

# Development and simulation of the Avoidance and navigation using mobile robots

Heyson Baez, Magister en Ingenieria<sup>1</sup>, Edgar Aguirre B, Magister en Ciencias de la Información y las Comunicaciones<sup>2</sup>, y Sergio González G, Magister en Administración de Empresas<sup>3</sup>

<sup>1</sup>Pontificia Universidad Javeriana, Universidad Distrital Bogotá, Colombia, Universidad Antonio Nariño, Bogotá, Colombia,

<sup>1</sup>heyson.baez@uniminuto.edu.co, <sup>2</sup> eaguirreb@uniminuto.edu.co, <sup>3</sup> segonzalez@uniminuto.edu.co

**Abstract—** *In this work, we present a development and simulation of control using by robot in navigation which using software tools for engineering and fundamental in Matlab software and tools such as Simulink, Animation 3D y Fuzzy Logic. In this development we propose by the navigation and the evasion of the objects in the robotics cooperatives platforms and industrial of this environment dynamics and statics. Been successfully the avoidance of objects and generating the route to reach the proposed goal. The Implemented of the rules with statics fuzzy logic software with Matlab, was simulated environment using statics and dynamics devices with the 3D Animation toolbox of Matlab, and finally, was interconnected with the simulation model (3D Animation) with the platform and fuzzy rules (Fuzzy Logic) through the Matlab Simulink toolbox. Finally, these favorable results are obtaining in the evasion and navigation in static and dynamics environments, equal to speed velocity control and in the trolley angle with fuzzy logic. These results are obtaining theoretical data close to the reality for the implementation in a real model as it is in the robotics platform played football using small devices or in an industrial, as one can see the result in a web video.*

**Keywords—**Navigation, Software, Tools, Control, Robotics

Digital Object Identifier (DOI):<http://dx.doi.org/10.18687/LACCEI2018.1.1.124>

ISBN: 978-0-9993443-1-6

ISSN: 2414-6390

# Desarrollo y Simulación de la Evasión y Navegación de Robots Móviles Utilizando Toolboxes de Matlab y Lógica Difusa

Heyson Baez, Magister en Ingeniería<sup>1</sup>, Edgar Aguirre B, Magister en Ciencias de la Información y las Comunicaciones<sup>2</sup>, y Sergio González G, Magister en Administración de Empresas<sup>3</sup>

<sup>1</sup>Pontificia Universidad Javeriana, Universidad Distrital Bogotá, Colombia, Universidad Antonio Nariño, Bogotá, Colombia,

<sup>1</sup>[heyson.baez@uniminuto.edu.co](mailto:heyson.baez@uniminuto.edu.co), <sup>2</sup>[eguirreb@uniminuto.edu.co](mailto:eguirreb@uniminuto.edu.co), <sup>3</sup>[segonzalez@uniminuto.edu.co](mailto:segonzalez@uniminuto.edu.co)

**Abstract**– En este trabajo, se presenta el desarrollo y simulación de un control difuso para la navegación de robots móviles utilizando herramientas de software para ingeniería y que se fundamenta en el software de MATLAB junto con Simulink, Animation 3D y Fuzzy Logic toolboxes. Teniendo en cuenta la búsqueda de que una plataforma robótica pueda navegar en plataformas de tipo cooperativas e industriales tanto en entornos estáticos y dinámicos. Logrando la evasión de obstáculos y generando una ruta para llegar a la meta propuesta. Para su implementación se desarrollaron reglas difusas por medio del software de Fuzzy Logic de MATLAB donde se implementó un entorno de simulación con obstáculos estáticos y dinámicos con el toolbox de Animation 3D de MATLAB y finalmente, se interconectó el modelo de simulación (Animation 3D) con la plataforma y sus reglas difusas (Fuzzy Logic) por medio del toolbox de Simulink de MATLAB. Obteniendo resultados favorables en la evasión y navegación en los dos entornos propuestos, como también en el control de velocidad y en el control del ángulo de giro obtenidas con la lógica difusa. Estos resultados buscan obtener datos teóricos cercanos a la realidad para la implementación en un modelo real como es la implementación en una plataforma robótica que juega fútbol Small Sise o en una plataforma móvil industrial para entornos dinámicos, como también se puede apreciar uno de los resultados en un video en la Web en [1].

**Keywords**–Navegación, Software de Ingeniería, Herramientas Computacionales, Lógica difusa Control, Robótica.

## I. INTRODUCCIÓN

La lógica difusa se está proyectando en múltiples enfoques, tanto del control inteligente como en el diseño de métodos complejos al no tener modelos precisos y exactos. Por otra parte, la navegación de una plataforma robótica es un proceso complejo y que se requiere aplicar en un entorno dinámico con objetos en movimiento o con la interacción con el ser humano, tales como, salas de museos, edificios de oficinas, escuelas, aeropuertos, centros comerciales, industrias y hospitales [2]. Por lo tanto, es necesario tener herramientas computacionales aplicables a procesos de software y hardware que determinen un comportamiento cercano al real para su desarrollo e implementación en los modelos reales, con lo anterior, la robótica móvil tiene la necesidad de desarrollar procesos seguros para la interacción con los seres humanos de manera que la plataforma robótica pueda responder a una lista de situaciones complejas o cooperativas. Esto incluye que el robot pueda situarse y navegar en un entorno desconocido, no estructurado y dinámico.

Por lo tanto, en este trabajo se presenta la implementación y simulación de un método de evasión, planificación de rutas y movimientos que debe realizar la plataforma robótica para navegar en ambientes estructurados utilizando herramientas de software de simulación que se asemejan a un entorno real. Esta planificación de movimientos es fundamental para el comportamiento y funcionamiento del robot ante un ambiente con obstáculos estáticos y dinámicos. Además, un problema fundamental en la navegación autónoma es la necesidad de interactuar con múltiples incertidumbres, desde la información sensorial recibida, como también interactuar con información incompleta y aproximada. Con lo anterior, la lógica difusa se proyecta de forma novedosa en los sistemas de control con enfoques en la inteligencia computacional y en la toma de decisión humana. Con lo anterior, la lógica difusa requiere de un software computacional que soporte la información externa del robot basada en la información de los sensores determinando las acciones que la plataforma robótica deberá realizar.

El objetivo principal es desarrollar un método de navegación y elección de rutas desde un punto inicial a una meta requerida utilizando herramientas de software de simulación como es el toolbox de MATLAB y esta investigación se desarrollo con la versión de MATLAB R2014a; generando un entorno simulado con múltiples obstáculos y elementos con movimiento dinámico, en donde, la plataforma robótica debió tener una ruta para evitar colisiones y así determine autónomamente las acciones a desarrollar para llegar a la meta. Cabe notar, que el entorno de trabajo de la plataforma robótica está en ambientes dinámicos e impredecibles.

Por otra parte, se debe tener en cuenta que el robot se complementa por múltiples sistemas de sensado y de control especializados que determinan los objetos u obstáculos presentes en el entorno en una distancia específica, en donde, el sistema de control recibe continuamente datos de sensado sobre los eventos que ocurren alrededor de la plataforma, determinando las acciones y comportamientos de evasión para lograr trasladarse en su entorno simulado por medio de la herramienta de software de Fuzzy Logic, Simulink y Animation 3D de MATLAB R2014a que determinan inicialmente los resultados preliminares y las correcciones a implementar. Además, las simulaciones a desarrollarse con las herramientas

Digital Object Identifier (DOI): <http://dx.doi.org/10.18687/LACCEI2018.1.1.124>

ISBN: 978-0-9993443-1-6

ISSN: 2414-6390

de software varían en su resultado en comparación con una implementación real debido al hardware de los dispositivos de sensado reales y su implementación adecuada.

El control difuso puede ser mejorado o adaptado para los posibles problemas de hardware existentes en su implementación, finalmente, para el sistema de control se utilizó lógica difusa que solucionó los múltiples problemas de incertidumbre y de la ambigüedad de la información que recibe el sistema.

Con lo anterior, este artículo se divide en cuatro etapas, la primera etapa expone el estudio de trabajos relacionados, la segunda etapa es el desarrollo del control difuso con Fuzzy Logic de *MATLAB* con la versión R2014a, la tercera etapa es la implementación de un modelo virtual con múltiples herramientas de software con Simulink y Animation y la cuarta etapa se fundamenta en los resultados de diferentes pruebas realizadas ante múltiples obstáculos estáticos y dinámicos por medio de la plataforma Matlab.

## II. TRABAJOS RELACIONADOS

Como punto de partida se expondrán algunos trabajos orientados al control difuso utilizado para controlar la navegación de los robots, evitando la colisión o presencia de un obstáculo a lo largo de su camino hacia el punto de destino, logrando llegar a la meta requerida por una plataforma robótica. Así, estos temas expuestos se pueden considerar como un punto de partida para el desarrollo del proyecto.

Inicialmente, una de las primeras propuestas desarrolladas con control difuso y toma de decisión en un robot móvil es el que realizó Gerke [3], en donde, Gerke expone un concepto avanzado para resolver el problema de la evasión de colisiones entre los multi-agentes, manteniendo las condiciones de tiempo real para los controladores de los robots industriales así como en [4]. Además, se realiza una descripción de los conjuntos difusos utilizados para eventos críticos entre los robots en un espacio de trabajo común. Finalmente se expone un método de planificación de trayectorias y evasión de obstáculos que controla a una plataforma robótica, generando las acciones de control ante cualquier obstáculo estático o dinámico que se pueden presentar en un ambiente de trabajo.

Por otra parte, en la robótica cooperativa Elmer [4], presenta una técnica de control difuso en la navegación con obstáculos y evasión de colisiones entre robots cooperativos que juegan fútbol. En donde, se tienen en cuenta múltiples posiciones dinámicas de los robots y de los obstáculos. Para el desarrollo de este método se requiere información necesaria sobre el entorno de los robots su punto de destino, la distancia entre robots y los obstáculos obtenidos en el sistema de visión de la cámara. Concluyendo con unos resultados del control más preciso y robusto en comparación con el enfoque con controladores convencionales. Takuya Kamano, entre otros

investigadores [5], presentan el uso de los conocimientos de cooperación para la captura de un problema con múltiples robots móviles. Para lograr la captura exitosa de un objetivo de escapar, los robots tienen que moverse hacia el blanco después de escapar de los robots conducir el blanco en la zona de captura formado por sí mismos. A continuación, el conocimiento cooperativo se representa como las reglas de formación difusos, y se utiliza para llegar a un objetivo. Los resultados experimentales han demostrado que el control de cooperación propuesto es efectivo para lograr llegar al objetivo.

Haber [6], demuestra la influencia positiva que han tenido las aplicaciones de la lógica y el control difuso en el ámbito productivo, especialmente frente a los procesos complejos y que, por tanto, son objeto propicio para la introducción del llamado control inteligente. Para Suraman, entre otros [7], describe el método para codificar los conjuntos difusos y el procedimiento para llevar a cabo inferencias difusas en la conducta de navegación de robots. En este documento, la toma de decisiones utiliza la lógica difusa para la coordinación, lo que proporciona una transición suave entre las conductas con una respuesta de salida sin problemas consiguientes. Algunas de los resultados experimentales se muestran también para la evasión de obstáculos, como de otras conductas de seguir paralelo a una pared y buscar la meta.

Entre los trabajos desarrollados para la robótica fija se encuentra Amer [8], que expone sobre el control de un robot industrial no lineal, con la incertidumbre y las perturbaciones externas que deben ser consideradas en el diseño de las leyes de control. Además, un control difuso es una buena estrategia para el movimiento de robots fijos garantizando la estabilidad asintótica global. Chih-Lyang Hwang y Shau-Fu Chao [9], también describen la aplicación de lógica difusa en un brazo robótico. Los modelos de referencia se basan en las respuestas sinusoidales del brazo del robot en función de los métodos desarrollados por Takagi Sugeno [10], por otra parte, los trabajos desarrollados en robótica móvil en [11] [12] se presenta los resultados experimentales relacionados con el diseño de un robot móvil con una forma diferencial, el cual tiene como misión seguir rutas programadas con el fin de comprobar la efectividad del diseño y de la plataforma en tres rutas diferentes. Finalmente, los experimentos de brazo de robot de dos articulaciones en el plano horizontal con (o sin) carga útil se llevaron a cabo para verificar la utilidad del control propuesto de Tzafestas S.G. [14], en donde él expone en su documento que la planificación de movimiento es una tarea primordial en el funcionamiento del robot, donde el objetivo es determinar caminos libres de colisiones para un robot que trabaja en un entorno que contiene algunos de los obstáculos en movimiento. Amur [2], presenta la importancia de un desarrollo de navegación ante un entorno dinámico.

Un problema importante en la navegación autónoma es la necesidad de tratar con la gran cantidad de incertidumbres de

la información sensorial recibida por el robot que es incompleta y aproximada, así como con el hecho de que el entorno en el que estos robots operan son entornos dinámicos y contienen elementos de variabilidad. Un enfoque para solucionar estos inconvenientes de la navegación es la lógica difusa basada en el comportamiento haciendo frente a la incertidumbre y la ambigüedad de la información que recibe el sistema. y con lo cual, es sumamente necesario realizar pruebas simuladas para determinar posibles comportamientos que no sean deseados en un modelo real y que pueden producir pérdidas económicas o daños de las plataformas robóticas o de daños en los equipos de desarrollo.

### III. CASO DE ESTUDIO

Este proyecto de estudio busca un método de planificación de rutas y movimientos que debio realizar la plataforma robótica para navegar en ambientes estructurados. Esta planificación de movimientos es fundamental para el comportamiento y funcionamiento del robot ante un ambiente con obstáculos estáticos y dinámicos expuestos en la figura 1.

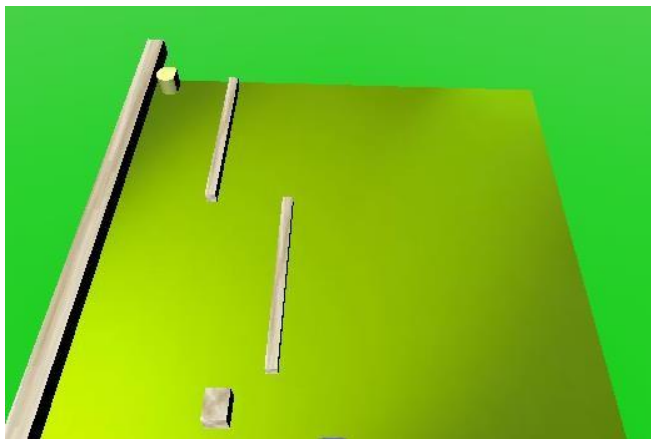


Figura 1. Ambiente estructurado con obstáculos estáticos y dinámicos.

Además, para la planificación de rutas y evasión de obstáculos se debe tener en cuenta la interacción con múltiples incertidumbres, por factores sensoriales como retardos o errores de la información recibida. Con lo anterior, como método de solución se elige la utilización de la lógica difusa para realizar las tareas requeridas de forma confiable y funcional. En donde, la teoría de la lógica difusa es inspirada en la capacidad humana de razonar con información basada en la percepción del entorno de trabajo.

Finalmente, para demostrar los resultados y validar el modelo implementado se realizo un modelo virtual simulado con el software MATLAB con lo cual, se pueda determinar las respuestas funcionales de la plataforma robótica y los posibles errores en su implementación.

### IV. PLANTEAMIENTO DE LA PROPUESTA

La propuesta que se presenta, se basa en realizar una secuencia de reglas para un control difuso que represente los posibles casos que se le pueden presentar a una plataforma robótica ante la necesidad de navegar en un entorno de trabajo con obstáculos estáticos y dinámicos. En donde, en la figura 2 la plataforma robótica tiene un sistema de sensado especializado por medio de una cámara superior que informa los cambios y obstáculos presentes en el entorno.

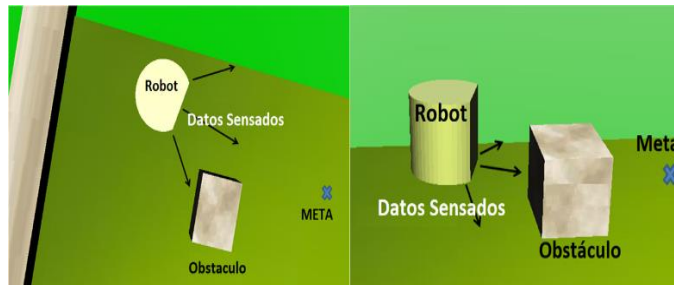


Figura 2. Plataforma robótica con sus datos de sensado y entorno de trabajo, vistas superior (Izq.) y vista lateral (der).

Por otra parte, la complejidad dinámica del entorno, los retardos e imprecisiones de la información recibida generan dificultades para el sistema de control y la toma de decisiones en la navegación. Con lo anterior, se decide aplicar un control difuso que puede realizar acciones de control en un entorno complejo. Además, la lógica difusa proporciona una metodología de reglas lingüísticas condicionales que generan un razonamiento y elección de decisiones acordes a lo esperado. Estas decisiones se basan en la información cercana obtenida y que tienen un grado de incertidumbre e imprecisión. Al aplicar un control difuso que realice las acciones de control en un entorno complejo y determine la forma de evitar los obstáculos, se debe tener en cuenta que por cada parámetro que el robot requiere para sus acciones deben existir unas reglas lingüísticas difusas. En este caso, la plataforma robótica requiere un control difuso de velocidad, un control difuso de movimiento angular y un control difuso de evasión de obstáculos.

A continuación, en la figura 3 se muestra un diagrama de bloques de la conexión de cada uno de los controles definidos en función de la navegación de la plataforma robótica; con lo anterior, cabe destacar que al tener tres sistemas de control para la plataforma robótica se pueden presentar posibles conflictos de recursos o de control, por lo tanto, se realiza un sistema de selección de prioridad de controles para la plataforma, el cual, dependiendo de los eventos presentes o futuros percibidos selecciona que sistema de control es el que tiene el dominio de la plataforma.

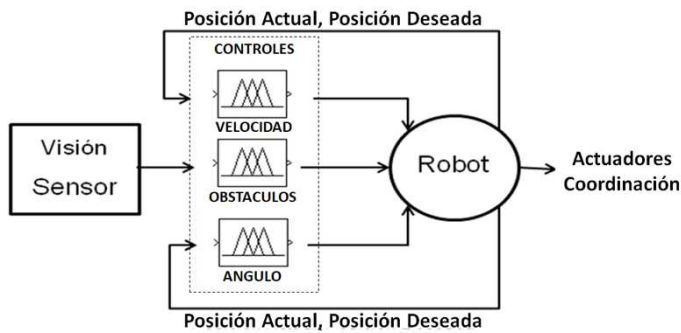


Figura 3. Diagrama de bloques de la conexión de cada uno de los controles definidos para la navegación de la plataforma robótica.

Para el control de velocidad y el control del movimiento angular de la plataforma; se debe considerar que en una capa superior de procesamiento y sensado, la plataforma puede **determinar su posición actual y su posición angular** ante una referencia específica. Luego de determinar su posición actual y angular el robot debe conocer la posición deseada a la que debe llegar. Por lo tanto, en el control difuso se determina como salida de control al error angular que tiene el robot, que se analiza por medio de la posición deseada y la distancia entre la posición actual y la posición deseada. Además, el sistema de control determina la velocidad y el movimiento angular que debe desarrollar la plataforma robótica para alcanzar el objetivo.

El control de **evasión de obstáculos** recibe la información del sistema de sensado y de la distancia a la que se encuentra el **robot al obstáculo** presente. Para concluir que entre más datos recibidos obtiene la plataforma robótica más condiciones lingüísticas requiere el control difuso. Por medio de Robert Babuska [15] se muestra un diagrama de bloques del diagrama del control difuso expuesto en la Figura 4 con fusificación y defusificación unido a un filtro dinámico externo. Además, se procesan tres pasos importantes: discretización, la inferencia y Defusificación.

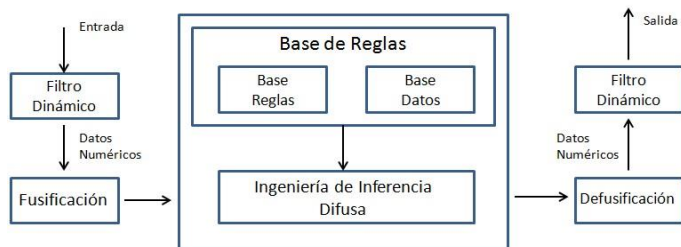


Figura 4. Diagrama del control de velocidad por medio del software Matlab.

Para este trabajo no se explicará el proceso de la lógica difusa, si se desea profundizar en estos temas se pueden referir a [15]. Cabe notar, que se pueden agregar casos especiales presentes en el entorno de trabajo de la plataforma robótica. En algunos artículos citados anteriormente se proponen otros parámetros como son: obstáculos emergentes que cierran toda

el área de movimiento u obstáculos continuos como paredes o de una forma geométrica específica.

Para realizar el modelo de control difuso, a continuación, se presentarán las correspondientes tablas lingüísticas desarrolladas para el funcionamiento del control difuso de este proyecto.

## V. IMPLEMENTACIÓN

En esta sección se expondrán la implementación desarrollada para la navegación y evasión de obstáculos de la plataforma, en donde, se exponen las reglas lingüísticas difusas implementadas en el toolbox de Fuzzy Logic de *MATLAB* R2014a y la implementación del entorno de simulación con Animation 3D y Simulink de Matlab.

### A. Reglas lingüísticas

Inicialmente, utilizando el toolbox de *MATLAB* de Fuzzy Logic se desarrolla la representación de las reglas lingüísticas para el control difuso de velocidad ver figura 6, además, este control de velocidad depende de dos variables de entrada como son: el error angular (orientación) y la distancia a la meta de la plataforma robótica las cuales, generan una variación difusa del control de velocidad.

En la figura 8 se exponen las reglas lingüísticas para el sentido del giro, en donde son requeridas como entradas difusas de la plataforma al error angular y a la distancia a la meta, con estas variables difusas y sus posibles combinaciones se obtiene el control de sentido de giro difuso. Finalmente, se presenta en la figura 10, las reglas a seguir en el sistema de evasión de obstáculos de la plataforma, además, se presentan en la evasión de obstáculos las múltiples entradas de los sensores de la plataforma, como son los sensores izquierdos, frontal izquierdo, frontal, frontal derecho y derecho, que generan la correspondiente salida de velocidad y ángulo de giro difuso.

Cada una de estas reglas implementadas se programaron e implementaron en el Toolbox de Fuzzy Logic de *MATLAB* con sus entradas y la salida de cada una de las variables a controlar y que a continuación se explicarán las generalidades de la selección de los controles difusos.

### B. Implementación de las Reglas Lingüísticas

En este caso, los controles difusos son seleccionados por los requerimientos de la plataforma robótica, tanto para su desplazamiento (control de velocidad y control de ángulo o giro), como para la evasión de obstáculos (control de evasión). Además, las reglas difusas se asignan fundamentadas en los posibles sensores exteroceptivos que tendría el robot en el modo real. Con lo anterior, En la figura 5, 6 se expone el modelo difuso aplicado por medio del software de Fuzzy Logic de *MATLAB* y que utiliza una inferencia difusa Mamdani y su defusificación es por medio de un centroide.



Para el **sistema de control de velocidad** en el diagrama presentado se encuentra expuesto en la figura 5, donde, se puede observar las dos entradas requeridas (Error Angular en rad/s y Distancia a la Meta en metros) y la obtención de su salida (Velocidad en m/s) este modelo presentado está desarrollado por medio del software *MATLAB* R2014a. Además, se observa el método difuso utilizado como es la defusificación (Centroide) e inferencia (Mamdani), estos métodos difusos son seleccionados por su proximidad a la salida esperada, es decir su respuesta de control es aproximada a la que se espera como respuesta del robot.

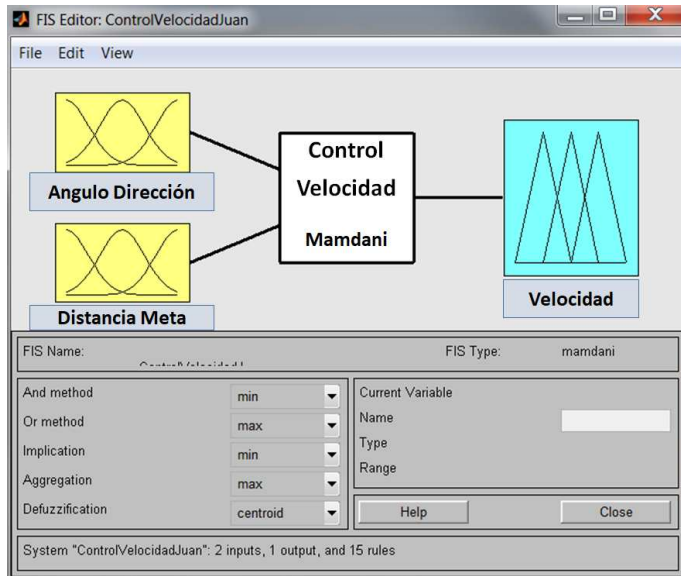


Figura 5. Sistema genérico con fusificación y defusificación unido a un filtro dinámico externo.

Cabe notar, que en la figura 6 el control de velocidad tiene un conjunto de condiciones lingüísticas, las cuales, pueden variar según la exactitud y la afinación que se requiera para el modelo de control implementado. En el caso del control de velocidad se implementaron **15 reglas difusas**.

1. If (ErrorAngularDireccion is SN) and (DistanciaMeta is N) then (Velocidad is Z) (1)
2. If (ErrorAngularDireccion is SN) and (DistanciaMeta is S) then (Velocidad is P) (1)
3. If (ErrorAngularDireccion is SN) and (DistanciaMeta is B) then (Velocidad is P) (1)
4. If (ErrorAngularDireccion is Z) and (DistanciaMeta is N) then (Velocidad is Z) (1)
5. If (ErrorAngularDireccion is Z) and (DistanciaMeta is S) then (Velocidad is P) (1)
6. If (ErrorAngularDireccion is Z) and (DistanciaMeta is B) then (Velocidad is P) (1)
7. If (ErrorAngularDireccion is N) and (DistanciaMeta is N) then (Velocidad is Z) (1)
8. If (ErrorAngularDireccion is N) and (DistanciaMeta is S) then (Velocidad is SP) (1)
9. If (ErrorAngularDireccion is N) and (DistanciaMeta is B) then (Velocidad is SP) (1)
10. If (ErrorAngularDireccion is SP) and (DistanciaMeta is N) then (Velocidad is Z) (1)
11. If (ErrorAngularDireccion is SP) and (DistanciaMeta is S) then (Velocidad is SP) (1)
12. If (ErrorAngularDireccion is SP) and (DistanciaMeta is B) then (Velocidad is P) (1)
13. If (ErrorAngularDireccion is P) and (DistanciaMeta is N) then (Velocidad is Z) (1)
14. If (ErrorAngularDireccion is P) and (DistanciaMeta is S) then (Velocidad is SP) (1)
15. If (ErrorAngularDireccion is P) and (DistanciaMeta is B) then (Velocidad is SP) (1)

Figura 6. Reglas difusas del control de velocidad.

Para el **sistema de control angular** en el diagrama presentado se presenta en la figura 7 donde, se tienen dos entradas que son

el Error Angular y Distancia a la Meta; la obtención de su salida es el control del Angulo requerido. Por otra parte, se observa el método difuso utilizado, con defusificación por medio de un centroide e inferencia (Mamdani), estos métodos difusos son seleccionados por su proximidad a la salida esperada, es decir su respuesta de control es aproximada a la que se espera como respuesta del robot.

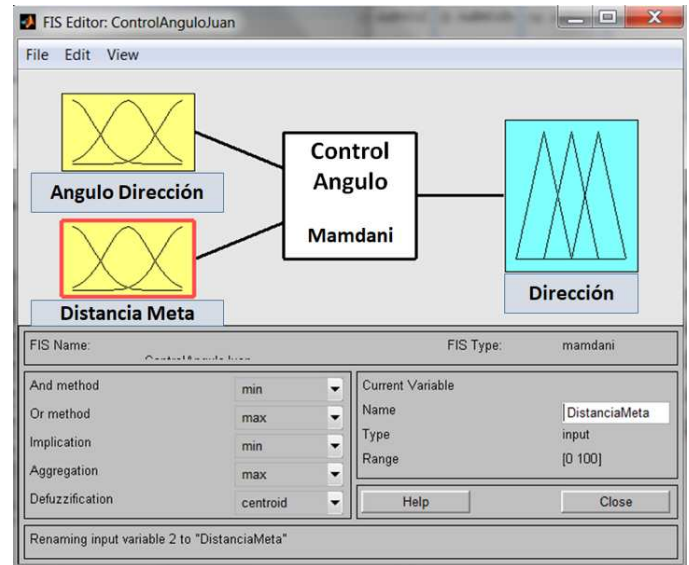


Figura 7. Diagrama del control angular por medio del software Matlab

Además, en la figura 8 el control Angular tiene un conjunto de condiciones lingüísticas, que pueden variar según la exactitud y la afinación que se requiera para el modelo de control implementado. En el caso del control Angular se implementaron **15 reglas difusas**.

1. If (ErrorAngularDireccion is Z) and (DistanciaMeta is N) then (Direccion is F) (1)
2. If (ErrorAngularDireccion is Z) and (DistanciaMeta is S) then (Direccion is F) (1)
3. If (ErrorAngularDireccion is Z) and (DistanciaMeta is B) then (Direccion is F) (1)
4. If (ErrorAngularDireccion is SN) and (DistanciaMeta is N) then (Direccion is RF) (1)
5. If (ErrorAngularDireccion is SN) and (DistanciaMeta is S) then (Direccion is RF) (1)
6. If (ErrorAngularDireccion is SN) and (DistanciaMeta is B) then (Direccion is RF) (1)
7. If (ErrorAngularDireccion is N) and (DistanciaMeta is N) then (Direccion is R) (1)
8. If (ErrorAngularDireccion is N) and (DistanciaMeta is S) then (Direccion is R) (1)
9. If (ErrorAngularDireccion is N) and (DistanciaMeta is B) then (Direccion is R) (1)
10. If (ErrorAngularDireccion is SP) and (DistanciaMeta is N) then (Direccion is LF) (1)
11. If (ErrorAngularDireccion is SP) and (DistanciaMeta is S) then (Direccion is LF) (1)
12. If (ErrorAngularDireccion is SP) and (DistanciaMeta is B) then (Direccion is LF) (1)
13. If (ErrorAngularDireccion is P) and (DistanciaMeta is N) then (Direccion is L) (1)
14. If (ErrorAngularDireccion is P) and (DistanciaMeta is S) then (Direccion is L) (1)
15. If (ErrorAngularDireccion is P) and (DistanciaMeta is B) then (Direccion is L) (1)

Figura 8. Reglas difusas del control angular o sentido del giro.

Para el sistema de control en la figura 9 se presenta la **evasión de obstáculos** con sus variables de entrada y las variables de salida. En donde, se observa **las múltiples entradas** presentes como son los sensores de la visión (sensor izquierdo, frontal izquierdo, frontal, frontal derecho y derecho), estas entradas pueden ser en cantidad menores siempre que la exactitud de la plataforma robótica sea inferior. Además, este modelo de control presenta un sistema MIMO con múltiples

entradas y múltiples salidas que, en este caso de control de velocidad, de evasión, y del ángulo requeridos para la evasión de los obstáculos de la plataforma robótica. Además, se observa el método difuso utilizado como es la defusificación (Centroide) e inferencia utilizando (Mamdani).

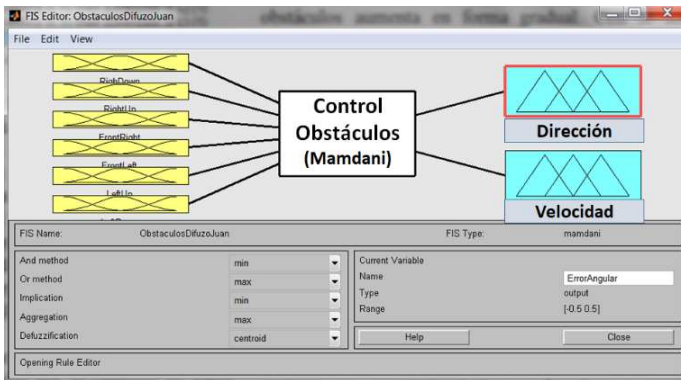


Figura 9. Sistema genérico con fusificación y defusificación unido a un filtro dinámico externo.

Finalmente, en la figura 10 el control de evasión de obstáculos tiene un conjunto de condiciones lingüísticas, que están determinadas por la cantidad de datos recibidos por los sensores. En el caso del control evasión de obstáculos se implementaron 15 reglas difusas.

1. If (FrontRight is F) and (FrontLeft is N) then (ErrorAngular is R)(Velocidad is Z) (1)
2. If (FrontRight is F) and (FrontLeft is M) then (ErrorAngular is R)(Velocidad is SP) (1)
3. If (FrontRight is N) and (FrontLeft is F) then (ErrorAngular is L)(Velocidad is Z) (1)
4. If (FrontRight is M) and (FrontLeft is F) then (ErrorAngular is L)(Velocidad is SP) (1)
5. If (RightUp is N) then (ErrorAngular is L)(Velocidad is Z) (1)
6. If (RightUp is M) then (ErrorAngular is L)(Velocidad is SP) (1)
7. If (LeftUp is N) then (ErrorAngular is R)(Velocidad is Z) (1)
8. If (LeftUp is M) then (ErrorAngular is R)(Velocidad is SP) (1)
9. If (LeftDown is N) then (ErrorAngular is R)(Velocidad is Z) (1)
10. If (RightDown is not N) then (ErrorAngular is L)(Velocidad is Z) (1)
11. If (RightUp is N) and (FrontRight is N) and (FrontLeft is F) then (ErrorAngular is L)(Velocidad is Z) (1)
12. If (RightUp is M) and (FrontRight is M) and (FrontLeft is F) then (ErrorAngular is L)(Velocidad is SP) (1)
13. If (FrontRight is F) and (FrontLeft is N) and (LeftUp is N) then (ErrorAngular is R)(Velocidad is Z) (1)
14. If (FrontRight is F) and (FrontLeft is M) and (LeftUp is M) then (ErrorAngular is R)(Velocidad is SP) (1)
15. If (FrontRight is N) and (FrontLeft is N) and (LeftUp is F) then (ErrorAngular is L)(Velocidad is Z) (1)
16. If (FrontRight is M) and (FrontLeft is M) and (LeftUp is F) then (ErrorAngular is L)(Velocidad is SP) (1)

Figura 10. Reglas difusas del control de evasión de obstáculos.

Teniendo como base los sistemas de controles ya desarrollados y sus reglas difusas implementadas, se debe tener en cuenta, que la plataforma robótica debe decidir, filtrar o priorizar qué sistema de control debe utilizar en los diferentes casos presentes en el entorno de trabajo. Lo anterior quiere decir, que los sistemas de control de velocidad, angular y de evasión de obstáculos, controlan las mismas variables de salida, con lo cual, pueden producirse conflictos, colisiones o limitantes en la utilización de los recursos del robot.

Es decir, que el robot tiene unas condiciones programadas de que control tiene la prioridad dependiendo de los eventos del entorno y que datos debe utilizar en un caso específico, delimitando los posibles conflictos e incongruencias en las respuestas de las acciones de la plataforma robótica.

### C. Implementación del Entorno de Simulación.

Con las reglas lingüísticas y el modelo de control difuso a implementar es necesario desarrollar el entorno virtual con el toolbox de Animation 3D de *MATLAB* R2014a, en donde, se comprobará el modelo implementado de navegación y evasión. Para este diseño se basa en imágenes prediseñadas en el software Solidwork o Solid Edge y en los cuales, se pueden definir fácilmente cuales son los links o DOF de la plataforma robótica a implementar y cuáles son los objetos estáticos y dinámicos presentes.

Por otra parte, en este software se puede recrear e interactuar con cualquier elemento calculado o diseñado en *MATLAB* o en otros softwares de ingeniería. Inicialmente, para el entorno virtual diseñado y desarrollado se tiene como punto de partida realizar un entorno de cancha de futbol para robots de Small Size; en donde, los otros robots son los obstáculos y la meta a recrear es el balón, que se muestra en la figura 11.

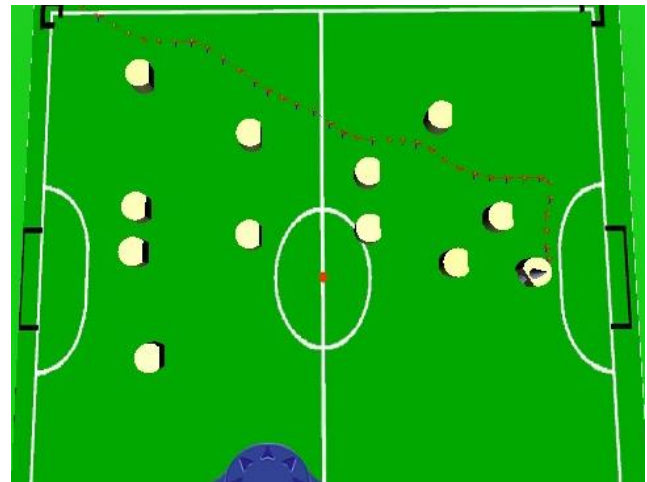


Figura 11. Diseño e implementación del entorno simulado.

Además, cada uno de los datos obtenidos por la posición actual, por la meta y por los sensores exteroceptivos son embebidos en el software de Simulink y enviados a los controles difusos de velocidad, angular y de evasión de obstáculos. Consecuentemente, cada control realiza su proceso difuso, en donde, estos controles modifican las mismas variables de salida que son el ángulo de giro y la velocidad, al modificar las mismas variables de salida se presentan conflictos en los recursos, colisiones de comunicación y problemas de coordinación, requiriendo realizar una configuración y programación detallada en el toolbox de Simulink y que se observa en la figura 12, esta programación es también evaluada en el sistema de simulación virtual de *MATLAB* y se denomina cada una de estas etapas del proceso: Datos enviados al Control, Controles Implementados y Selección de Control a Utilizar por la Plataforma. Para finalmente obtener los datos finales requeridos por las variables de salida (Angulo de giro y



Velocidad) que son enviados al bloque del entorno virtual de la plataforma Robótica y que se expone en la figura 12.

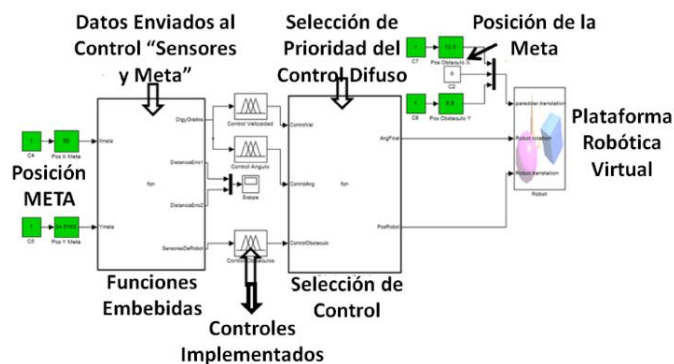


Figura 12. Desarrollo de simulación realizado en Matlab.

## VI. RESULTADOS

En esta sección se presentan los resultados obtenidos en el desarrollo del proyecto, junto con algunas imágenes de los comportamientos generados por la plataforma robótica. En este caso, se realizan inicialmente las pruebas de interacción con un simple obstáculo estático y dinámico; para proseguir con una etapa de pruebas con múltiples obstáculos estáticos y luego dinámicos y que se relacionan a cada uno de los controles difusos implementados, y que se pueden observar en las figuras 6, 8 y 10. Por otra parte el desarrollo de la plataforma virtual se presenta en la figura 13 que expone el entorno desarrollado con una vista superior.



Figura 13. Plataforma virtual vista superior.

### A. Resultados con simples Obstáculos.

Los resultados iniciales se fundamentan en las pruebas con obstáculos simples, presentando los comportamientos obtenidos por la plataforma robótica. En donde, inicialmente se realiza la prueba del control de velocidad y control angular si obstáculos con el objetivo de llegar a la posición deseada y que se expone en la figura 14.

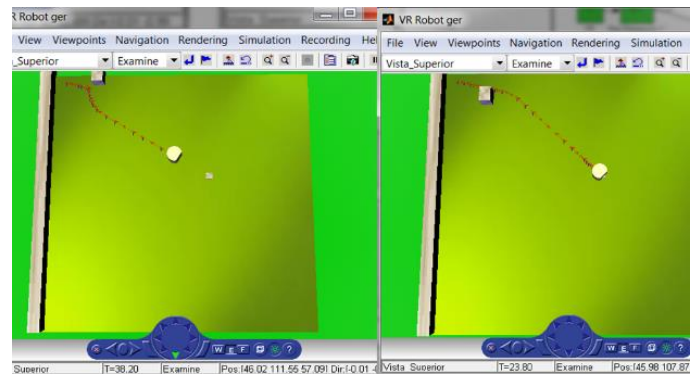


Figura 14. Plataforma virtual robot y su recorrido sin obstáculos.

Consecuentemente, en la figura 15 se realizaron múltiples pruebas con simples obstáculos estáticos y dinámicos demostrando el control de velocidad y el control de la evasión de obstáculos en el comportamiento de la plataforma robótica y logrando llegar a la posición deseada.

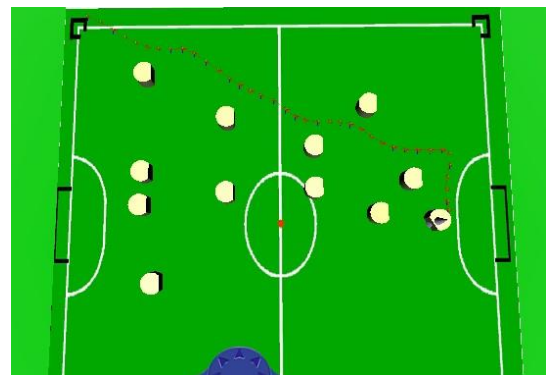


Figura 15. Comportamiento del robot y su recorrido con un obstáculo.

### B. Resultados con múltiples obstáculos

Para observar la viabilidad del modelo propuesto, se presentan algunos casos desarrollados y simulados. En donde, la plataforma robótica se traslada de una posición inicial a una posición Meta objetivo evitando los obstáculos presentes en el ambiente de trabajo y que en este caso es la cancha de futbol de Small Size. El primer caso de validación se muestra en la figura 16 en donde, se coloca una gran cantidad de robots que representan los obstáculos móviles en la cancha de futbol. Este robot con el control difuso realiza un recorrido funcional sin presentar colisiones con los obstáculos y logra llegar a la posición meta objetivo.



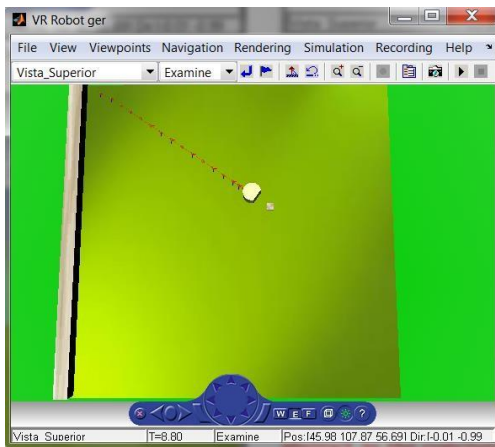


Figura 16. Caso de múltiples obstáculos presentes en una cancha de fútbol.

A continuación, se presentan las gráficas que muestran los resultados obtenidos del sistema de control difuso implementado. En la figura 17, se presenta la posición objetivo (Línea superior) y el recorrido de la plataforma robótica para lograr llegar a la posición objetivo (Línea Inferior) esta gráfica se podría denominar gráfica del error de posicionamiento del robot a la meta, objetivo. Por lo tanto se puede demostrar que la plataforma llega a la posición deseada y el error de posicionamiento es menor al 1%. En la Figura 18 se puede observar el comportamiento resultante de las dos variables de salida del control difuso que son la velocidad (a) y giro (b) requeridos por el robot.

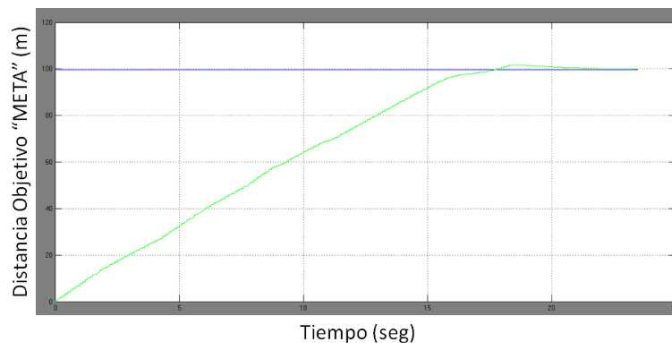


Figura 17. Error de posicionamiento del robot a la meta.

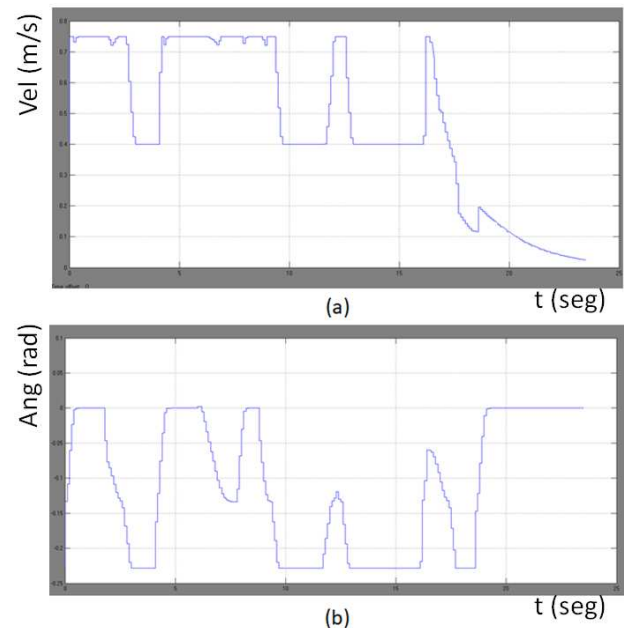


Figura 18. Comportamiento resultante del control difuso variables de salida (a) velocidad y (b) giro.

En el segundo caso figura 19, se presenta una prueba en donde se cambia la posición de la meta determinando el comportamiento dinámico de la plataforma ante los cambios presentes en el entorno de trabajo.

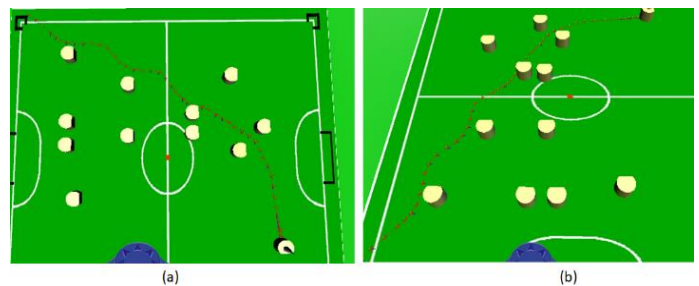


Figura 19. Cambio de la posición de la meta en la cancha de fútbol (a) vista superior y (b) vista frontal.

En la Figura 20, se presenta la posición meta, objetivo (Línea Superior) y el recorrido de la plataforma robótica para lograr llegar a la posición meta, objetivo (Línea Inferior) se denomina error de posicionamiento del robot a la meta, objetivo. Por lo tanto se puede corroborar nuevamente que la plataforma llega a la posición deseada y el error de posicionamiento es mínimo.

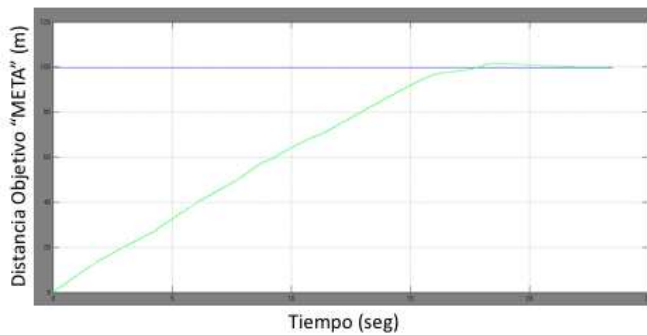


Figura 20. Error de posicionamiento del robot a la meta.

En la figura 21, se puede observar el comportamiento resultante de las dos variables de salida del control difuso de la variación del posicionamiento de los obstáculos de prueba, en donde, se observa la respuesta de la velocidad (a) y giro (b) requeridos por el robot. Por otra parte, la plataforma realiza el recorrido hacia la posición objetivo con el control de velocidad y simultáneamente realiza la corrección de su dirección o ángulo de giro.

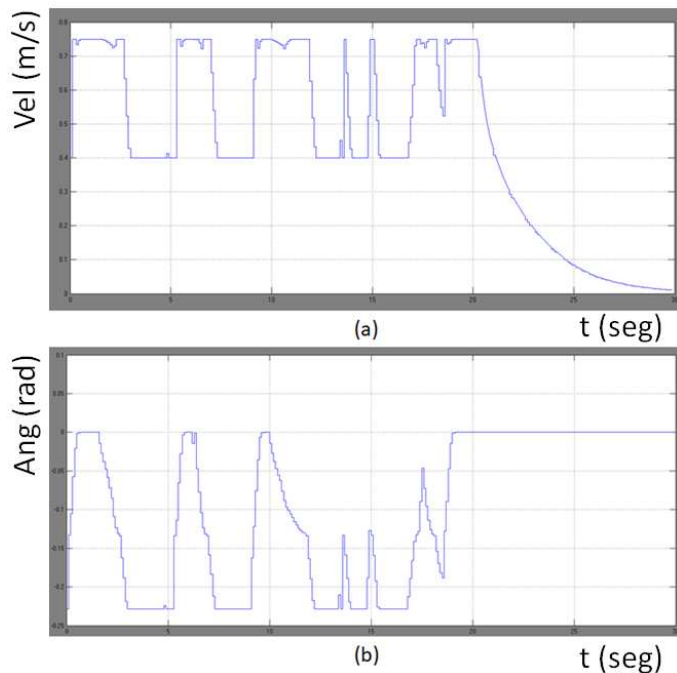


Figura 21. Comportamiento resultante del control difuso en relación a las variables de salida (a) velocidad y (b) giro.

Esto es sumamente favorable para las aplicaciones de fútbol robótico competitivo, tanto para el desplazamiento como para la velocidad de reacción, además, estos dos movimientos simultáneos no tienen ninguna limitación de movimiento, debido a que las plataformas en donde se aplicará el control difuso se mueven en cualquier dirección. Para el tercer caso, expuesta en la figura 22 se realizó el movimiento dinámico de un robot contrincante generando la reacción de la evasión del

obstáculo para lograr llegar a la meta. Teniendo en cuenta, que la distancia que genera el robot ante el obstáculo puede ser modificado por las funciones de pertenencia según necesidades de seguridad para la plataforma.

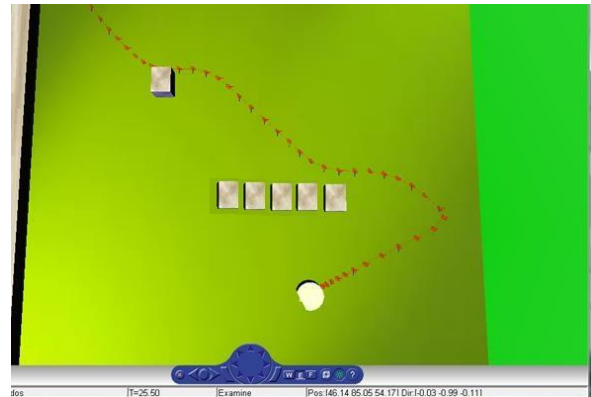


Figura 22. Movimiento dinámico de un obstáculo y el comportamiento obtenido por la plataforma robótica controlada por medio del control difuso.

En la figura 23, se presenta la posición meta, objetivo (Línea Amarilla) y el recorrido de la plataforma robótica para lograr llegar a la posición meta, objetivo (Línea morada), que se denomina error de posicionamiento del robot a la meta objetivo. Por lo tanto, se puede corroborar nuevamente que la plataforma llega a la posición deseada y el error de posicionamiento es mínimo a pesar que requiere de reacciones cambiantes debido a obstáculos dinámicos. En todos los casos, el robot es capaz de navegar su camino hacia la posición meta, objetivo con éxito evitando los obstáculos, sin importar si los obstáculos son estáticos o dinámicos.

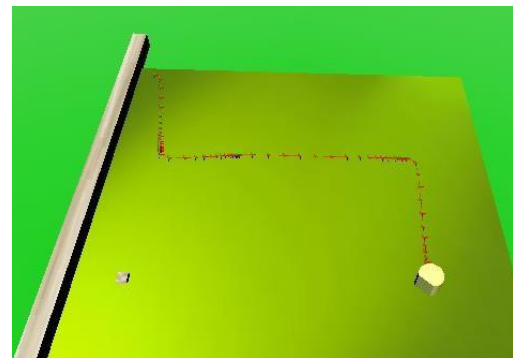


Figura 23. Error de posicionamiento del robot en relación a la meta.

Otros de los casos realizados son las pruebas pertinentes al comportamiento del robot a cambios bruscos de posición meta, objetivo para determinar la velocidad de reacción del sistema de control y su funcionalidad en las plataformas aplicadas. En la figura 24, la plataforma presenta cambios repentinos de dirección debido a que se le asigna múltiples posiciones del meta objetivo. En la figura 25, se presenta la posición de la meta que varía en diferentes instantes de tiempo

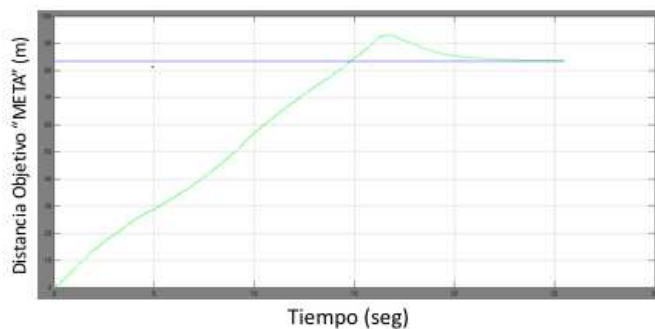


Figura 24. Movimiento dinámico de un robot (obstáculo) y comportamiento obtenido por la plataforma robótica controlada por medio del control difuso.

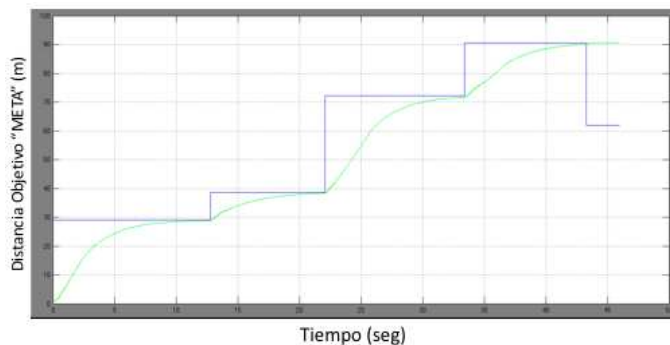


Figura 25. Error de posicionamiento del robot a la meta objetivo.

(Línea Superior) y el recorrido de la plataforma robótica para lograr llegar a la posición meta, objetivo (Línea Inferior), que se denomina error de posicionamiento del robot al meta objetivo. Observando que la plataforma está diseñada para realizar el recorrido hacia la meta con un tiempo de estabilización aproximadamente de 10 segundos, el error de estado estacionario es de un 2% y el sobre impulso máximo es insignificante. Finalmente se puede observar que el tiempo de respuesta de la plataforma es relativamente elevado en el orden de los microsegundos dependiendo del procesador que se está utilizando para el control difuso. En el trabajo experimental, la funcionalidad del sistema propuesto fue probada en un entorno que simula el entorno dinámico lleno de robots que juegan fútbol (Robo-Cup Small Size).

El ambiente fue diseñado teniendo en cuenta varias situaciones, tales como: robots en movimiento múltiples, robots obstaculizando el desplazamiento y obstrucciones fijas múltiples. En todos los casos el robot alcanzó la posición meta objetivo con una tasa de éxito del 90% sin embargo el 10% es por la proximidad del robot a los obstáculos dinámicos que limita la reacción y genera otros tipos de programas para casos especiales en que los obstáculos dinámicos se dirigen hacia el robot. teniendo en cuenta este caso especial, el robot en un 90% ejecuta sus movimientos y genera un comportamiento de acuerdo con las reglas generadas en los casos de dirección a la meta y cuando detecta un objeto u otro robot cambia al modo de control de evasión de obstáculos y cuando logra evadir el

obstáculo, vuelve a tomar el modo de llegar a la posición meta objetivo hasta lograr llegar a la meta.

## VII. TRABAJOS FUTUROS RELACIONADOS

Uno de los trabajos futuros es la implementación de este control difuso en una plataforma real y realizar sus correspondientes experimentaciones para aplicarlo a una plataforma robótica o plataformas robóticas cooperativas; como es en una competencia de fútbol de Robots de Small Sise o en un conjunto de robots industriales con obstáculos fijos y dinámicos que representan las personas del entorno; validando el modelo propuesto junto al tiempo de reacción y la evasión de obstáculos dinámicos complejos.

## VIII. CONCLUSIONES

Debido a las múltiples necesidades de navegación requeridos en las plataformas robóticas como de la Robo-Cup una excelente manera de realizar y estructurar la navegación se presenta en este artículo. Las dificultades en los comportamientos esperados en un robot para evitar colisiones con su entorno o con otros robots fueron desarrollados con este artículo y con la lógica difusa y se pueden complementar con el video que se encuentra en la Web. En donde, los automatismos y programación de la plataforma robótica cumplieron con la evasión y navegación en entornos con obstáculos dinámicos teniendo en cuenta la aplicación de controles difusos y condiciones simples para la coordinación y resolución de conflictos internos del robot simulado en esta investigación. [1].

En el trabajo experimental, la funcionalidad del sistema propuesto fue probada en un entorno que simula el entorno dinámico lleno de robots que juegan fútbol (Robo-Cup Small Size) [16]. El ambiente fue diseñado teniendo en cuenta varias situaciones, tales como: robots en movimiento, múltiples robots obstaculizando el desplazamiento y obstrucciones. En todos los casos el robot tiene que alcanzar la posición meta objetivo. En donde, el robot ejecuta sus movimientos y genera un comportamiento de acuerdo con las reglas generadas en los casos de dirección a la meta y cuando detecta un objeto u otro robot cambia al modo de control de evasión de obstáculos, al lograr evadir el obstáculo vuelve a tomar el modo de control de velocidad y dirección con el fin de lograr llegar a la meta con éxito. Teniendo en cuenta que los tiempos de procesamiento de los sistemas de control están en el orden de los 40 us o menores, pero, varían por el tipo de procesador utilizado y la virtualización (animación) del proceso que acapara recursos.

## REFERENCIAS

- [1] H. Báez, «Difuso», 2017. [En línea]. Available: <https://www.youtube.com/watch?v=3zR-KxpvQRO..>
- [2] A. Y. y. A. M. S. Amur, «Fuzzy Logic Based Navigation of Mobile Robots Recent Advances», *Mobile Robotics*, pp. 287-310, 2011. , 2011.
- [3] M. G. y. H. Hoyer, «FUZZY Collision Avoidance», *Industrial Robots IEEE*, pp. 1-16, 1995.
- [4] T. K. a. H. O. W. K. Tanaka, «Backing Control Problem of a Mobile Robot with Multiple Trailers», *Fuzzy Modeling and LMI-Based Design IEEE, Transactions on Systems, Man, and Cybernetics* , vol. 28, nº 3, pp. 329-337, 2003.
- [5] E. P. D. y. O. A. Maravillas, «Cooperative Mobile Robots with Obstacle and Collision Avoidance Using Fuzzy Logic Proceedings of the 2002 IEEE», *International Symposium on Intelligent Control*, pp. 75-80, 2002.
- [6] T. Y. T. S. Y. H. y. H. H. T. Kamano, «Design and Implementation of Fuzzy Cooperative Catching Controller for Multiple Mobile Robots IEEE», *Mobile Robots IEEE*, pp. 1749-1754, 2000.
- [7] R. Haber, «Aplicaciones de la lógica borrosa en control: tres alternativas interesantes y exitosas pp.», *Ingeniería Electrónica, Automática y Comunicaciones*, vol. XXIV, nº 3, pp. 37-46, 2003.
- [8] V. G. y. B. S. S. Parasuraman, «Fuzzy Decision Mechanism Combined with Neuro-Fuzzy Controller for Behavior Based», *Robot Navigation IEEE*, pp. 2410-2416, 2003.
- [9] F. Amer, «Fuzzy Pre-compensated Fuzzy Self-Tuning Fuzzy PID Controller of 3 DOF», *Planar Robot Manipulators IEEE/ASME International Conference on Advanced Intelligent Mechatronics*, pp. 1-6, 2010.
- [10] C. L. H. y. S. F. Chao., «A Fuzzy-Model-Based Variable Structure Control for Robot Arms: Theory and Experiments», *IEEE International Conference on Systems, Man and Cybernetics*, pp. 5252-5258., , 2004.
- [11] T. Sugeno, «Fuzzy Modeling for Process Control Inglaterra: Industrial Automation», *Robotics and Artificial Intelligence*, 2008.
- [12] S. G. y. E. A. A. J. Bernal, «Robot seguidor de ruta por medio de una plataforma diferencial», *Revista Inventum*, pp. 7-13, 2012.
- [13] F. Z. H. M. Y. S. F. T. S. Z. S. Y. N. H. O. N. Z. Azlan, «Fuzzy Logic Controlled Miniature LEGO Robot for Undergraduate Training», *System IEEE*, pp. 2184-2188, 2007.
- [14] S. G. T. y. P. Zavlangas, «Industrial and Mobile Robot Collision Free Motion Planning Using Fuzzy Logic Algorithms Industrial Robotics», *Theory, Modelling and Control*, pp. 301- 334, 2006.
- [15] R. Babuska, «Fuzzy Modeling for Control USA», *Kluwer Academic Publishers*, 1998.
- [16] Robo Cup Federation, «Robo Cup Federation», Noviembre 017. [En línea]. Available: <http://www.robocup.org/leagues/7>.