

CONTROL HÍBRIDO PID-DIFUSO EN ROBOT SEGUIDOR DE LÍNEA NO HOLONÓMICO

Franyelit Suarez¹, Enio Díaz², Sergio Velásquez³, Jairo Velásquez⁴

¹Pontificia Universidad Católica del Ecuador, Sede Esmeraldas, Ecuador, frangelits@gmail.com

^{2,3 y 4}UNEXPO Vicerrectorado Puerto Ordaz, Ciudad Guayana, Bolívar 8050 Venezuela.

E-mail: eniodiazrios@gmail.com, frangelits@gmail.com

Recibido (28/05/18), aceptado (10/07/18)

Resumen: Este artículo representa la fabricación de un robot de seguidor de línea con algoritmo de control híbrido basado en lógica difusa. Para el control de posición del robot se implementó un control PD+I difuso y para el control de velocidad se utilizó un controlador PI con realimentación odométrica, el robot en estudio es un robot seguidor de línea tipo tracción diferencial. Desde un punto de vista matemático podemos considerar el robot móvil tipo tracción diferencial como un robot no holonómicos y sus ecuaciones cinemáticas, dinámicas y de control son muy complejas. De la consideración anterior la lógica difusa es una buena alternativa para el desarrollo de controladores basado en inteligencia artificial y la experticia del proceso a controlar.

Palabras Clave: Controlador difuso, microcontrolador, cinemática, edometría, robótica.

HYBRID PID-DIFFUSED CONTROL IN ROBOT NON-HOLONOMIC LINE FOLLOWER

Abstract: This paper represents the manufacture of a line follower robot with a hybrid control algorithm based on fuzzy logic. To control the position of the robot, a diffuse PD + I control was implemented and for the speed control, a PI controller with odometrical feedback was used. The robot under study is a line tracker of the differential traction type. From a mathematical point of view we can consider the mobile robot type differential traction as a non-holonomic robot and its kinematic, dynamic and control equations are very complex. From the previous consideration the fuzzy logic is a good alternative for the development of controllers based on artificial intelligence and the expertise of the process to be controlled.

Keywords: Fuzzy controller, microcontroller, kinematics, edometry, robotics.

I. INTRODUCCIÓN

Los robots seguidores de línea son un tipo de robots móvil de navegación guiada, que se encarga de seguir una trayectoria planeada sobre una superficie. Abarcan un gran campo de aplicaciones, en la industria se les denomina vehículo guiado automatizado y su función principal está relacionada con el transporte de mercancías dentro de la misma empresa.

Desde su punto de vista matemático un robot móvil tipo tracción diferencial, se puede clasificar como un robot no holonómicos, en que su movimiento se produce por dos desplazamientos, y el vehículo solo posee dos grados de libertad, por los que el análisis cinemático, dinámico y de control son más complejos.

El presente estudio trata en la construcción de un robot móvil seguidor de línea con algoritmo de control basado en lógica difusa, que da respuesta a la problemática del control basado en un modelo matemático definido por la cinemática del mismo. Para lograr la autonomía del robot móvil el Kernel del control se desarrolló en dos partes, la primera un control de velocidad de las ruedas laterales basadas en la realimentación de odometría mediante un control PI clásico y la segunda en un control de posición del robot basado en un control PD+I difuso.

Para la validación del comportamiento del robot, nos basamos el modelo cinemático del robot desarrollado en Matlab-Simulink, para posteriormente verificar este comportamiento con el robot utilizando una interfaz de comunicación desarrollada en Labview, que muestra un mapeo del robot ante una trayectoria recorrida así como, las salidas del modelo cinemático corriendo el Kernel de controlador. Como hardware de control utilizamos un Dspic33ep256mu806, la comunicación con Labview la realizamos de forma inalámbrica con módulos Xbee series 2.

II. MARCO TEÓRICO

2.1 Modelado de cinemático robots diferenciales.

Un robot móvil tipo tracción diferencial cuenta con dos ruedas laterales de tracción, que usualmente tienen acoplados codificadores rotativos, una o dos ruedas de estabilización que mantiene el balance axial del vehículo. La traslación y rotación de este tipo de robot, clasificado como no holonómicos, está determinado por el movimiento independiente de sus ruedas de tracción. Figura. 1.

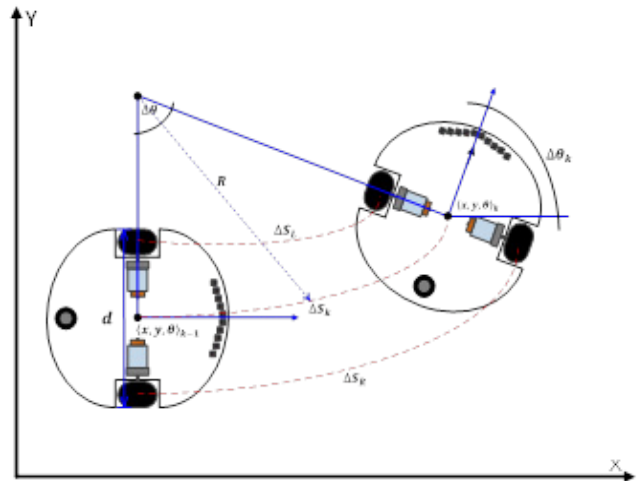


Figura 1. Desplazamiento relativo entre instante de muestreo. ΔS_k es el arco recorrido por el centro del robot y $\Delta\theta_k$ es el giro diferencial en el intervalo $[k-1, k]$. d es la distancia entre las ruedas motrices

De la figura 1 podemos obtener las siguientes expresiones:

$$\Delta S_k = (\Delta S_R + \Delta S_L)/2 \quad (1)$$

$$\Delta\theta_k = (\Delta S_R - \Delta S_L)/d \quad (2)$$

Aquí ΔS_R y ΔS_L son las distancias recorridas por la rueda derecha e izquierda respectivamente, estas distancias pueden ser calculadas de la cinemática rueda-sensor como se muestra en la figura 2.

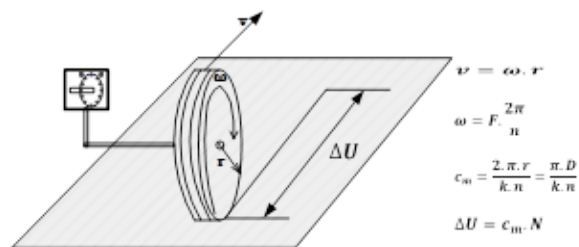


Figura 2. Medición del desplazamiento lineal de las ruedas laterales mediante codificadores rotativos. Donde N corresponde a la cantidad de pulsos que genera el encoder en el intervalo ΔU y $c_m = \frac{2\pi r}{k \cdot n}$ se conoce como constante de sensibilidad odométrica y esta expresada en milímetros por pulsos.

Si mantenemos la frecuencia de muestreo constante y aplicando el teorema de Nyquist-Shannon, mantenemos una elevada tasa de muestreo sobre nuestro sistema de odometría de un robot diferencial, podemos estimar su posición y orientación del mismo mediante las siguientes ecuaciones en diferencia.

$$\mathbf{x}_k = \mathbf{x}_{k-1} + \Delta S_k \cos(\theta_{k-1} + \Delta\theta_k/2) \quad (3)$$

$$\mathbf{y}_k = \mathbf{y}_{k-1} + \Delta S_k \sin(\theta_{k-1} + \Delta\theta_k/2) \quad (4)$$

$$\theta_k = \theta_{k-1} + \Delta\theta_k \quad (5)$$

Estas expresiones representa el modelo cinemático por estimación odométrica, que nos permiten determinar la posición y orientación del robot en un instante $[k-1, k]$, ante una planeación guiada

2.2 Modelado del motor DC

La odometría es una parte importante del sistema de localización del robot móvil tipo tracción diferencial, mejorar la exactitud y precisión de los parámetros que modelan conduce a una mejor navegación, es por ello que al hablar de odometría nos referimos a encoder acoplados a los actuadores del robot, en la mayoría de los casos motores DC.

Definir las ecuaciones que modelan el motor izquierdo y derecho del robo móvil, vienen determinadas por el voltaje aplicado y por la salida de velocidad angular que produce dicho voltaje, tomando en cuenta que los motores tienen cajas reductoras para poder lograr un buen torque. La función de transferencia según Azcón [1]:

$$H(s) = \frac{k_t \cdot r_2}{(L_a s + R_a) \cdot (J_{eq} s + b_{eq}) + k_t k_v r_2} \quad (6)$$

Donde:

k_t es la constante del motor.

k_v es la constante de velocidad del motor.

r_2 es la máxima eficiencia del conjunto motor-reductor.

L_a es la inductancia del motor.

b_{eq} es la inercia del equivalente en el primario del motor.

b_{eq} es la resistencia viscosa en el primario del motor.

2.3 Controladores difusos.

El control difuso, puede ser expresado mejor como un control basado en la experiencia que interpreta el sentido común, en lugar de números o ecuaciones. Las variables de procesos no se miden en sentido común, sino en números por consiguiente se hace necesario una adaptación previa antes de introducir el estado de la variable del controlador esta etapa se denomina fusificación [2], como puede observarse en la figura 3.

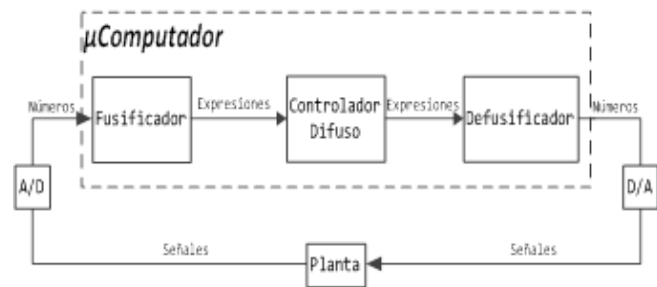


Figura 3. Lazo de un controlador difuso.

La etapa de Fusificación se encarga de transformar la variable de entrada a conjuntos difusos con un grado de membresías dentro de las distintas expresiones posibles, para ello busca una correspondencia entre el estado de la variable de entrada, y las funciones de pertenencias definidas por el experto. Determinados los estados de las variables de entradas sus expresiones lingüísticas se pueden establecer las bases del conocimiento mediante métodos de expertos que permiten definir reglas tales como si...entonces (IF-THEN). Se define unas series de relaciones que interpreta el sentido común, y permite generar una acción de control definida por el experto, que debe ser aplicada a la planta. Luego la acción de control debe ser traducida a un valor real o número (Defusificación).

III. DISEÑO PROPUESTO

3.1 Diseño propuesto del robot móvil, Hardware.

Unos de los objetivos primordiales de la presente investigación son el diseño y construcción de un robot móvil tipo tracción diferencial. En este punto de la investigación se parte de los componentes disponibles en la Tabla I. como punto de partida se cuenta con los motores DC y sus respectivos encoder, tomando en consideración que los motores tienen que estar los más cerca entre ellos para lograr reducir el espacio para la distribución de los componentes, ya que se utiliza como base de apoyo de los motores la misma placa de circuito impreso (PCB), Fig.4 y Fig.5.

Tabla I componentes utilizados en el robot móvil

Cantidad	Descripción	ID parte
2	Motores DC 7.2V 50:1	HN-GH7.2-2414T
2	Ruedas de goma 56x25 mm	Tamiya 70111
1	Rueda loca de acero	Tamiya 70144
1	Microcontrolador DSP	dSPIC33ep256mu806
1	Batería de 7.2 voltios 3300mAh	Tenergy 7.2V 3300mah
1	Driver control de Motores	Pololu #: 708
10	Módulos sensores IR	CNY70
12	Led Neopixel WS2812B	COM-13667
2	Módulos Xbee Series 2 – 2mW	WRL-12571
1	Tarjeta Modulo Xbee Sparkfun	BOB-08276
1	Explore Xbee USB Sparkfun	WRL-11812
2	Placas de circuito impreso FR4	Mouser: 590-555
1	Regulador de voltaje de 3.3V	Pololu #: 2111

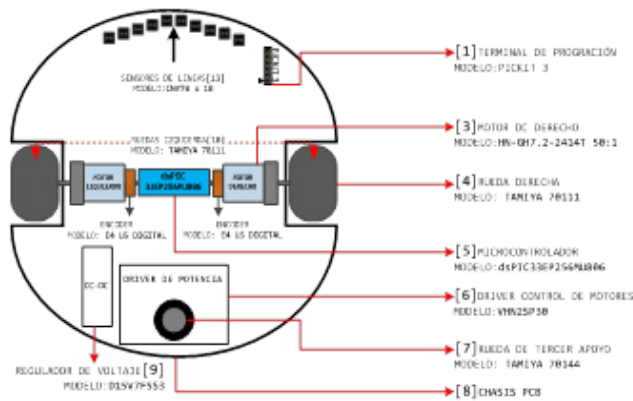


Figura 4. Diseño del hardware robot móvil, placa inferior.

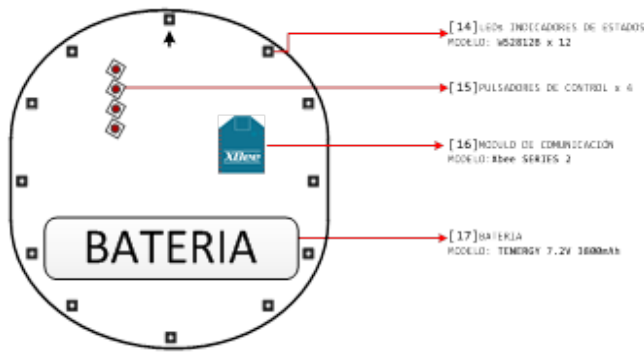


Figura 5. Diseño del hardware robot móvil, placa superior.

3.2 Diseño del control robot móvil, Software.

El diseño del control del robot móvil, plantea un control de posición y orientación del robot basado en un controlador Difuso PD+I y para el control de velocidad de las ruedas motrices un control PI, ambos controladores están embebido en el DSP del robot conformado por un microcontrolador dspic33ep256mu806 de microchip, la razón principal del uso de este microcontrolador es que incorpora dos módulo de interface en cuadratura (QEI), muy útil para leer los encoder de las rudas laterales. En la figura 6 se muestra el diagrama en bloque de la propuesta planteada.

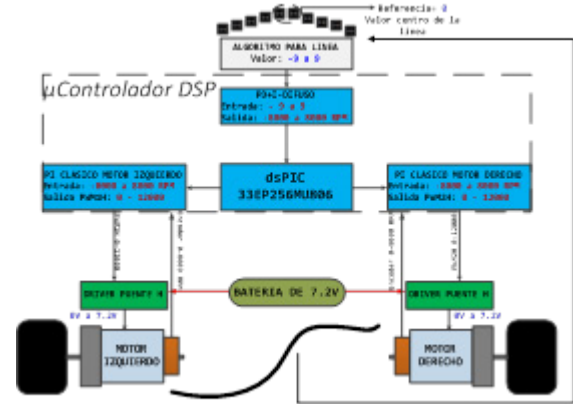


Figura 6. Diagrama en bloque del control propuesto. La lectura de la línea a seguir por el robot arroja un valor de -9 a 9, que es la entrada de nuestro controlador salida -8000 a 8000 RPM que representa la velocidad de referencia al control PI de las ruedas laterales del robot.

3.3 Diseño control PI de velocidad ruedas del robot.

Como se puede observar en la figura 6, el control de velocidad de las ruedas motrices del robot, es un control PI que se ejecuta en DSP del robot. Partiendo del modelo del motor DC descrito anteriormente y de la expresión 6, se hace una simulación en Matlab – Simulink, para posteriormente obtener las constantes de los parámetros del controlador K_p y K_i usando un script desarrollado en Matlab. Una vez obtenidas las constante y obtener la respuesta transitoria deseada se procede a crear el código en C que corre en el Kernel DSP.

En la tabla 2, se puede observar los datos del motor DC HN-GH7.2-2414T, que es un motor que trabaja a 7.2V con una caja reductora de 50:1 y un torque de 7.13 Kg-cm, el motor gira ante una entrada de 7.2V a 8500 RPM.

Tabla II parámetros de fabricación motor HN-GH7.2-2414T

Parámetros	Valor
v	7300 RPM
K_t	0.0094 Nm/A
K_v	0.0094 V/rad/seg
r_2	0.5244
L_a	0.00024 H
R_a	11 Ω
J_{eq}	$2.9 \cdot 10^{-8}$ Kg.m ²
b_{eq}	$3.77 \cdot 10^{-10}$

Con los parámetros de la tabla anterior, la función de transferencia del motor DC queda de la siguiente manera:

$$H(S) = \frac{0.00493}{6.96 \cdot 10^{-12} S^2 + 3.19 \cdot 10^{-7} S + 3.9766 \cdot 10^{-5}} \quad (7)$$

De la expresión 7, procedemos a discretizar usando un tiempo de muestreo de 10ms, que será ejecutada en unas de las interrupciones disponibles en el DSP, la función de transferencia en tiempo discreto se puede observar en la expresión 8.

$$H(Z) = \frac{72.27Z + 0.08298}{Z^2 - 0.2865Z} \quad (8)$$

En la figura 7 se muestra el diagrama en bloque del controlador PI, el controlador PI tiene realimentación proveniente de la odometría de los encoder del robot.

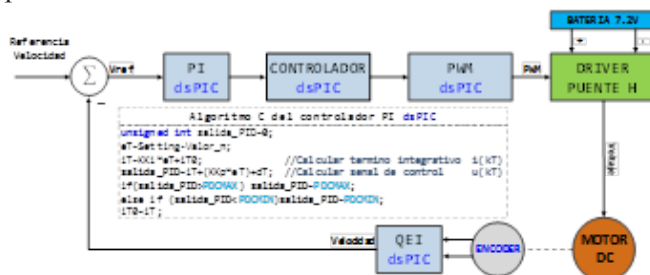


Figura 7. Diagrama en bloque del control de velocidad PI de las ruedas laterales del robot, usamos el módulo QEI del DSP para leer la velocidad del encoder.

3.4 Diseño del control de posición PD+I Difuso del robot.

El objetivo principal de la presente investigación es el desarrollo e implementación de un controlador basado en lógica difusa, que nos permita al robot el seguimiento de una trayectoria planeada, es ahí la importancia de este tipo de controlador que no necesitamos modelo cinemático y dinámico del robot para desarrollar el controlador sino el conocimiento de un experto.

Partiendo de un controlador PID clásico y de su respuesta transitoria de un sistema de segundo orden estable, iniciamos el diseño de un controlador PID difuso,

sabemos que para el rastreo de trayectoria usamos la acción proporcional y derivativa, y seleccionamos unas de las estructuras derivadas de los controles PID como es el PD+I difuso, como se muestra en la figura 8.

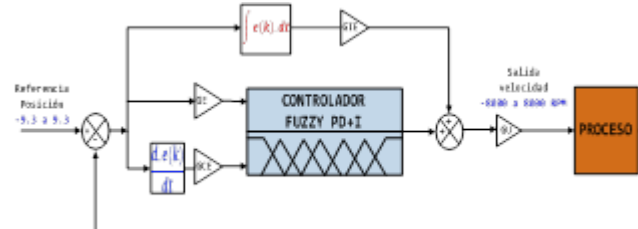


Figura 8. Estructura del controlador PD+I Difuso, el controlador se ejecuta en el Kernel del DSP, tienen como entrada la posición del robot y como salida la velocidad de referencia hacia el control PI de velocidad de las ruedas laterales del robot.

El controlador difuso del robot móvil es el encargado de mantener en trayectoria el robot móvil, mediante la lectura de la posición de referencia a seguir, en este caso la posición de referencia en cero, que representa un posición en X y Y P(0,0) y Angulo phi 0°, que es cuando el robot esta sobre la línea y centrado. Los sensores de posición conformados por 10 sensores CNY70, entrega la información de posición del robot, con valores de -9.3 cuando esta al extremo más izquierdo con respecto a su posición de referencia 0 y un valor de 9.3 cuando esta al extremo más derecho con respecto a su posición de referencia 0.

La salida de nuestro controlador difuso es el valor de referencia de velocidad hacia el controlador PI. En el robot la velocidad varía de 0 a 8000 RPM, por diferencia en sus velocidades el robot gira o no, de manera más rápida o lenta dependiendo el radio de curvatura.

3.5 Modelo del controlador PID Difuso.

Para el diseño del controlador se utilizaron las herramientas de dispuesta Matlab como el Fuzzy Logic Designer y en Labview como Fuzzy System Designer. Partimos de los datos de entradas y salidas del controlador difuso como puede observarse en la figura 8. Establecemos que nuestro controlador tiene dos entradas y una salida, procedemos de la siguiente manera:

A. Reglas de control.

Para definir las reglas base de controlador PID difuso, consideramos las siguientes notaciones:

- NG: Negativo grande.
- NM: Negativo mediano.
- NP: Negativo pequeño.
- Z: Cero.
- PP: Positivo pequeño.
- PM: Positivo mediano.
- PG: Positivo grande.

De la cual obtenemos la siguiente matriz FAM con la implicación de las reglas difusas como se muestra en la siguiente tabla III.

Tabla III. Reglas de control, para controlador PID Difuso

$\Delta e(k)$	NG	NM	NG	Z	PG	PM	PG
NG	NG	NG	NG	NG	NM	NP	Z
NM	NG	NG	NM	NM	NP	Z	Z
NP	NG	NM	NM	NP	Z	Z	PP
Z	NM	NP	NP	Z	PP	PP	PM
PP	NP	Z	Z	PP	PM	PM	PG
PM	Z	Z	PP	PM	PM	PG	PG
PG	Z	PP	PM	PG	PG	PG	PG

B. Funciones de pertenencias.

Para definir las funciones de pertenencias utilizamos la herramienta de Labview, Fuzzy System Designer. El universo de discurso para el error y la derivada del error fue ± 9.3 , como nuestro sistema difuso esta implementado en un microcontrolador DSP, se utilizan funciones de pertenencias trapezoidales y triangulares, como se aprecia en la figura 9.

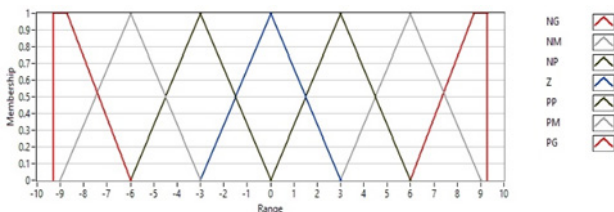


Figura 9. Función de pertenencias para el error y la derivada del error, con un universo de discurso de -9.3 a 9.3.

Para la salida del controlado difuso, utilizamos funciones de pertenencias triangulares, se empleó un universo de discurso de ± 10000 RPM figura 10, la salida del control PID difuso en la referencia al control PI de velocidad del robot móvil.

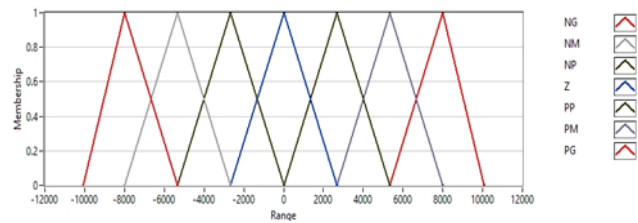


Figura 10. Función de pertenencias para salida del controlador con universo de discurso -10000 RPM a 10000 RPM

C. Mecanismo de inferencia.

Utilizamos mecanismo de inferencia por factores de cofinancia, donde dependiendo de los grados de activación de cada regla por las variables de entradas del error y la derivada del error, obtenemos una variable de salida. Utilizamos el operador lógico AND como el mínimo de dos factores, para luego combinar estos resultados mediante el operador lógico OR como el máximo.

D. Método de defusificación.

Utilizamos el método propuesto por Mandani [2], obteniendo la salida por método promedio de los centros o máximos (COM), como se muestra en la siguiente expresión 9.

$$El \quad y_0 = \frac{\sum_{i=1}^n \alpha_i \cdot c_i}{\sum_{i=1}^n \alpha_i} \quad (9)$$

Uso del método COM es por su reducido costo computacional y que es el implementado en nuestro Kernel del controlador Difuso en el DSP. La salida del controlador difuso se muestra en la figura 11.

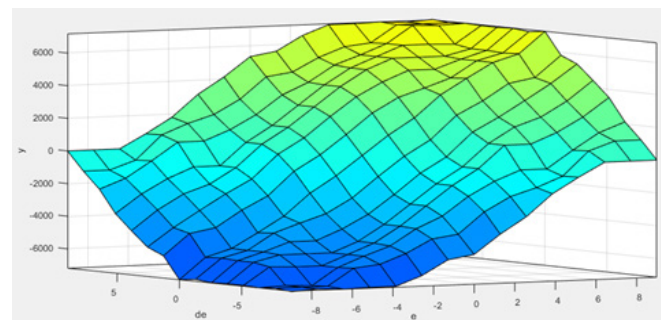


Figura 11. Salida del controlador difuso en función de las variables de entradas.

$$VEL = \frac{Fin * 60}{CPR} \quad (10)$$

3.6 Diseño del Kernel PD+I Difuso en el DSP.

Para la implementación del Kernel PD+I Difuso en el DSP del robot móvil, se procedió al uso de las Matlab-Simulink y Labview, ambas herramienta presentan ventajas y desventajas, en la cual se aprovecharon las

ventajas de cada una, para poder desarrollar el Kernel Difuso implementado en lenguaje C.

Se crearon modelo del PID difuso en Labview y Matlab con las herramientas dispuestas para tal fin, se compararon ambas salidas y observando la salida del controlador en Fuzzy System Designer de Labview, procedemos a codificar el condigo en C. el uso de Fuzzy System Designer es, porque nos muestra el que reglas se activaron en función de sus entradas y el grado de pertenencia de la misma. En la tabla IV se muestra una tabla comparativa de la salida del controlador difuso de los modelos creados en Matlab, Labview y el Kernel desarrollado en C.

Tabla IV salidas de los modelos PID Difuso

Entradas		Salida MATLAB	Salida LABVIEW	Kernel en C
e(k)	du(k)	Y(k)	Y(k)	Y(k)
2000	1000	4400	4399.53	4399
-2000	1000	-1070	-1199.76	-1199
8000	-1000	11300	10500.2	10500
4000	-4000	2040	2000	2000
4000	4000	-2040	-2000	-2000
800	100	1590	1445.61	1445
7000	8000	10600	12000	12000
1000	8000	8310	9499.81	9499

Podemos observar, que en la salida del Kernel desarrollado en C, se comporta muy similar a los de los modelos creados en Matlab y Labview.

IV. RESULTADOS EXPERIMENTALES

4.1 Prueba de controlador PI de velocidad del robot.

Unos de los puntos de partida en la presente investigación fue la prueba de la odometría utilizada en el robot móvil, probamos los encoder utilizando el módulo QEI disponible en el DSP, establecemos la comunicación entre el robot y el computador utilizando los módulos Xbee, con la interfaz grafica desarrollada en Labview vemos los datos de velocidad de los motores izquierdo y derecho.

Partiendo de la expresión 10, que representa el cálculo de velocidad en RPM en función de la frecuencia generada por los encoder, y sabiendo que por cada vuelta que gira una rueda del robot produce 420 pulsos tenemos:

Donde:

***Fin** es la frecuencia generada por el encoder del robot.

***CPR** es el número de pulsos- vueltas del disco óptico.

***VEL** es la velocidad del motor medidas en RPM.

Para validar los resultados aplicamos un voltaje de 7.2V a los motores, tanto izquierdo y derecho del robot mó-

vil, para ver su respuesta transitoria a lazo abierto y obtenemos la siguiente respuesta, figura 12.

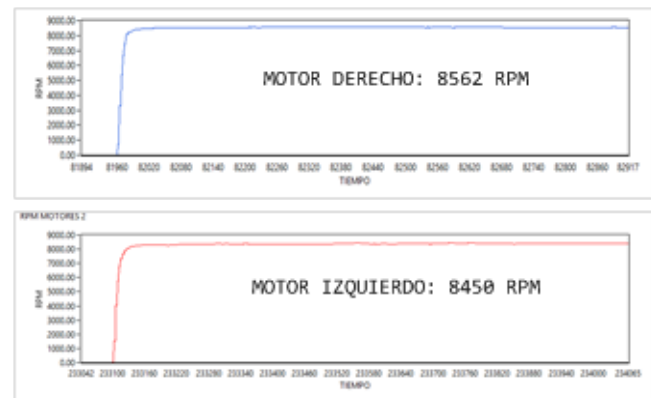


Figura 12. Respuesta temporal de los motores laterales del robot móvil ante una entrada de voltaje de 7.2V.

En la tabla V tenemos una comparación entre los datos suministrado por el fabricante y los obtenidos por DSP del robot.

Tabla V. Comparación de los resultados experimentales de velocidad

	DATOS FABRICANTE	MODELO SIMULINK	MOTOR DERECHO	MOTOR IZQUIERDO
Velocidad RPM	8750	8524	8562	8450
Error %	0	2.58283	2.14857	3.42857

Terminada la prueba de en lazo de abierto de los motores se procede a la prueba de lazo cerrado, obteniendo la siguiente respuesta para una entrada escalón de 4000 RPM, como se muestra en la figura 13 y 14.

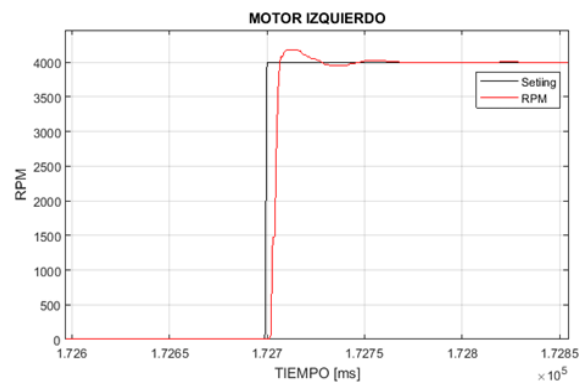


Figura 13. Respuesta transitoria motor izquierdo robot móvil.

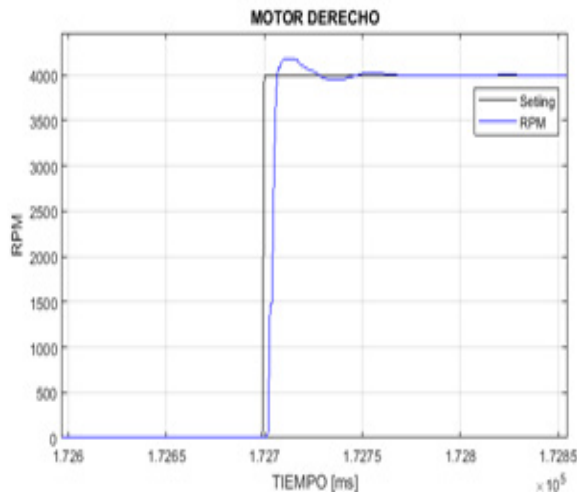


Figura 14. Respuesta transitoria motor derecho robot móvil.

Los parámetros de las repuestas temporales se muestran en la tabla VI, la entonación de los lazos de control fueron realizadas en Matlab usando un Script para tal fin.

Tabla VI. Parámetros repuestas transitorias del controlador PI

Tipo Controlador	Parámetros transitorios				
PI	Tr. (ms)	Ts. (ms)	Mp. %	Tp. (m)	(%)ess.
Motor Izquierdo	8.25	95.6	5	10.4	0.175
Motor Derecho	8.35	95.4	4.5	10.3	0.075

4.2 Prueba de controlador Difuso.

Para la prueba del controlador difuso del robot móvil, primero probamos el Kernel desarrollado en lenguaje corriendo en el DSP del robot, usamos como planta a controlar el modelo del motor DC utilizados en el robot móvil. Realizamos la simulación en Simulink para el comportamiento en la regulación de velocidad del motor DC, como se muestra en la figura 15. En la figura 16, se muestran las respuestas transitorias de la simulación.

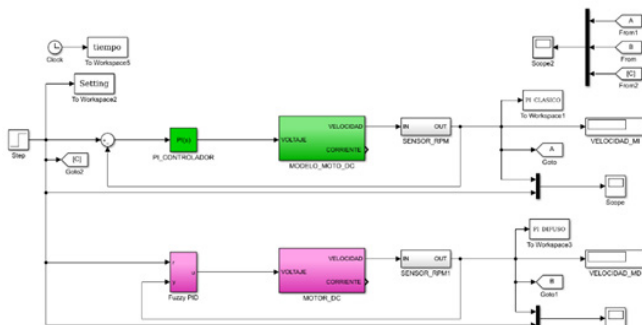


Figura 15. Simulación del controlador Difuso y controlador PI clásico en el control de un motor DC.

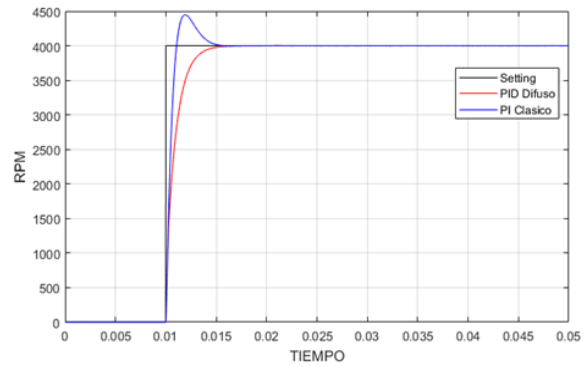


Figura 16. Respuesta transitoria de la simulación en Simulink del controlador Difuso y controlador PI clásico en el control de un motor DC.

En la tabla VII, se muestran los parámetros de las respuesta transitoria obtenida ante una señal escalo de 4000 RPM, podemos observar, que el controlador PID Difuso tiene mejor respuesta transitoria que el controlador PI clásico, se reduce el máximo sobre impulso (M_p) de manera considerable en comparación con el controlador PI clásico.

Para validar la simulación antes descrita, se procede a la prueba del Kernel difuso, que representa la segunda prueba descrita en esta sección.

Tabla VII. Respuesta transitoria controlador PI vs PID difuso Simulink

Tipo Controlador	Parámetros transitorios				
	Tr. (ms)	Ts. (ms)	Mp. %	Tp. (m)	(%)ess.
PI clásico	11.08	20.00	11.50	10.40	0.00
PI Difuso	15.05	15.09	15.05	10.3	0.00

Con la ayuda de Labview, se procedió a obtener las respuestas transitorias del Kernel desarrollado en C, como se muestra en la figura 17, repitiendo el procedimiento anterior y comparando los parámetros transitorios.

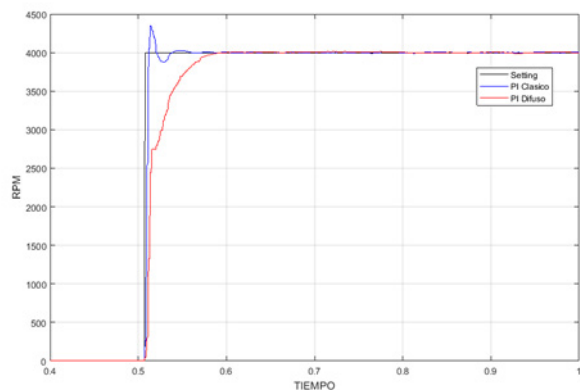


Figura 17. Respuesta transitoria del controlador Difuso

Difuso y controlador PI clásico en el control de un motor DC en el robot móvil.

Tabla VIII. Respuesta transitoria controlador PI vs PID difuso Motor DC

Tipo Controlador	Parámetros		transitorios		
	Tr.(ms)	Ts.(ms)	Mp.%	Tp.(m)	(%ess.
PI clásico	51.17	70.08	8.17	51.47	0.250
PI Difuso	59.49	71.45	0.30	61.00	0.120

En la tabla VIII, podemos observar que el controlador difuso presenta el mismo comportamiento que en la simulación desarrollada en Simulink, igualmente el parámetro más importante para tomar en consideración es el máximo sobre impulso (Mp).

Prueba de controlador Difuso en el seguimiento del trayectoria Robot móvil.

Para la prueba en el seguimiento de línea, se cuenta con una pista como se muestra en la figura 18, donde empleamos curvas de 90° como entrada escaló, curvas en sentido horario y antihorario y rectas largas, para observar los parámetros transitorios de los motores laterales del robot. Para observar el desplazamiento del robot en el eje X y Y, así como su ángulo Phi de desvío, se crea un modelo cinemático del robot diferencial en Labview como se muestran en la figura 19.

Las pruebas fueron realizadas en un ambiente controlado de luz artificial, con una pista con fondo blanco y una línea negra.

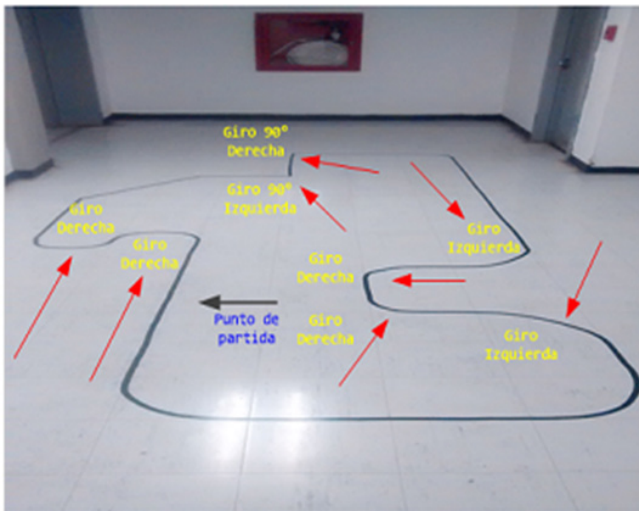


Figura 18. Pista de prueba para el seguimiento de trayectoria robot móvil.

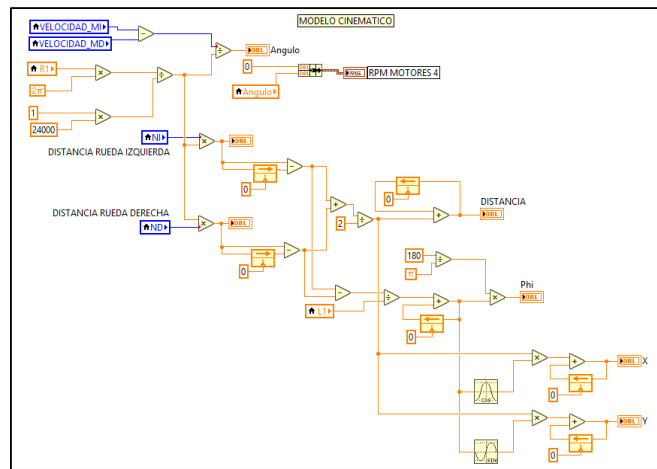


Figura 19. Modelo cinemático odométrico vi Labview robot móvil.

Para verificar el desempeño del robot, se evalúan tres diferentes tipos de control:

- CPD:** control de rastreo de línea mediante control PD.
- CPDO:** control de rastreo de línea mediante control PD y realimentación odométrica con control PI de velocidad.
- CFPDO:** controla de rastreo de línea mediante controla PD+I difuso y realimentación odométrica con control PI de velocidad.

La prueba se realiza a una velocidad constaten en línea recta de 4000 RPM en el eje motor o 80 RPM en la rueda del robot móvil, evaluamos el mapeo del robot ante la trayectoria mostrada en la figura 18, las respuesta temporales del motor izquierdo y derecho, así como el desplazamiento en el eje X y el eje Y además de su ángulo de desvío. También evaluamos el tiempo empleado para realizar un vuelta, así como la distancia total recorrida por el robot móvil.

Resultados control CPD

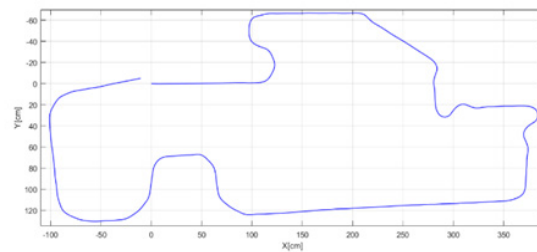


Figura 20. Trayectoria obtenida por estimación odométrica.

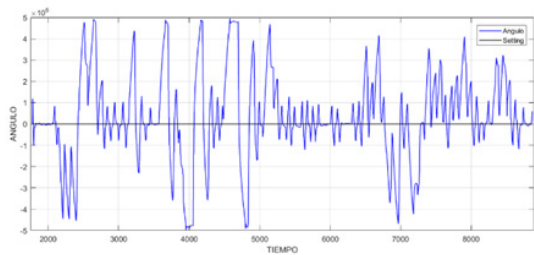


Figura 21. Variación del Angulo Phi en la trayectoria.

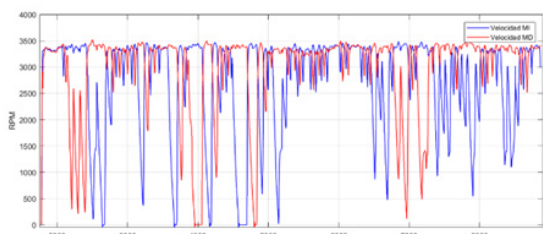


Figura 22. Velocidades ruedas laterales del robot móvil.

Resultados control CPDO

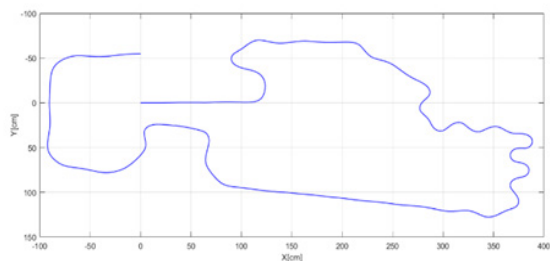


Figura 23. Trayectoria obtenida por estimación odométrica

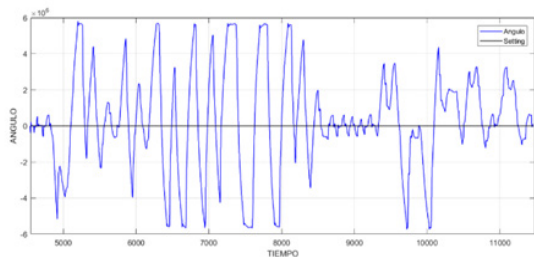


Figura 24. Variación del Angulo Phi en la trayectoria.

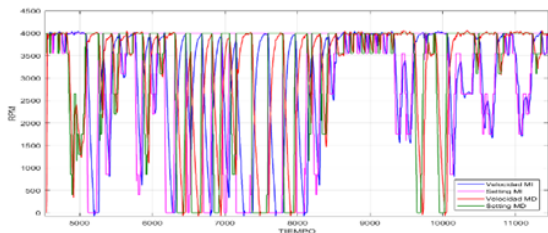


Figura 25. Velocidades ruedas laterales del robot móvil.

Resultados control CFPDO

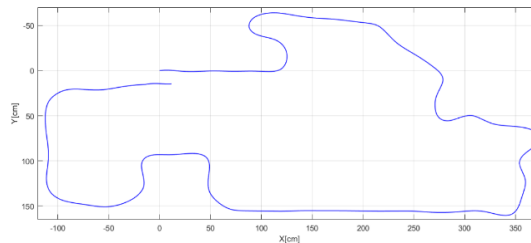


Figura 26. Trayectoria obtenida por estimación odométrica.

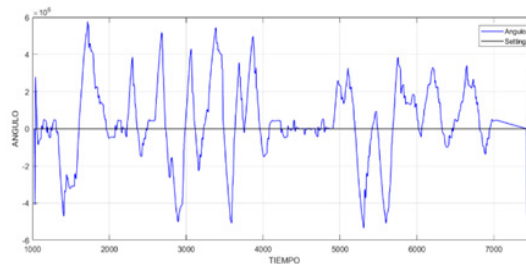


Figura 27. Variación del Angulo Phi en la trayectoria.

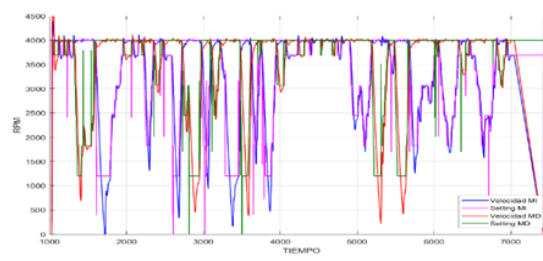


Figura 28. Velocidades ruedas laterales del robot móvil

Tabla IX. Tabla comparativa de tiempo y distancia

		CPD		
Vueltas	Tiempo	Tiempo total	Distancia[cm]	Distancia Total[cm]
1	1:23.28	1:23.28	1355.58	1355.58
2	1:23.45	2:46.74	1298.54	2654.12
3	1:23.94	4:10.69	1307.35	3961.47
4	1:25.43	5:36.12	1308.06	5269.53
5	1:21.55	6:57.68	1308.42	6577.95
		CPDO		
Vueltas	Tiempo	Tiempo total	Distancia[cm]	Distancia Total[cm]
1	1:20.52	1:20.52	1393.53	1393.53
2	1:19.25	2:39.78	1372.66	2766.19
3	1:20.14	3:59.92	1372.66	4138.85
4	1:19.44	5:19.36	1376.11	5514.96
5	1:19.80	6:39.26	1378.88	6893.84
		CFPDO		
Vueltas	Tiempo	Tiempo total	Distancia[cm]	Distancia Total[cm]
1	1:13.85	1:13.85	1385.36	1385.36
2	1:15.35	2:29.20	1337.26	2703.62
3	1:13.15	3:42.36	1344.84	4047.46
4	1:14.41	4:56.77	1336.19	5417.65
5	1:15.10	6:11.86	1344.10	6761.75

Para el control CPD en la figura 20, el mapeo por estimación odométrica se asemeja mucho al trayecto de la pista, presentando una respuesta sub amortiguada en los cruces de 90° hacia la izquierda y derecha, el tiempo empleado para cinco vueltas es 6:57.68 y un recorrido total de 6577.95 cm, el ángulo presenta oscilaciones sucesivas cuando esta línea y las velocidades de los motores laterales no permanece constante, esto debido a que no se tiene realimentación odométrica de velocidad.

En el control CPDO el mapeo de la trayectoria pista, presenta oscilaciones a lo largo del trayecto, presentando mejor desempeño cuando el robot se desplaza en línea recta, como se puede observar la figura 23, además que se desvía significadamente de la posición inicial P (0,0). El tiempo empleado el recorrido es de 6:39.26 para cinco vueltas y una distancia de 6893.84 cm, la variación del ángulo de desvió es bastante considerable a comparación del control CPD como puede observarse en la figura 24, el Tiempo empleado es mucho menor al control CPD, debido a la realimentación odométrica como se puede observar en la figura 25.

Por ultimo control basado en lógica difusa CFPDO por estimación odométrica, el mapeo presenta mejor desempeño que el control CPD y CPDO, como se observa en la figura 26, el ángulo de desvió también presenta mejor desempeño que los controles anteriores figura 27, al igual que comportamiento del control de velocidad de los motores laterales como se puede observar en la figura 28, el tiempo empleado para cinco vueltas 6:11.86 y una distancia de 6761.75 cm. Es tipo

de control PD+I difuso presento un desempeño muy aceptable al no depender del modelo cinemático para el desarrollo de control, y tiene mejor respuesta al cruce de 90° que los controles anteriores. En la tabla IX vemos el desempeño por vuelta de los tres controles utilizados en la pista.

V. CONCLUSIONES

Después de explorar los paradigmas dominantes en el diseño de controladores difusos, así como la implementación en el control de robots móviles, se presenta una estrategia de control para un sistema no lineal por realimentación odométrica que cumple con todas las características de desempeño del control convencional y que no depende estrictamente del modelo cinemático y dinámico del robot móvil tipo tracción diferencial.

Por lo tanto la lógica difusa cubre un amplio rango de aplicaciones en la robótica, mejorando de manera significativa cualquier control conocido y como fue de mostrado en el desarrollo de la presente investigación. El control PD+I por lógica difusa (CFPDO) cumple con el objetivo de control de seguimiento de trayectoria por realimentación odométrica, para las pruebas recorrido se redujo de manera significativa los tiempo empleados para tal fin, en comparación al control convencional PD sin realimentación odométrica (CPD) y el control PD con realimentación odométrica (CPDO).

La implementación de un Kernel con algoritmo difuso en controlador DSP, presenta un código más

complejo que su par en el control clásico, ya que la precisión en el Kernel difuso está determinado al número de entradas y salidas de sus funciones de membresías. Para la investigación en estudio se utilizaron un total de siete funciones de membresías para entradas y salidas lo que conduce a cuarenta y nueve reglas que tienen que diseñarse para el control CFPDO, lo que con lleva a muchas líneas de código en comparación al control convencional PD y PI utilizados en el control CPD y CPDO. Otro punto es que el Kernel de control CFPDO se combina el control clásico con el control difuso, teniendo un control híbrido y así aprovechamos las ventajas de los dos controles.

La construcción del robot móvil es uno de los objetivos específicos de la investigación, y fue uno de los que más tiempo se empleó para completarlo, al final del diseño nos permitió validar los modelos cinemáticos empleados en MATLAB® Simulink® con los arrojados por el robot con su modelo cinemático corriendo el Kernel del DSP, el uso de Toolbox de LABVIEW® en desarrollo de modelo difusos, así como sus herramientas gráficas nos permito una mejor interpretación del comportamiento del robot en las medición de la variables odométrica de velocidad y del modelos cinemático de posición X, Y y el ángulo de desvíoPhi.

La selección de un controlador difuso para como algoritmo de control de seguidor de línea, se debió a que no es necesario contar con un conocimiento detallado de la dinámica del sistema representada en expresiones matemática. Las pruebas realizadas en la pista, permitieron comprobar que el mejor desempeño lo presento el controlador basado en inteligencia artificial (CFPDO), que los controles tradicionales.

VI. REFERENCIAS

- [1] A. Azcon, Análisis y diseño del control de posición de un robot móvil con tracción diferencial, Catalunya, España: Memoria de la Universidad Rovira I Virgili, 2003.
- [2] E.H. Mamdani, Application of fuzzy algorithms for control of simple dynamic plant, vol. 121, New York, USA, 1974.
- [3] L. A. Zadeh, Fuzzy sets, 3 ed., vol. 8, 1965, pp. 338-353.
- [4] F. Villafuerte, «Diseño de un controlador por lógica Difusa aplicado a un Robot Manipulador,» Querétaro, México, 2012.
- [5] M. S. Uzer y N. Yilmaz, A real-time object tracking by using fuzzy controller for vision-based mobile robot, 22 ed., vol. 6, 2011, pp. 4800-4820.
- [6] F. Umar, A. Muhammad, U. Asad, A. Ghulam y H. Athar, Fuzzy Logic Reasoning System for Line Following Robot, 2014, pp. 244-248.
- [7] M. Togai y H. Watanabe, A VLSI Implementation of Fuzzy Inference: Toward an Expert on a Chip, vol. 38, 1986, pp. 147-163.
- [8] J. A. Shaw, The PID control Algorithm: How it works, how to tune it, and how to use it, 2003.
- [9] R. SHAKYA, K. RAJANWAL, S. PATEL y R. K. MAURYA, OPTIMIZATION AND DESIGNING OF PID, FUZZY & PID-FUZZY CONTROLLER, 1 ed., vol. 5, 2014.
- [10] F. A. Salem, Dynamic and Kinematic Models and Control for Differential Drive Mobile Robots, vol. 3, 2013.
- [11] S. B. Saat, «Dc Motor Speed Control Using Fuzzy Logic Controller,» Malaysia, 2014.
- [12] T. Ross, Fuzzy Logic with Engineering Applications, New Mexico, USA: WILEY, 2010.
- [13] P. Ponce, Inteligencia Artificial con Aplicaciones a la Ingeniería, México: Alfaomega, 2010.
- [14] J. M. Pérez, Memoria Mapeo con robot móvil: caracterización y modelado.
- [15] S. Parra, «Generación de Mapa de entorno para navegación de vehículo terrestre autónomo,» Santiago de Chile, Chile, 2014.
- [16] E. Pardo, «Quadcopter: Construcción, control de vuelo y navegación GPS,» Madrid, España, 2012.
- [17] M. Paletta, Inteligencia Artificial Básica, Ciudad Guayana, Venezuela: Fondo Editorial Uneg, 2010.
- [18] D. Navarro, G. B. Gilabert, L. H. Ríos y M. Bueno, Mejoras de la localización odométrica de un robot diferencial mediante la corrección de errores sistemáticos, 2007.
- [19] V. Mohan, «Fuzzy logic controller for an autonomous mobile robot,» Cleveland, USA, 2005.
- [20] T. B. Matías y M. I. D. Vicente, LÓGICA BORROSA.
- [21] A. Ismail, A. A. Zaman y K. Terashima, «Fuzzy Logic Approach for line Following Mobile Robot Using an Array of Digital Sensor,» 2006.
- [22] A. Ibrahim, Fuzzy Logic for Embedded System Applications, USA: Newnes, 2004.
- [23] E. Diaz, «DISEÑO DE UN ROBOT SEGUIDOR DE LINEA CON ALGORITMO DE DECISIÓN MULTIPLE,» Guayana, Venezuela, 2011.
- [24] A. Corripio, Control Automático de procesos: Teoría y Práctica, Mexico: Limusa Wiley, 2014.
- [25] G. Bermudez, «Modelamiento cinemático y odométrico de robots móviles: aspectos matemáticos,» 2003.
- [26] V. Balaji, M. Balaji, M. Chandrasekaran, M. Khan y I. Elamvazuthi, Optimization of PID Control for High Speed Line Tracking Robots., 2015, pp. 147-154.