

Sistema de Control con Lógica Difusa y Diseño de Plataforma Autónoma con Ruedas Mecanum a Partir de su Arquitectura

Víctor Costella Batista

Universidad Tecnológica de Panamá
vcostella@yahoo.com

Humberto Rodríguez

Universidad Tecnológica de Panamá
humberto.rodriguez@utp.ac.pa

Resumen- Este artículo presenta la implementación de una plataforma robótica móvil autónoma con ruedas mecanum para estudios en ambientes de carga. Se exploran los posibles movimientos y se eligen los que van a ser estandarizados dentro del sistema. Se propone una arquitectura híbrida novedosa para la interacción de los módulos de hardware y software del sistema. Se utilizan ecuaciones cinemáticas y de lógica difusa a bajo nivel para controlar la velocidad de los motores de las ruedas. Se hacen comparaciones de desempeño del sistema al variar el número de funciones de pertenencia en el programa de control del vehículo. Se analizan los desplazamientos laterales y rotacionales con respecto al centro de la plataforma. Los resultados muestran mejoras en el desempeño del sistema de control del robot y su consecuente fortalecimiento de los procesos de localización de la plataforma robótica.

Palabras claves- Arquitectura Híbrida, Lógica Difusa, Navegación Robótica, Ruedas Mecanum, Sistema de Control.

Abstract- This work presents the implementation of an autonomous robot platform with mecanum wheels for studies in cargo environments. The movements of the system are explored, and some are selected as standards. The proposal includes new hybrid architecture for the interaction of the hardware and software modules of the system. Kinematic equations and low level fuzzy logic are used to control the motor velocities of the wheels. This work also includes a comparison of the performance of the system when varying the number of the membership functions in the control program of the vehicle. Lateral and rotational displacements of the system with respect to the center of the system are analyzed. The results show improvements in the control system of the vehicle and the resultant strengthening of the localization processes of the robotic platform.

Keyword- Hybrid Architecture, Fuzzy Logic, Robotics Navigation, Mecanum Wheels, Power Control.

Tipo de Artículo: original

Fecha de Recepción: 21 de julio 2013

Fecha de Aceptación: 11 de octubre de 2013

1. Introducción

La omnidireccionalidad consiste en la capacidad de un vehículo para trasladarse hacia cualquier dirección y con cualquier configuración [1]. Una plataforma equipada con ruedas mecanum permite movimientos omnidireccionales. Con éstas se logra mayor maniobrabilidad en ambientes congestionados, donde existen obstáculos que impiden la movilidad hacia la meta deseada. En [2] se presenta un estudio reciente sobre las aplicaciones prácticas de vehículos con ruedas mecanum que abarca ambientes de carga o almacenaje (estos espacios son cada vez más reducidos). El uso y estudio de vehículos autónomos con ruedas mecanum en estos y otros tipos de ambientes se está haciendo popular debido a que las maniobras requeridas pueden hacerse engorrosas o simplemente riesgosas para la carga [3]. Las ruedas mecanum proveen el movimiento omnidireccional debido a una serie de rodillos montados en la rueda a 45° y que hacen contacto con el piso. La dinámica de una rueda mecanum hace que se creen fuerzas en los ejes horizontal y vertical, y el conjunto de las fuerzas en las cuatro ruedas hace que una plataforma robótica pueda tener movimientos omnidireccionales y rotacionales sobre su eje central. Su ventaja principal radica en que no se requiere un sistema de dirección [4]. Sin embargo, el control de los movimientos se hace difícil, pues existen factores atenuantes de las velocidades, como el deslizamiento debido al tipo de superficie, el contacto reducido entre el piso y las ruedas, y el efecto de la distribución de la masa de la plataforma robótica [5]. Diversas estrategias han sido abordadas para lograr un control apropiado de robots con ruedas mecanum. En [6,7,8] se ofrecen algunos ejemplos. En todos ellos se hace uso del control cinemático por sobre el dinámico, lo cual consiste en controlar la velocidad de los motores para lograr el movimiento deseado. La estrategia utilizada en este trabajo es la de controlar el robot con ruedas mecanum utilizando su cinemática y no su dinámica. El sistema de control de los motores está basado en el uso de lógica difusa, que ha sido utilizada con éxito en muchas aplicaciones con robots a distintos niveles de software. Algunos ejemplos se ofrecen en [9, 10, 11]. El controlador de motor por

lógica difusa fue propuesto inicialmente por Mamdani y Assilian en 1974 [12]. Este tipo de control es utilizado en problemas de tipo intuitivo con conocimiento experto y tiene la desventaja de que requiere un alto esfuerzo computacional. El método propuesto por Tagaki-Sugeno [13] es computacionalmente más efectivo y es más atractivo en sistemas dinámicos no lineales. En este trabajo se utilizará el método propuesto por Tagaki-Sugeno.

2. Modelo Cinemático

Variando las direcciones y velocidades de las ruedas la plataforma robótica se pueden realizar movimientos omnidireccionales. A pesar de que se logran movimientos al utilizar una, dos, tres y cuatro ruedas, en el trabajo efectuado se consideraron sólo los movimientos que se obtenían al utilizar dos o cuatro ruedas. Estos movimientos fueron estandarizados pues con ellos se pueden lograr las exigencias de desarrollo posteriores como lo son seguir una ruta planificada y evadir los obstáculos en el área de trabajo. Con los cambios en los motores descritos, se logra efectuar los desplazamientos de la figura 1 [14-16].

	Adelante		Derecha
	Atrás		Izquierda
	Sentido del Reloj		Ángulo Derecha
	Sentido Contrario del Reloj		Ángulo Izquierda

Figura 1. Movimientos estandarizados en la plataforma robótica. Se asume que las velocidades de cada uno de los motores son de la misma magnitud [2,4,16].

La distribución de las fuerzas que interfieren debido al empuje de los motores y a la ubicación de los rodillos a 45° hace que la plataforma robótica se desplace en una u otra dirección, o simplemente gire sobre su eje central. Las ecuaciones que definen las velocidades del centro de rotación son las siguientes:

$$\begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{\theta} \end{bmatrix} = \frac{1}{4} \begin{bmatrix} 1 & 1 & 1 & 1 \\ -1 & 1 & 1 & -1 \\ -T & T & -T & T \end{bmatrix} \begin{bmatrix} R_w \dot{\theta}_1 \\ R_w \dot{\theta}_2 \\ R_w \dot{\theta}_3 \\ R_w \dot{\theta}_4 \end{bmatrix} \quad (1)$$

$$T = \frac{1}{L_x + L_y} \quad (2)$$

Donde \dot{x} y \dot{y} son los componentes de la velocidad del centro de la plataforma robótica con respecto al sistema de coordenadas global, y $\dot{\theta}$ es la velocidad angular del centro de la plataforma robótica. R_w es el radio de la rueda y $\dot{\theta}_i$ es la velocidad angular de cada rueda [17]. L_x representa la mitad del ancho de la plataforma robótica a partir de su centro geométrico, y L_y representa la mitad del largo de la plataforma robótica a partir de su centro geométrico.

3. Arquitectura de Software

La base del diseño de un robot autónomo móvil es su arquitectura de software. Sin arquitecturas apropiadas, no es posible crear robots autónomos eficientes. La arquitectura describe las tareas de los bloques funcionales, su organización y su interacción para formar un sistema integrado. Ella está organizada de acuerdo a como se interrelacionan los distintos componentes entre sí. Debido a esto, existen tres tipos de arquitecturas básicas: deliberativas (en las que existe un paso de planificación entre el sensado y la acción), reactivas (en las que simplemente existen acciones luego del sensado), e híbridas (que son una combinación de las arquitecturas deliberativas y reactivas) [18]. Hoy por hoy, las arquitecturas híbridas han demostrado ser las más adecuadas cuando se trata de robots móviles navegando en ambientes de trabajo no estructurados. Diversos autores han propuesto diseños de arquitecturas en ambientes de carga [19-21]. En [22] se ofrece una revisión de las arquitecturas de control para la navegación de robots autónomos.

La arquitectura híbrida propuesta en este trabajo se muestra en la figura 2. En la parte superior de la figura se muestran los cinco módulos principales que componen el sistema. Estos módulos corresponden a la programación de alto nivel. A través de ellos se ejecutan los principales algoritmos del sistema. En este trabajo se designaron los siguientes módulos tomando como

referencia algunas arquitecturas híbridas ya existentes para ambientes de carga:

1. Sensorial: adquiere y almacena la data proveniente de los sensores del robot.
2. SLAM (Localización): define la posición del robot en el ambiente de trabajo de forma confiable.
3. SLAM (Mapeo): crea, almacena, y actualiza la información de los mapas locales y globales (a ser implementado a futuro).
4. Planificación: genera la ruta más eficiente para llegar a la meta.
5. Acción: coordina las actividades de movimiento tomando en cuenta acciones reactivas, evade obstáculos, ejecuta la trayectoria planificada, monitorea el rendimiento del robot y, finalmente, genera el movimiento del robot [23-25].

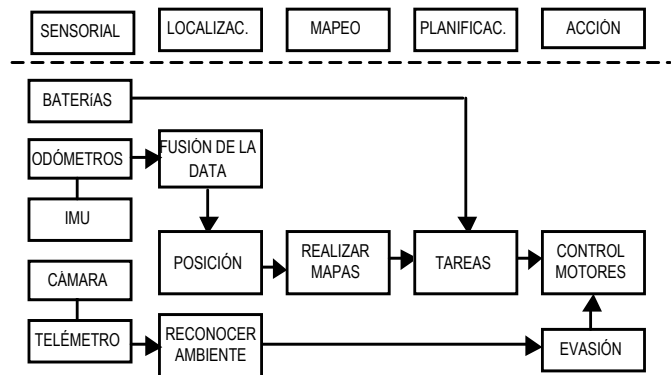


Figura 2. Diagrama de la Arquitectura de software seleccionada para la plataforma robótica.

4. Control de Velocidad de Motores con Lógica Difusa

En la teoría de control clásico, el control de motores se realiza por medio de un controlador proporcional integral derivativo (PID). Sin embargo, si no es modelado apropiadamente, con este controlador el sistema (robot móvil en nuestro caso) puede tener una respuesta inaceptablemente larga ante cambios en la trayectoria deseada. Esto último ocurre cuando el modelo del sistema es difícil de obtener, es parcialmente desconocido, se encuentra ajustado inapropiadamente o es altamente no lineal. Malhotra, Kaur y Singh [26] hacen una comparación de la respuesta de un motor DC utilizando un controlador proporcional integral derivativo versus la respuesta utilizando un controlador

por lógica difusa, mostrando la respuesta superior cuando se usa lógica difusa.

Los controles de lógica difusa han sido utilizados efectivamente en el control de procesos industriales, sistemas mecánicos, sistemas médicos, robots autónomos, y otros. Estos controles son muy útiles cuando los procesos son demasiado complejos para el análisis, o cuando los recursos de información son imprecisos o presentan niveles de incertidumbre. En [27-29] se hace referencias a algunas de las aplicaciones de sistemas por medio de lógica difusa. En el campo de la robótica, el control por lógica difusa es muy popular. Ha sido utilizado en robot móviles con ruedas mecanum para lograr la corrección en la dirección y la búsqueda de la meta, o logro de objetivos. Algunos ejemplos se dan en las referencias [30-32]. Su ventaja es su fácil implementación a través de un conjunto de reglas programables, con las cuales se controla la velocidad de los motores. Su desventaja es la cantidad de sentencias condicionales IF – THEN – ELSE que surgen en la programación y que roban tiempo de ejecución al programa. En la plataforma robótica con ruedas mecanum, la salida del controlador de lógica difusa consiste en una señal que va al controlador de servos y es convertida en pulsos modulados para los manejadores (drivers) de los motores. El controlador por lógica difusa utiliza el error en la velocidad ϵ y la tasa de cambio en el error de la velocidad del motor $\dot{\epsilon}$ para mejorar en la función de salida al controlador.

5. Controlador de Velocidad con Lógica Difusa Basado en Una Arquitectura Híbrida

En este trabajo se plantea una integración de los conceptos de control por lógica difusa propuestos por Tagaki y Sugeno y los conceptos de arquitectura híbrida. El controlador de motores propuesto utiliza lógica difusa y el método de Tagaki y Sugeno (ver figura 3). El error en la velocidad ϵ y el cambio en el error de velocidad $\dot{\epsilon}$ son utilizados como entradas al control difuso, para que, por medio de una serie de reglas establecidas, se varíe la amplitud de los pulsos a los manejadores de los motores DC y así la velocidad de los motores se acerque a la velocidad programada. Los codificadores ofrecen la retroalimentación, y el lazo de control se cierra.

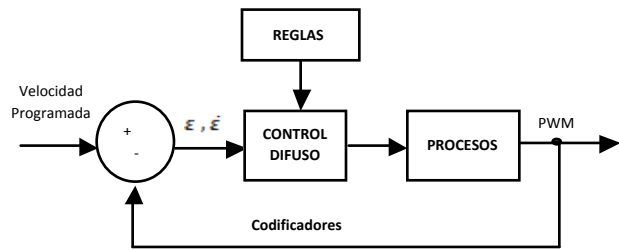


Figura 3. Diagrama del control Difuso utilizado.

La combinación de conceptos hace que el sistema de software sea más coherente y que la programación sea más unificada. En la figura 4 se muestra como se parte del concepto modular de arquitectura híbrida hasta llegar al bajo nivel de lógica difusa y en la figura 5 se muestra plataforma física en donde se implementó, respectivamente. El módulo de planificación determina la ruta dependiendo de la meta deseada y lo que el módulo de localización le presenta hasta que se envían los comandos para que el módulo de acción los ejecute; este módulo se encuentra en el nivel más bajo. El sistema de control de velocidades por lógica difusa, ubicado en el módulo de acción, es retroalimentado por las lecturas de velocidad angular de los codificadores (ODO); pero también los cambios de velocidad dependen de las correcciones en la ruta, que son determinadas por el módulo de planificación a un nivel más alto.

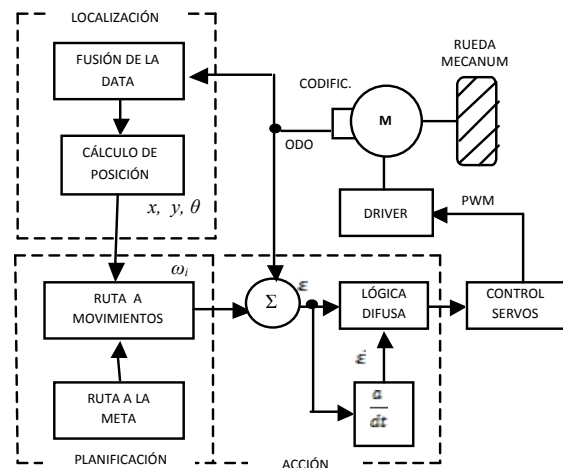


Figura 4. Interacción del control de velocidades del motor por medio de lógica difusa con otros módulos en un sistema con arquitectura híbrida.



Figura 5. Robot Móvil de con ruedas mecanum desarrollado para la implementación de la arquitectura híbrida.

De acuerdo con [8,26], se siguieron cuatro pasos básicos para el diseño del controlador por lógica difusa a partir del de la arquitectura híbrida que se quiere implementar:

1. Definir las variables lingüísticas para entrada y salida.
2. Definir el conjunto de pertenencias de entradas y salidas.
3. Crear las reglas de lógica difusa.
4. Defusificar.

El diseño de un controlador más efectivo estuvo sujeto al número de funciones de pertenencia utilizadas. En el procedimiento se detallan dos modelos: uno con tres y otro con cinco funciones de pertenencia.

Paso 1. Las entradas son los errores (ϵ) en base a las velocidades esperadas. Las salidas difusas modifican las consignas del controlador de servo, que a su vez las transforma en pulsos PWM que serán enviados a los drivers de los motores.

Paso 2. Se definen los conjuntos difusos los cuales contienen las funciones de pertenencia. Las variables lingüísticas de entrada son el error ϵ y el cambio en el error $\dot{\epsilon}$, tomando en cuenta que los datos de retroalimentación serán suministrados por los codificadores de cuadratura en unidades de pulsos/s. El error se define de la siguiente forma:

Las figuras 6 y 7 muestran las funciones de pertenencia para el error en las velocidades y para la tasa de cambio del error en las velocidades respectivamente.

Función de Pertenencia	Error ϵ (pulsos/s)
3 funciones de pertenencia	
NE (negativo)	< 0
CE (cero)	0
PO (positivo)	> 0
5 funciones de pertenencia	
NG (negativo grande)	< -200
NP (negativo pequeño)	$-200 \leq \epsilon < 0$
CE (cero)	0
PP (positivo pequeña)	$0 < \epsilon \leq 200$
PG (positivo grande)	> 200

Figura 6. Funciones de pertenencia para el error en las velocidades.

Función de Pertenencia	Cambio en el error $\dot{\epsilon}$ (pulsos/s)
3 funciones de pertenencia	
NE (negativo)	< 0
CE (cero)	0
PO (positivo)	> 0
5 funciones de pertenencia	
NG (negativo grande)	< -100
NP (negativo pequeño)	$-100 \leq \dot{\epsilon} < 0$
CE (cero)	0
PP (positivo pequeña)	$0 < \dot{\epsilon} \leq 100$
PG (positivo grande)	> 100

Figura 7. Funciones de pertenencia para el cambio en error en las velocidades.

Luego de esto, se especificaron las funciones de pertenencia para la salida (controlador de servos), que se muestran en la figura 8.

Función de Pertenencia	
3 funciones de pertenencia	
AU (aumentar)	
NC (no cambie)	
DI (disminuya)	
5 funciones de pertenencia	
AR (aumentar rápidamente)	
AL (aumentar lentamente)	
NC (no cambie)	
DL (disminuir lentamente)	
ER (disminuir rápidamente)	

Figura 8. Funciones de pertenencia para la salida.

Paso 3. Se definen las reglas de lógica difusa para todas las posibles combinaciones de entrada y salida. En la figura 9 se puede apreciar cómo se define, de forma matricial, el conjunto de reglas de lógica difusa para 3 funciones de pertenencia.

$\begin{matrix} \epsilon \\ \epsilon \end{matrix}$	NE	CE	PO
NE	AU	DI	DI
CE	AU	NC	DI
PO	AU	AU	DI

Figura 9. Conjunto de Reglas de lógica difusa con 3 funciones de pertenencia.

En la figura 10 se define, de forma matricial, el conjunto de reglas de lógica difusa para 5 funciones de pertenencia.

$\begin{matrix} \epsilon \\ \epsilon \end{matrix}$	NG	NP	CE	PP	PG
NG	AR	AR	AL	AL	NC
NP	AR	AL	AL	NC	DL
CE	AL	AL	NC	DL	DL
PP	AL	NC	DL	DL	DR
PG	NC	DL	DL	DR	DR

Figura 10. Conjunto de Reglas de lógica difusa para 5 funciones de pertenencia.

Paso 4. Defusificación

En este trabajo se utiliza el método del promedio ponderado debido a que es muy rápido computacionalmente, y los resultados obtenidos son bastante exactos. En este método la salida es obtenida por medio del promedio ponderado de cada salida del conjunto de reglas definidas en el paso 3. La fórmula general es la siguiente:

$$Z^* = \frac{\sum_{i=1}^n m^i w_i}{\sum_{i=1}^n m^i} \tag{6}$$

Donde Z^* es la salida defusificada, m^i es la pertenencia de la salida de cada regla, y w_i es el peso asociado con cada regla.

6. Resultados y su Discusión

La plataforma robótica funciona con cinco velocidades que se logran por el envío de posiciones al controlador de servos.

En el movimiento hacia adelante y con la velocidad muy lenta, cada motor debe girar aproximadamente a 33.6 rpm. Con esto es necesario obtener velocidades angulares de 2800 pulsos por segundo en el codificador de cuadratura. El radio de las ruedas es de 0.0762 metros y los codificadores tienen una resolución de 5000 pulsos/rev. Utilizando (2) obtenemos un valor T de 2.7240 m⁻¹.

Con esta información, se utiliza (1) para obtener una velocidad de avance de 0.268 m/s. El sistema fue simulado en Matlab, obteniendo las gráficas de superficie que se muestran en las figura 11 y 12 con los ciclos de control utilizando tres y cinco funciones de pertenencia.

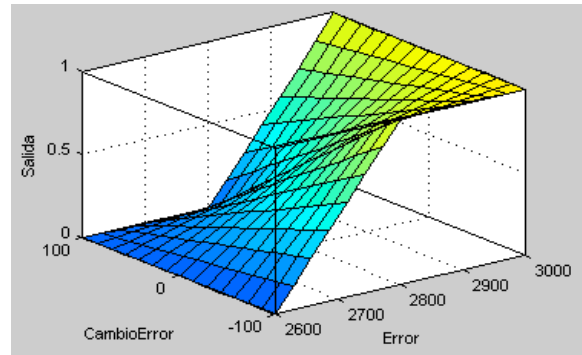


Figura 11. Gráfica de superficie con tres funciones de pertenencia.

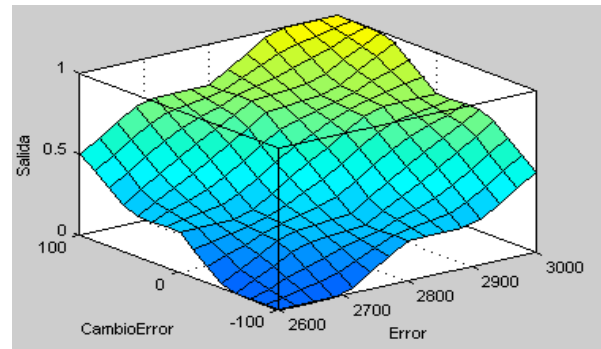


Figura 12. Gráfica de superficie con cinco funciones de pertenencia.

Se realizaron pruebas de 10 segundos con el movimiento hacia adelante, a velocidad muy lenta (33.6 rpm para cada motor del robot). Las lecturas de los codificadores fueron tomadas cada 0.24 segundos aproximadamente.

El objetivo de estas pruebas consistía en determinar si el control de motores por medio de lógica difusa a partir de una arquitectura híbrida producía desviaciones significativas en el rumbo del robot. También con las pruebas se busca determinar si es pertinente utilizar cinco funciones de pertenencia (cuyo coste computacional es más alto) o tres funciones de pertenencia.

Como se puede apreciar en las figuras 13 y 14, cuando se utilizan 3 funciones de pertenencia, la respuesta del controlador de servos es inestable, provocando cambios en el proceso de localización que se detallan posteriormente.

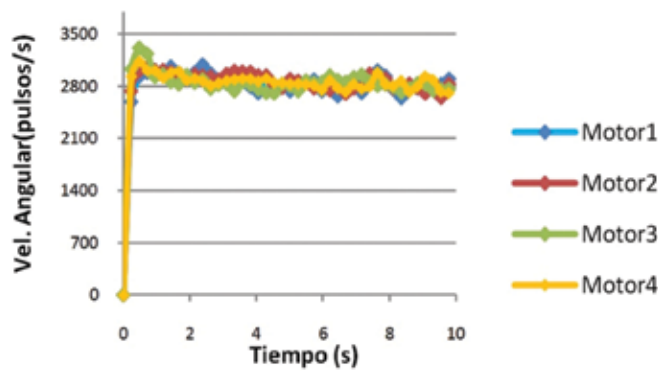


Figura 13. Velocidad de los motores (pulsos/s) cuando se utilizan 3 funciones de pertenencia.

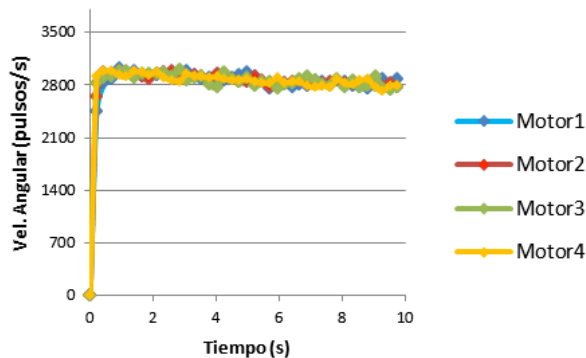


Figura 14. Velocidad de los motores (pulsos/s) cuando se utilizan 5 funciones de pertenencia.

Como muestra, por medio de (1) y (2) se calculó la localización del robot con odometría, utilizando el movimiento hacia adelante. En este movimiento, el avance debe ser lo más recto posible, sin desviaciones laterales ni giros notables con respecto al centro. La

figura 15 muestra el desplazamiento lateral (en cm), utilizando tres funciones de pertenencia. Como se puede notar, la variación al cabo de 10 segundos es menor a 1 cm, con oscilaciones regulares.

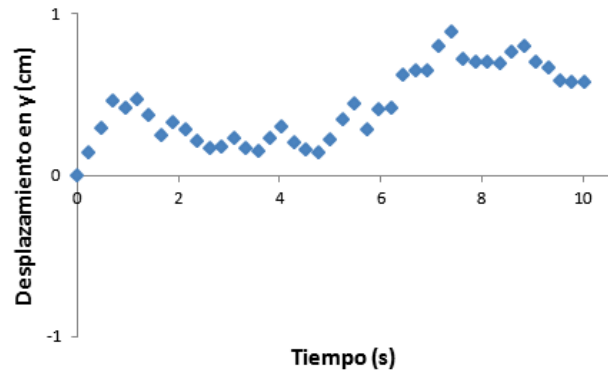


Figura 15. Desplazamiento lateral cuando se utilizan 3 funciones de pertenencia.

Comparativamente, en la figura 16 se muestra el desplazamiento lateral (en cm), utilizando cinco funciones de pertenencia. Se puede apreciar que las variaciones son muy ligeras, menores a los 0.3 cm.

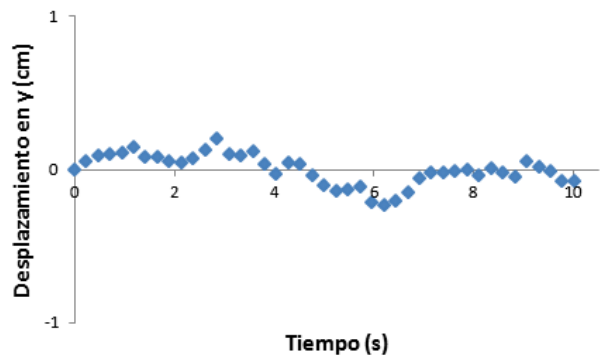


Figura 16. Desplazamiento lateral cuando se utilizan 5 funciones de transferencia.

Las gráficas 17 y 18 muestran la variación del rumbo (en grados) cuando se utilizan 3 y 5 funciones de pertenencia. En el primer caso, se producen desviaciones en el rumbo de hasta dos grados. En el segundo caso la variación en del rumbo es mucho menor, llegando a un máximo de unos 0.5 grados. Esta diferencia significativa en las desviaciones del rumbo es la que nos lleva a elegir,

pese a su coste computacional, al controlador con 5 funciones de pertenencia (25 sentencias condicionales IF-THEN-ELSE) por sobre el controlador con 3 funciones de pertenencia (9 sentencias condicionales IF-THEN-ELSE).

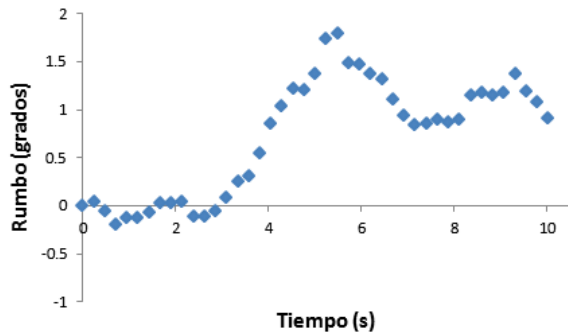


Figura 17. Rumbo (en grados) con 3 funciones de pertenencia.

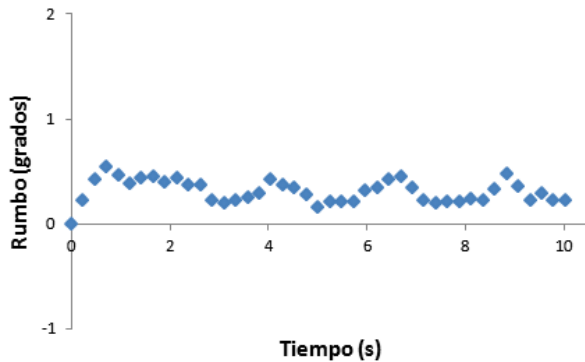


Figura 18. Rumbo (en grados) con 5 funciones de pertenencia.

7. Conclusiones

Este escrito presenta el diseño de un controlador por medio de lógica difusa utilizando el método de Tagaki-Sugeno. Este diseño parte del concepto de arquitectura híbrida, donde la plataforma robótica utilizada deberá ejecutar acciones reactivas/deliberativas con rapidez, determinando con la mayor precisión y exactitud posible su localización en el entorno. El controlador Tagaki-Sugeno es una buena alternativa comparado con el PID tradicional. El sistema de control por lógica difusa con cinco funciones de pertenencia es más robusto que el sistema de control con tres funciones de pertenencia, pero con la particularidad de que la carga computacional se incrementa en un 64% (9 versus 25 sentencias condicionales IF-THEN-ELSE). Corregir el rumbo del

robot no sólo depende de la acción de control de los motores, puesto que los experimentos realizados no incluyen una caracterización de los errores sistemáticos y del ruido que repercuten en el cálculo del rumbo del sistema.

8. Trabajo Futuro

El diseño del controlador de la plataforma robótica a partir del concepto de arquitectura híbrida contribuye a mejoras en el rumbo y posterior proceso de localización del robot. Esto se debe a que asumiendo un modelo de velocidades constantes, que en teoría, se aplica a la mayoría del recorrido, se puede lograr aplicar la técnica de Localización y Mapeo Simultaneo (SLAM) utilizando otros sensores. En una etapa de software de alto nivel, con la aplicación de lógica difusa en la programación del módulo de navegación se pueden corregir las desviaciones en la ruta producidas por errores sistemáticos o por las respuestas del controlador de servos y los motores. Estos ajustes deben ayudar a robustecer la técnica de SLAM.

Referencias Bibliográficas

- [1] Ioan Doroftei et al., "Omnidirectional Mobile Robot - Design and Implementation", Bioinspiration and Robotics: Walking and Climbing Robots, pp. 513, ISBN 978-3-902613-15-8, 2007.
- [2] Florentina Adascalitei and Ioan Doroftei, "Practical Applications for Mobile Robots based on Mecanum Wheels - a Systematic Survey", Proceedings of International Conference On Innovations, Recent Trends and Challenges in Mechatronics, Mechanical Engineering and New High Tech Products Development, MECAHITECH, Vol 3. 2011.
- [3] Haoyong Yu, Steven Dubowsky and Adam Skwersky, "Omni Directional Mobility Using Active Split Offset of Castors", MIT, 2001.
- [4] Jefri Efendi et al., "Designing Omni-Directional Mobile Robot with Mecanum Wheel", American Journal of Applied Sciences 3 (5): 1831-1835, 2006.
- [5] Keiji Nagatani, Satoshi Tachibana, Makoto Sofue and Yutaka Tanaka, "Improvement of Odometry for Omnidirectional Vehicle Using Optical Flow Information", Proceedings of IEEE/RSJ International Conference on Intelligent Robots and Systems, 2000.
- [6] Kyung-Lyong Han, Oh Kyu Choi, In Lee, Inwook Hwang, Jin S. Lee and Seungmoon Choi, "Design and Control of Omni-Directional Mobile Robot for Mobile Haptic Interface", International Conference on Control, Automation and Systems, 2008.
- [7] Mark Killpack, Travis Dayle, Cressel Anderson and Charles Kemp, "Visual Odometry and Control for an Omnidirectional Mobile Robot with a Downward-Facing Camera", Georgia Institute of Technology, 2010.
- [8] Samuel A, Miller, Arthur T, Bradley and Linda L, Bahala, "Network Interfaces and Fuzzy-Logic Control for a Mecanum-Wheeled Omnidirectional Robot", Nasa Langley Research Center, 2005.
- [9] Oscar Castillo et al., "Fuzzy Logic Tracking Control for Unicycle Mobile Robots", Engineering Letters, 13:2, 2006.
- [10] Tang Sai Hong, Danial Nakhaeinia and Babak Karasfi, "Application of Fuzzy Logic in Mobile Robot Navigation", Elmer Dadios (Ed), Intech, 2012.
- [11] Viorel Stoian and Mircea Ivanescu, "Robot Control by Fuzzy Logic", Frontiers in Robotics, Automation and Control, Alexander Zemliak (Ed), Intech, 2008.

- [12] E. H. Mandani and S. Assilian, "An experiment in Linguistic Synthesis of Fuzzy Controllers", *International Journal of Man-Machine Studies*, 1975.
- [13] Tomohiro Tagaki and Michio Sugeno, "Fuzzy Identification of Systems and Its Applications to Modeling and Control, *IEEE Transactions on Systems, Man and Cybernetics*, Vol. SCM-15, 1985.
- [14] Nkgatho Tlale and Mark de Villiers, "Kinematics and Dynamics Modeling of a Mecanum Wheeled Mobile Platform", *Council for Scientific and Industrial Research*, 2006.
- [15] Olaf Diegel et al., "Improved Mecanum Wheels Design for Omni Directional Robots", *Australian Conference on Robotics and Automation*, Auckland, 2002.
- [16] Nkgatho Sylvester Tlale, "On distributed Mechatronics Controller for Omni-Directional Autonomous Guided Vehicles", *OpenUP*, 2007.
- [17] Jungmin Kim, Jungje Park and Sungshin Kim, "Inertial Navigation System for Omni-directional AGV with Mecanum Wheel", *Advances in Mechanical Engineering*, Volume 2, Number 1, 2012.
- [18] Valdir Grassi Junior, Sarangi P, Parikh and Jun Okamoto Junior, "Hybrid Deliberative/Reactive Architecture for Human-Robot Interaction", *ABCMS Symposium Series in Mechatronics*, Vol. 2, pp. 563-570, 2006.
- [19] Tua Tamba et al., "Time-Varying Feedback Control of an Unmanned Autonomous Industrial Forklift", *Proceedings of the 17th World Congress, The International Federation of Automatic Control*, Seoul, Korea, July, 2008.
- [20] Ashley Tews, "Safe and Dependable Operation of a Large Industrial Autonomous Forklift", 2009.
- [21] Najdan Vuković and Zoran Miljković, "New Hybrid Control Architecture for Intelligent Mobile Robot Navigation in a Manufacturing Environment", *FME Transactions*, 2009.
- [22] D. Nakhaeinia, et al., "A Review of Control Architectures for Autonomous Navigation of Mobile Robots", *International Journal of the Physics Sciences*, Vol. 6(2), pp. 169-174, 2011.
- [23] Hugo Silva and Alberto Álvares, "Development of a Hybrid Architecture for the autonomous control of Mobile Robots, *ABCMS Symposium Series in Mechatronics* , Vol. 4 , pp. 749-757, 2010.
- [24] John Leonard et al., "A Perception-Driven Autonomous Urban Vehicle", *Journal of Field Robotics* 25(10), pp. 727-774 , 2008.
- [25] Seth Teller Matthew et al., "A Voice-Commandable Robotic Forklift Working Alongside Humans in Minimally-Prepared Outdoor Environments", *Air Force under Air Force Contract FA8721-05-C-0002*, 2010.
- [26] Rahul Malhotra, Tejbeer Kaur y Gurpreet Singh Deol, "DC Motor Control Using Fuzzy Logic Controller", *International Journal of Advanced Engineering Sciences and Technologies*, Vol. 8, Issue No, 2, 291-296.
- [27] Dorota Jankowska, Anna Justina Milewska and Urszula Gorska, "Applications of Logic in Medicine, *Studies in Logic, Grammar, and Rethoric*, 2010.
- [28] N.O. Rubens, "The application of Fuzzy Logic To the Construction of The Ranking Function Of The Information Retrieval Systems", *Computer Modeling and New Technologies*, Vol 10, pp. 20-27, 2006.
- [29] Abdel Ghano Aissaoui and Ahmed Tahour, "Applications of Fuzzy Logic Control of Electrical Machines", *Fuzzy Logic-Controls, Concepts, Theories and Application*, Elmer Dadios (Ed), Intech, 2012.
- [30] Rasif Rashid et al., "Differential Drive Wheeled Mobile Robot (WMR) Control Using Fuzzy Logic Techniques", *Asia International Conference on Mathematical/Analytical Modelling and Computer Simulation*, 2010.
- [31] Tzafestas S,G, and Zavlangas P, "Industrial and Mobile Collision-Free Motion Planning Using Fuzzy Logic Algorithms", *Industrial Robotics: Theory, Modeling and Control*, 2006.
- [32] S.K. Harisha, Ramkanth Kumar P, M, Krishna, and S.C. Sharma, "Fuzzy Logic Reasoning to Control Mobile Robot on Predefined Strip Path", *World Academy of Science, Engineering and Technology*, 2008.