**Julio Cesar Ramos-Fernández** received the Ph.D. degree in the framework of international cooperation between France-Mexico ECOS-NORD/ANUIES-SEP-CONACYT, in 2008 graduated with a double degree in the field of Computer Science in Mexico and Ph.D. in Engineering and Applied Sciences in France. He is professor at the Polytechnic University of Pachuca in the mechatronics department, since 2008. His research interests focus on mechatronic systems, fuzzy modeling and control applied to precision agriculture.