

PLANNED NAVIGATION OF A LEGO NXT ROBOT

Steffen Jose Cantillo Molina, Julie Stephany Berrío Pérez

Abstract — The following paper presents how several navigation algorithms, methodologies and the tests' results, using three global (Voronoi diagrams, occupation maps, visibility graphs) and three local (neural network, fuzzy logic, potential fields) techniques, for a Lego NXT mobile robot to arrive to a settled goal were developed. Analyzing the techniques' pros and cons concerning to events the robot may face, and how the structures' own features stablish parameters for the robot reaching its goal are detailed. Next, every technique's algorithm and the field conditions such as obstacle evasion and next space available are described. Elements such as used amount of time, movement efficacy and the robot's reaction speed were chosen as the results' main criteria, due to the performance and efficiency rates of each technique. After evaluation, the advantages each technique has compared to the others will be shown in a table. Finally, the obtained solutions are put to test in hypothetical events in order to demonstrate how every technique can be applied; concluding that determining which technique to use depends mainly on the test conditions and the amount of environment data available.

Keywords—Robotics, Navigation, Planning, techniques.

I. INTRODUCCIÓN

El avance tecnológico ha arrojado resultados que equiparan los sueños de Da Vinci, tal es el caso del robot ASIMO de la compañía Honda (robot humanoide capaz de movimiento autónomo y realización de sencillas tareas). Este robot recuerda ligeramente a uno de sus inventos más espectaculares, que fue su prototipo de armadura autómatas, que consistía en una armadura capaz de moverse independientemente, empleando un sistema de poleas dentro de la misma.

Con el paso del tiempo, la capacidad de los robots para desarrollar tareas simples pero convenientes para su propósito ha aumentado, teniendo como ejemplos el ya mencionado ASIMO, la aspiradora robot, o el robot espacial Curiosity, mandado a Marte para su exploración, surcando terrenos, por lo general rocosos, evitando cualquier inconveniente que le impida cumplir su cometido. Para esto, Curiosity, además de tener un control en Tierra, posee la capacidad de estimar y/o planear trayectorias por las cuales transitar.

El planeamiento de trayectoria se encarga de proyectar los movimientos del robot a lo largo del camino planeado de

S. Cantillo es estudiante de Ingeniería Mecatrónica de la Universidad Autónoma del Caribe, Barranquilla, Colombia (e-mail: steffen.cantillo@uac.edu.co).

J.S. Berrío es PTC de Ingeniería Mecatrónica de la Universidad Autónoma del Caribe, Barranquilla, Colombia (e-mail: julie.berrio@uac.edu.co).

forma tal que no se afecte la maniobrabilidad de éste, es decir no se sobrepasen las restricciones en cuanto a velocidades, aceleraciones y radios de giro del robot específico sobre el cual se esté trabajando. [1]

El propósito en este trabajo es ensamblar un robot y programarlo para que alcance un objetivo establecido por un área establecida y una meta definida, en la cual encontrará obstáculos que ha de evadir a medida que avanza.

II. NAVEGACIÓN EN ROBOTS MÓVILES

Las tareas involucradas en la navegación de un robot móvil son: la percepción del entorno a través de sus sensores, de modo que le permita crear una abstracción del mundo; la planificación de una trayectoria libre de obstáculos, para alcanzar el punto destino seleccionado; y el guiado del vehículo a través de la referencia construida. De forma simultánea, el vehículo puede interactuar con ciertos elementos del entorno. Así, se define el concepto de operación como la programación de las herramientas de a bordo que le permiten realizar la tarea especificada [2].

III. ESQUEMAS DE NAVEGACIÓN

La clave del esquema para adaptarse a diversos entornos, se pueden definir con mayor precisión de la siguiente forma:

A. Planificación global:

Construir o planificar la ruta que lleve al robot a cada una de las submetas determinadas por el control de misión, según las especificaciones del problema que debe resolverse. Esta planificación es una aproximación al camino final que se va a seguir, ya que en la realización de esta acción no se consideran los detalles del entorno local al vehículo [3].

1) Planificación basada en diagramas de Voronoi

Sea S un conjunto de entidades geométricas en un plano, el diagrama de Voronoi corresponde a una partición del plano en torno a S , siendo los puntos pertenecientes a cada región de Voronoi los más cercanos a una sola entidad de S [4].

En robótica móvil han sido aplicados el espacio correspondiente a la distancia euclidiana entre puntos y el principio del vecino más cercano. Así pues, si se considera a $P = \{p_1, p_2, \dots, p_n\}$ un conjunto de puntos no alineados en el plano y sea $d(p_i, p_j)$ la distancia euclidiana entre los puntos p_i y p_j , la región Voronoi $R(p_i)$ generada por el punto p_i queda definida por la ecuación [5]:

$$R(p_i) = \{p \in R^2; d(p, p_i) \leq d(p, p_j), \forall j \neq i\}$$

En este contexto, a los puntos p_i se les denomina generadores de Voronoi y al conjunto de todas las regiones de Voronoi $R(p_1), R(p_2), \dots, R(p_n)$, diagrama de Voronoi de P, como puede verse en la figura 1. Además, a una frontera común a dos regiones se le denomina arista de Voronoi y a un punto de corte entre tres o más aristas, vértice de Voronoi [5].

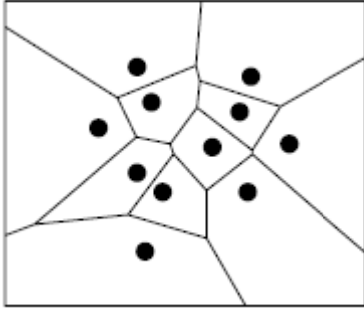


Figura 1. Diagrama de Voronoi. Figura tomada de *Computational Geometry*. de Berg, M., Cheong, O., van Kreveld, M., Overmars, M. (2008). Springer (Ed.) ISBN 978-3-540-77973-5. P 155

2) Planificación basada en grafos de visibilidad

Los grafos de visibilidad proporcionan un enfoque geométrico en un entorno bidimensional en el cual los obstáculos están modelados mediante polígonos. Para generar un grafo de visibilidad, se han de tener dos puntos que se puedan unir con un segmento de recta si y solo si la recta no interseca un obstáculo (a lo que se le denomina visibilidad). De haber un segmento tangencial a algún obstáculo, son visibles los puntos afectados. En otras palabras, el segmento definido debe yacer en el espacio libre del entorno [6]. Así, considerando como nodos de los grafos de visibilidad la posición inicial, la final y todos los vértices de los obstáculos del entorno, el grafo resulta de la unión mediante arcos de todos aquellos nodos que sean visibles. (ver figura 2)

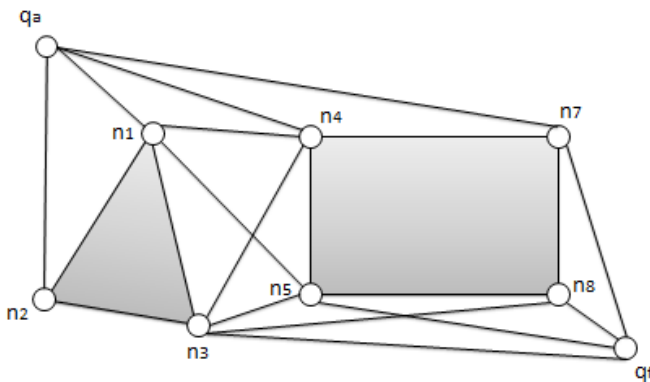


Figura 2. Grafo de visibilidad en un entorno de dos obstáculos

3) Planificación basada en Mapas de Ocupación

Se basan en representar el terreno como una retícula, donde cada una de las casillas contiene un valor útil para el robot, que suele ser la certitud de ocupación, es decir, desde -1 (es

seguro que está libre) hasta +1 (es seguro que está ocupada) pasando por 0 (no hay evidencia en ningún sentido). Estos mapas se pueden construir por métodos visuales, mediante la toma de imágenes por un par estéreo de cámaras (o una sola que va a bordo del robot y se sitúa en varias posiciones), a partir de las proyecciones de puntos límite de un objeto, como se ve en la siguiente figura [7]:

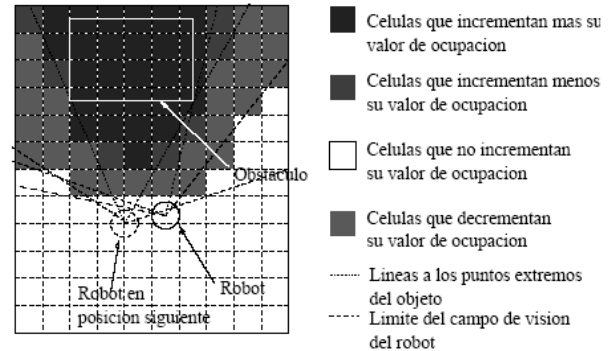


Figura 3. Construcción visual de un mapa de ocupación. Imagen tomada de "Apuntes de Robótica" de Juan Domingo Esteve, Universitat de Valencia. (2001) Pp 7/14.

B. Planificación local:

Resolver las obstrucciones sobre la ruta global en el entorno local al robot para determinar la ruta real que será seguida. El modelo del entorno local se construye mediante la fusión de la información proporcionada por los sensores externos del robot móvil [3]. Dependiendo de la técnica, así será cómo el robot sabe a dónde llegar

1) Campos de potenciales artificiales

En el método de campos de potencial artificial, el robot se concibe como una partícula inmersa dentro de un campo de potencial cuyas variaciones locales reflejan la estructura del entorno, como se ve en la figura 4. Los obstáculos son modelados por cargas repulsivas y la meta como una carga atractiva. El movimiento del robot se establece como modo iterativo, calculando en cada paso la fuerza generada por el campo de potencial y utilizado en la dirección de ésta para modificar el comportamiento del robot [8][9].

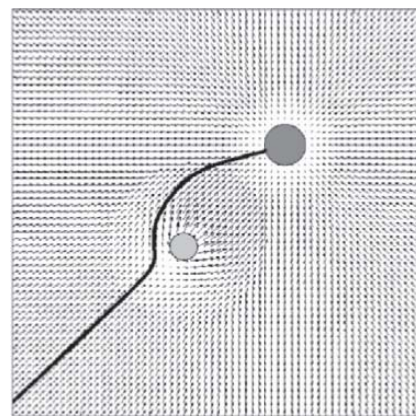


Figura 4. Trayectoria en un ambiente con un obstáculo y un objetivo con campos de potencial artificial. Imagen tomada de *Aplicación del método de campos de potencial artificial para un robot móvil autónomo*. Bermúdez Bohórquez, G. R., Rojas Castelar, L. A., Montiel, H., & Ceballos, M. (2004). Bogotá: Universidad Nacional.

2) Lógica difusa

Un conjunto difuso A en un universo X es un conjunto definido por una función de membresía μ_A representando un mapeo:

$$\mu_A: X \rightarrow \{0,1\}.$$

Donde el valor de $\mu_A(x)$ para el conjunto difuso A es llamado valor de membresía o grado de membresía de $x \in X$. El grado de membresía representa el grado de x perteneciente al conjunto difuso de A [10]. Las entradas hechas difusas son inferidas en una base de reglas difusas, que es usada para caracterizar la relación entre entradas y salidas difusas. Por ejemplo, si x es A entonces z es X, donde A y C son valores difusos definidos en el universo x y z, respectivamente. Este mecanismo de inferencia provee un arreglo de acciones de control de acuerdo a las entradas difusas. Como las salidas están difusas, un método de defuzzificación es requerido para transformar salidas difusas en valores de salida que pueden ser aplicados en un plano real [11]. De esta forma, el rango de movimiento del robot es definido por criterios basados en las lecturas que obtenga el sistema sensorial del vector (ver figura 5).

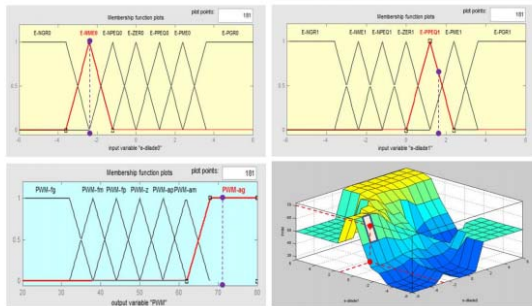


Figura 5. Partición de variables lingüísticas en conjuntos difusos y superficie de control del sistema de navegación fuzzy del Robot Walling. Figura tomada de *Motion control and navigation of multiple mobile robots for obstacle avoidance and target seeking: a rule-based neuro-fuzzy technique*. Pradhan, S. K., Parhi, D. R., & Panda, A. K. (2009). *Proceedings of the Institution of Mechanical Engineers*, 275-287.

3) Redes neuronales

Una red neuronal es un modelo computacional con un conjunto de propiedades específicas, como son la habilidad de adaptarse o aprender, generalizar u organizar la información, todo ello basado en un procesamiento eminentemente paralelo [12] (ver figura 6). En el ambiente de entrenamiento se ubican obstáculos asegurando una gran cantidad de patrones de entrenamiento para el correcto aprendizaje de la red neuronal [13]. El control del robot se basa en variables relacionadas con el ambiente tales como:

- Angulo del robot.
- Coordenadas del robot.
- Coordenadas de la meta.

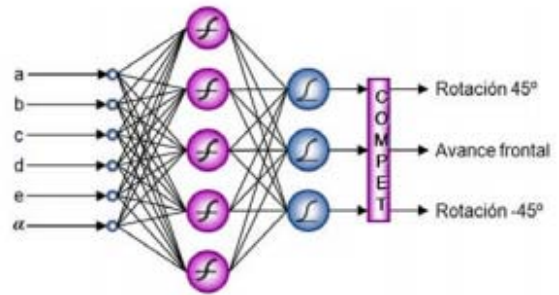


Figura 6. Estructura de la red neuronal (MLP). Figura tomada de *Navegación planificada de un robot móvil*. Berrio, J. S. (2009). Cali: Universidad Autónoma de Occidente.

IV. DESCRIPCIÓN DE LOS CRITERIOS DE EVALUACIÓN

Para este trabajo, los criterios de evaluación para las técnicas estudiadas se basan en el rendimiento y eficiencia que cada una pueda aportar a la navegación del robot. Los criterios, en específico, constan de:

A. Tiempo de acción

Se trata de mediciones del tiempo, en segundos, a distintas pruebas realizadas a cada técnica, que al final de cada ronda se compararan para obtener un promedio de cuánto dura el robot en alcanzar la meta.

B. Velocidad de reacción

Determina cómo el robot reacciona frente a los obstáculos, ya sea si logra evadir obstáculos con rapidez, o al chocar cuánto se demora en alejarse del obstáculo. Bajo este criterio tan bien se analiza la reacción esperada con respecto a la real.

C. Manejabilidad

Se evalúa qué tanto se comporta el robot con respecto a lo esperado, es decir, si el robot se mueve de manera acorde a la técnica propuesta, como por ejemplo, que tan frecuente el robot choca con algún objeto, cuánto tiempo queda encasillado en algún lugar del campo, etc.

D. Sentido de orientación

Se evalúa si el robot se mantiene en ruta hacia la meta y cuánto puede llegar a ser su desfase angular en grados con respecto a la misma, es decir, si avanza directamente hacia la meta, si se aleja de ésta, si llega rodeándola, entre otras.

A partir de las observaciones de cada criterio, se obtiene para cada técnica, una conclusión en la cual se resaltan las cualidades y defectos de los mismos. Posteriormente, se comparan todas las técnicas para determinar una ruta rápida, eficiente y adecuada para el robot, dependiendo de la situación en la que se encuentre.

V. DISEÑO DEL EXPERIMENTO

Como ya se ha mencionado, un robot será puesto en un campo lleno de obstáculos estáticos, los cuales ha de sortear

para alcanzar una meta específica. Para lograr este cometido, el robot cuenta con varios sistemas sensoriales y una serie de algoritmos cuyo objetivo es hacer que el robot llegue a la meta por sí solo.

Se escogió el lenguaje de programación leJOS (Java para robots LEGO NXT) para escribir los comandos respectivos al robot. La razón de esta elección radica en la libertad de acción con respecto al lenguaje original del robot LEGO NXT, por ejemplo, mayor rango de control de la velocidad del motor, mejor manejo de variables locales y globales, cálculos extensos, entre otros. Con esto, en el caso de campos de potencial, obtener un resultado preciso se hace posible.

Para el procesamiento de imágenes la herramienta MATLAB, debido a su versatilidad. Con la posibilidad de extraer las propiedades de la imagen a trabajar y a continuación aplicar cualquier otra acción que pueda transformar los datos arrojados por la imagen a comandos que pueda entender el robot.

Siendo éste un trabajo comparativo, las condiciones del campo (varias cajas ubicadas en un lugar específico sobre una superficie opaca), serán iguales para todas las técnicas (locales y globales) debido, principalmente, a la limitada memoria del LEGO NXT Brick no alcanzaría para un análisis dinámico (por ejemplo reacciones inmediatas en colisiones con obstáculos dinámicos en técnicas locales), por tanto, la cantidad de datos necesarios puede ralentizar la capacidad de procesamiento del sistema.

A. Navegación local

En las pruebas correspondientes a las técnicas de navegación local se establecieron condiciones iniciales, iguales para todas. Éstas consistieron en que el robot (con una posición angular relativa con respecto a la meta, designada previamente) avanza por el campo en línea recta durante 1000 milisegundos a una velocidad de 540 grados/segundo y al finalizar, el subsistema (localizado en la parte superior del robot) compuesto por un sensor ultrasónico montado en un motor es activado por fases, en las cuales el motor gira cierta cantidad de grados determinada y el sensor obtiene lecturas, que serán guardadas en un vector dinámico. Dependiendo de la técnica que se trate, así será la activación del sensor, por ejemplo:

1) Red neuronal

Estas fases consisten en un giro de 36° del motor (partiendo desde 0° hasta 180° , para un total de 5 giros) Una vez obtenidas las 5 lecturas, se guardan en un vector que, junto a la posición angular del robot con respecto a la meta, servirán como entradas para la red neuronal diseñada.

Esta red multicapa con capa oculta cuenta con seis entradas, cinco de ellas corresponden a las lecturas ultrasónicas y la sexta correspondiente a la posición angular relativa del robot. La red procesa estos datos en paralelo y arroja dos datos de salida, uno para definir el sentido de giro del robot (derecha, izquierda, derecho) y la orientación del movimiento (adelante, atrás), (ver figura 7).

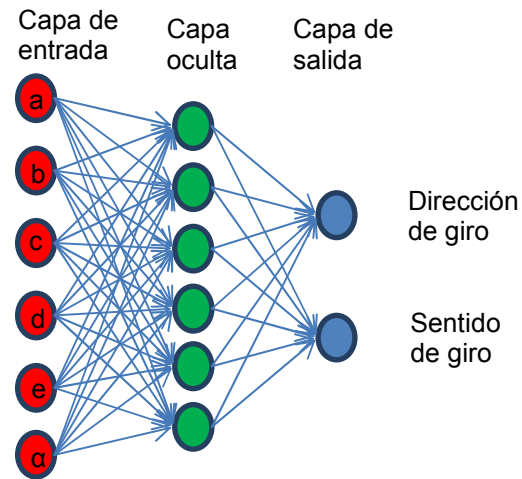
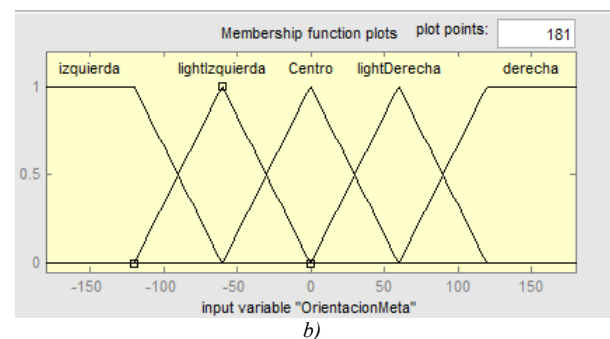
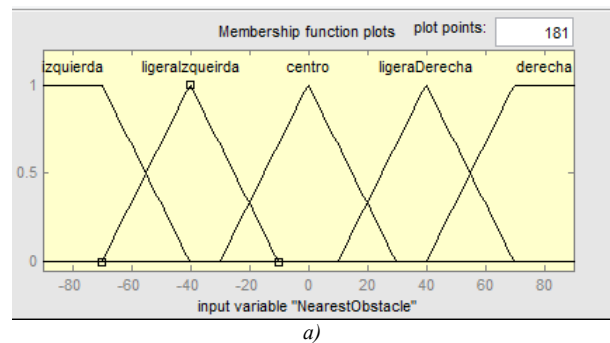


Figura 7. Esquema de una red neuronal. Figura creada por el autor.

2) Lógica difusa

En este caso cada uno en intervalos de rotación es de 10° , por tanto, 18 lecturas. Adicionalmente, se crea un vector que contenga los valores de las posiciones angulares del sensor ($10^\circ, 20^\circ, \dots, 180^\circ$). De cada vector, se escoge el mayor de los datos y se guardan en variables que serán comparadas con los valores de vectores de referencia previamente definidos. De ser iguales, se toman los valores de posición de los primeros, y con estos se realiza una búsqueda en la tabla lookup (obtenida a través del método Mamdani-Sugeno). Dentro de esta tabla se almacena (en un rango entre -60° y 60°) las posibles direcciones que tomará el robot en su siguiente movimiento (ver figura 8).



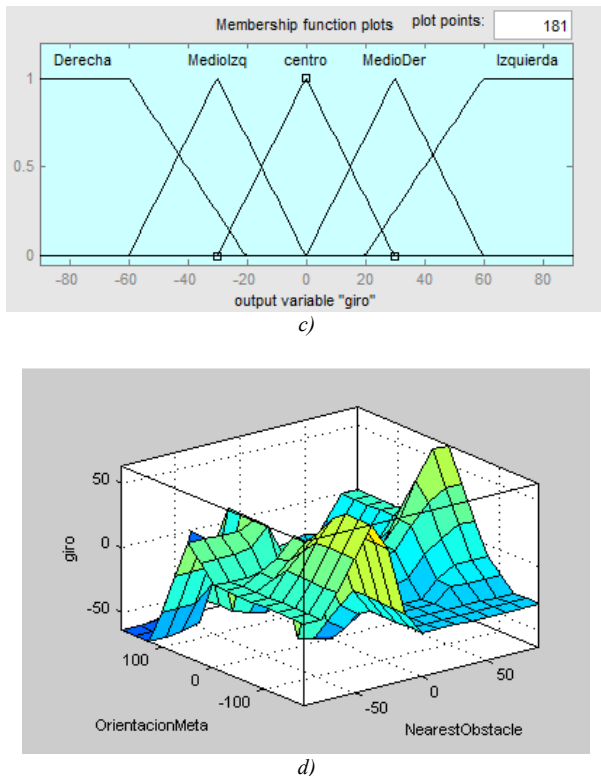


Figura 8. Partición de variables lingüísticas en conjuntos difusos y superficie de control del sistema de navegación fuzzy del Robot LEGO NXT. a) variables correspondientes a la lectura del sensor ultrasónico. b) variables correspondientes a la posición angular relativa del robot con respecto a la meta. c) variables con respecto a la dirección de giro. d) superficie de las variables. Figura creada por el autor.

3) Campos de potencial

En este caso, cinco lecturas cada una en intervalos de rotación de 45° . Con estos datos se realiza un cálculo vectorial para determinar la nueva dirección del robot. Para este cálculo, se define cada vector a partir de la magnitud de la lectura del sensor y la posición angular del motor al momento de la lectura:

$$x = d_3 + d_2 * \cos(45^\circ) + d_4 * \cos(-45^\circ)$$

$$y = d_1 - d_5 + d_2 * \sin(45^\circ) - d_4 * \sin(-45^\circ)$$

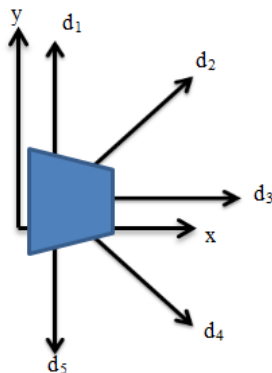


Figura 9. Representación vectorial para las lecturas del sensor en campos de potencial. Figura creada por el autor.

se toma como referencia positiva el lado izquierdo del

robot, por tanto, de ser el resultado positivo, el robot girará a la izquierda, y de ser negativo a la derecha. Una vez arrojado el resultado, el robot girará y avanzará en línea recta en la dirección recién definida, se detendrá y obtendrá la lectura del sensor de color (ver figura 9).

En todos los casos, el robot avanzará por un periodo de un segundo a 540 grados/segundo en la dirección recién definida. Pasado este lapso, se detiene, actualiza la posición angular relativa del robot (reemplazando el antiguo valor con el nuevo valor arrojado por la técnica), y obtendrá la lectura del sensor de color, ubicado en la parte inferior del robot. Si a lectura equivale al color azul (color de la meta), el proceso se detendrá, de lo contrario seguirá el proceso hasta conseguirlo.

B. Navegación global

Las condiciones iniciales de misma configuración del campo, velocidad constante de 540 grados/segundo se aplican también para las técnicas globales. Dado esto, y la utilización de una cámara colocada por encima del campo, se toma la imagen del mapa a navegar, para poder reconocer las figuras presentes en él (robot, meta, obstáculos) detallando las características de las mismas (área, puntos extremos, centroides, etc.). Una vez recopilada esta información, con ayuda de Matlab se obtienen puntos cardinales que servirán de guía para las rutas.

La razón por la cual se optó por utilizar Matlab fue por comodidad de procesamiento, es decir, sabiendo que el programa cuenta con una gran cantidad de funciones para procesar imágenes y hacer los cálculos vectoriales necesarios para las rutas a seguir (distancias entre punto y punto, ángulo de desfase del vector con respecto a la horizontal, etc.). Para que el robot entendiera los resultados arrojados por Matlab, el programa en lenguaje leJOS controla solamente el giro y avance de los motores, lo que significa que la interacción entre el procesamiento de imágenes y el programa ejecutado es manual por parte del usuario.

1) Grafos de visibilidad

Las características servirán como patrones de entrenamiento para la red neuronal base del algoritmo. Esta red arroja las coordenadas de los puntos extremos de cada obstáculo, además de aquellas de los centroides del robot y de la meta, que serán graficadas encima de la foto. A continuación, se comprueba la vecindad de todos los puntos, y aquellos que cumplan estas condiciones, se conectan con segmentos de línea partiendo del centroide del robot hasta el de la meta. Una vez conectados todos los puntos, se desfazará el camino para establecer un camino libre de obstáculos (ver figura 10).

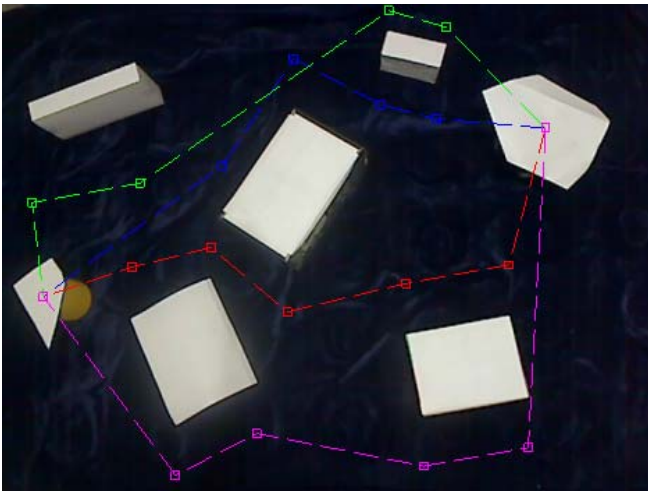


Figura 10. Trazado de caminos usando Grafos de Visibilidad. Figura creada por el autor.

2) Diagramas de Voronoi

Se seleccionan los centroides de los obstáculos para formar los diagramas de Voronoi. A partir de la selección de los centroides, varios segmentos de recta se trazan sobre el campo, los cuales se intersectan unos con otros hasta formar las respectivas celdas para cada objeto en el campo. Una vez formados, se procede a seleccionar los vértices vecinos, que serán utilizados para trazar el camino desde el robot hasta la meta. Para evitar una paradoja en la construcción de vértices, aquellos segmentos de recta que sobresalen el borde del campo se conectan a un vértice infinito. Con esto se dejan definidos los vértices finitos, que servirán como base para la construcción de los caminos (ver figura 11).

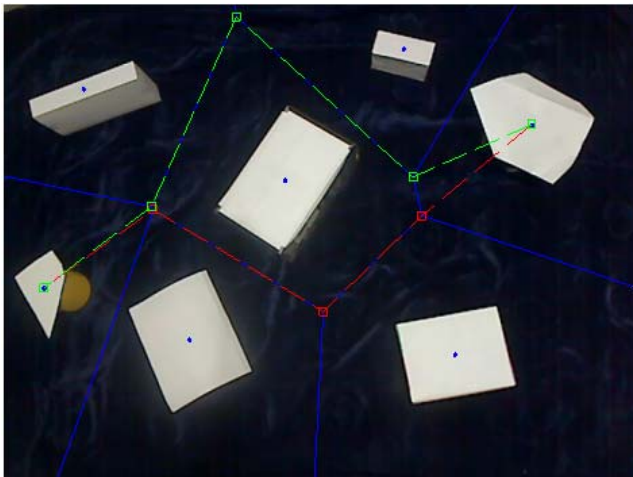


Figura 11. Trazado de caminos usando Diagramas de Voronoi. Figura creada por el autor.

3) Mapas de ocupación

El campo es dividido en cuadrillas, que corresponden a cierta cantidad de píxeles de la imagen. Si se trata de píxeles de alguno de los obstáculos, se hará una subdivisión de las ya existentes cuadrillas. Paralelo a esto, se grafican puntos aleatorios que vayan desde el robot hasta la meta. De cruzarse uno de estos puntos con los píxeles de cualquiera de los obstáculos, se desfazará hasta quedar posicionado en una

sección libre de obstáculos. Una vez confirmado que no haya un espacio ocupado en cruce con el camino a construir, se traza las secciones de rectas conectoras de los puntos dispuestos (ver figura 12).

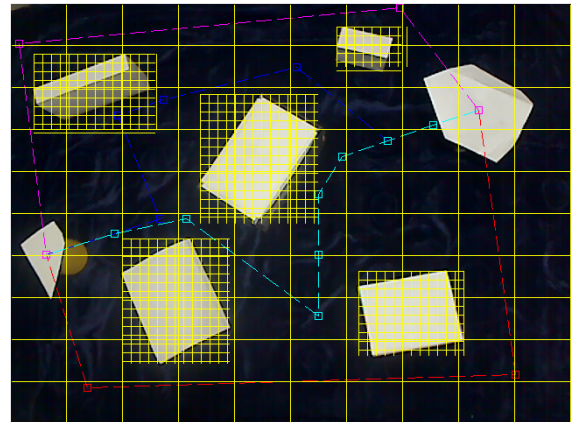


Figura 12. Trazado de caminos usando Mapas de Ocupación. Figura creada por el autor.

VI. RESULTADOS

A. Navegación local

Basándose en los valores puntuales, la retroalimentación en estas técnicas parte de la incertidumbre del robot con respecto a lo que se encuentra más allá de su campo de percepción. Factores como la cantidad de datos utilizados para procesar una reacción frente a eventos acontecidos durante su navegación o la complejidad de los cálculos necesarios para dicha reacción no necesariamente determinan superioridad de una determinada técnica con respecto a las demás, pero resalta la conveniencia que pueda tener para con ciertas situaciones.

Refiriéndose a la cantidad de datos utilizados, es fácil pensar que a mayor número de datos mejor el resultado, pero lo observado en lógica difusa, donde se recopilaban varias lecturas, se demostró que al solo escoger el valor mayor del vector del sensor (con su respectiva posición angular), el robot tendía a entrar en un bucle casi infinito (interrumpido por un choque contra un obstáculo o una pared), siendo éste consecuencia del encasillamiento en una sola región en la tabla look-up (más que todo por la coincidencia de que casi siempre el dato de posición angular no variaba). Esto sucedió por la presencia de mínimos locales, la cual causa que el robot llegue a un punto de estancamiento.

Siguiendo esta línea, la red neuronal y los campos de potencial deberían de estar mano a mano por, aparentemente, ser muy parecidos entre sí. Si se observa detalladamente, con campos de potencial era más probable llegar a una ruta más directa hacia la meta al calcular vectorialmente el mejor espacio para avanzar, en vez de inferir cual dirección estaba más despejada tal como arrojó la red neuronal.

Sin embargo, con la red neuronal se reestablecía el punto origen y la orientación del robot, y considerando que tuvo un comportamiento más “impulsivo” (avance arremetedor contra un obstáculo en su punto ciego), y un poco de suerte (el obstáculo cedía ante su empuje, o lograba separarse siquiera

un par de milímetros), que en ciertas ocasiones creaban una desorientación parecida a la vista en el bucle de la lógica difusa, originando así desvíos y alargamientos en el camino hacia la meta. Por el contrario, al recurrir a un cálculo vectorial en campos de potencial, se obtuvo una distancia exacta entre la meta y la siempre cambiante posición del robot, denotando una ruta rápida y directa en cada ocasión, a pesar del hecho de que el robot tiende a buscar el espacio más despejado, lo que significa que puede que se aleje aún más. No obstante, saber la distancia contribuye a un mayor conocimiento de la situación.

En la tabla 1 se puede apreciar que los tiempos de campos de potencial son ligeramente mayores a los de la red neuronal; pero no solo con llegar rápida a la meta rápidamente se obtienen condiciones ideales a la hora de realizar la tarea establecida. Por esta y varias razones más (reflejadas en los criterios de evaluación), la técnica de campos de potencial resultó ser la más adecuada.

B. Navegación global

Con las técnicas globales se tiene la posibilidad de obtener previamente conocimiento del entorno y determinar con mayor facilidad una ruta adecuada para el robot.

En la tabla 2 se puede apreciar que todas las anotaciones son prácticamente las mismas, queriendo decir que un campo con obstáculos estáticos no representa gran reto para el robot con una visión panorámica, además de que el promedio de los tiempos de todas las técnicas tiende a 17 segundos, haciendo que la diferencia no sea tan determinante. En este caso, es más prudente analizar otros factores como la facilidad de procesamiento de datos, la cantidad de los mismos o la distribución espacial que se obtiene de cada una.

A simple vista, los grafos de visibilidad ofrecen una gran libertad a la hora de definir un camino, debido al hecho de que solo necesita vértices que no intercepten algún sólido. Los caminos escogidos en esta ocasión son la conjetura de prácticamente todos los vértices, siendo que se puede obtener un camino casi idéntico con menor cantidad de puntos medios, como por ejemplo en el camino resaltado en azul (el desfase angular producido al retirar el segundo y cuarto punto es poco significativo y aun así se conservaría una ruta espaciosa para el robot. No obstante, la disposición de dichos obstáculos determina, por ejemplo, si el camino resultara largo, corto, muy curvilíneo, etc. Un fenómeno parecido ocurre con los mapas de ocupación, con la diferencia de que no está sujeto estrictamente a las dimensiones de los obstáculos, por lo que hay mayor libertad de formar un camino más conveniente (de acuerdo al criterio del usuario).

Si de restricciones se trata, los diagramas de Voronoi ofrecen precisiones espaciales a través de las aristas de las subregiones, convirtiendo los vértices vecinos en puntos de conexión. Por más conveniente que esta técnica pueda parecer, elementos como la forma de los diagramas, la imposibilidad de modificar sus dimensiones, o la paradoja del vértice infinito limitan la versatilidad que pudiera tener, conllevando así a una

posible eventualidad que impida un desempeño óptimo (en el camino resaltado en verde, la llanta izquierda del robot estuvo a milímetros de rozar el obstáculo).

Independiente de sus características, todas las técnicas cumplen con el objetivo de llegar a la meta rápida y fluidamente, sin choques contra los obstáculos y/o las paredes del circuito. Sin embargo, hay que resaltar que la mejor técnica para este caso es diagramas de Voronoi porque tratándose de un campo de obstáculos estáticos, construir las celdas (acomodando el centroide de la celda) proporciona caminos fluidos.

Tabla 1. Tabla de resultados de técnicas locales.

Técnica	Tiempo de acción	Velocidad de reacción	Manejabilidad	Sentido de orientación
Red neuronal	Primero intento: 11:35.88 Segundo intento: 04:55:23 Tercero intento: 04:35.66 Cuarto intento: 02:44.95	Media: recorrido turbulento, choques con los obstáculos fueron recurrentes y aparatosos.	Pobre: choques frecuentes contra las paredes del circuito, enganche de las llantas con los obstáculos, estancamiento en las esquinas.	Medio: ligeras desviaciones. Los giros realizados encaminaban al robot de manera adecuada.
Lógica difusa	Primero intento: 20:20.25 Segundo intento: 23:44.92 Tercero intento: 14:16.34	Media: choques contra los obstáculos menos frecuentes.	Pobre: choques frecuentes contra las paredes del circuito enganche de las llantas con los obstáculos, estancamiento en las esquinas	Medio: el robot se mantuvo en ruta hacia la meta, salvo cuando un mismo obstáculo fue evadido varias veces.
Campos de potencial	Primero intento: 14:19.57 Segundo intento: 09:40.80 Tercero intento: 06:41.41 Cuarto intento: 05:36.55	Rápida: distención para con los obstáculos (fuerzas repulsivas) inmediata después del cálculo respectivo.	Media: capacidad de salir de atascos con considerable velocidad de reacción.	Medio: dirección a la meta alterada algo notable, con rencuentro oportuno del camino a la meta.

VII. CONCLUSIONES

La cualidad de las técnicas de navegación local permite que el robot explore distintos campos, alcanzando su objetivo de variadas formas. Pero dicha libertad se ve coartada con situaciones propias de cada técnica. Como se mencionó en los resultados, los locales mínimos causan un estancamiento en la función de la lógica difusa, fenómeno que se puede apreciar también en la red neuronal, diferenciándose éste último en una precipitación (afortunada) hacia la meta.

Las técnicas globales resultan una buena elección en cuanto a rapidez para alcanzar el objetivo. Al estudiar el campo con antelación, se puede obtener una ruta en donde los obstáculos estén lo más alejados posibles del robot. Esta evasión impecable favorece grandemente el rendimiento, logrando un aminoramiento de los gastos energéticos del robot y una conclusión rápida a su recorrido.

Tabla 2. Tabla de resultados de técnicas globales.

Técnica	Tiempo de acción	Velocidad de reacción	Manejabilidad	Sentido de orientación
Diagramas de Voronoi	1er camino: 00:17.92 2do camino: 00:19.28	Rápida: evasión sagaz de los obstáculos cuando se acerca.	Alta: el robot se mantiene en ruta hacia la meta, con desfases angulares despreciables.	alto
Grafos de visibilidad	1er camino: 00:15.17 2do camino: 00:18.57 3er camino: 00:16.25 4to camino: 00:19.31	Rápida: evasión sagaz de los obstáculos cuando se acerca.	Alta: el robot se mantiene en ruta hacia la meta con desfases angulares despreciables	alto
Mapas de ocupación	1er camino: 00:14.01 2do camino: 00:17.36 3er camino: 00:19.48 4to camino: 00:15.83	Rápida: evasión sagaz de los obstáculos cuando se acerca.	Alta: el robot se mantiene en ruta hacia la meta con desfases angulares despreciables	Alto

Determinar qué estrategia utilizar depende principalmente de las condiciones de trabajo y en la cantidad de información que se tenga del ambiente. Tratándose de un ambiente bien conocido, cuyas características estén establecidas y se haya podido evaluar con anterioridad las medidas a tomar, la elección de entrada más adecuada tiende a las estrategias de navegación global. En cambio, conociendo únicamente la información suministrada por los sensores que se encuentra a bordo del robot, la opción más acorde resulta ser las estrategias locales.

REFERENCIAS

- [1] GONZALES F Alfonso, BARRERO Jaime, Algoritmos genéticos aplicados al planeamiento de trayectorias de robots móviles. Modelado y simulación. Grupo CEMOS, Escuela de Ingenierías Eléctrica, Electrónica y Telecomunicaciones E3T, Universidad Industrial de Santander.
- [2] Newman P. A., Kempf K.G. (1.985) "Opportunistic Scheduling for Robotic Machine Tending". Proc. of Second Conference on Artificial Intelligence Applications. Pp 168-175 .
- [3] Levi P. (1.987) "Principles of Planing and Control Concepts for Autonomous Mobile Robots". Proc. of 1987 IEEE International Conference on Robotics and Automation. pp 874-881.
- [4] Wein R., van den Berg J.P., and Halperin D. The visibility-Voronoi complex and its applications, Computational Geometry: Theory and Applications 36 (2007), no. 1, 84-87.
- [5] Benavides, F. (2012). Planificación de movimientos aplicada en robótica autónoma móvil. Montevideo: Universidad de la República.
- [6] Nilsson N.J. (1.969) "A Mobile Automaton: An Application of Artificial Intelligence Techiques". Proc. of the 1st. International Joint Conference on Artificial Intelligence. Pp 509-520.
- [7] Esteve, J.D. (2001) "Apuntes de Robótica". Universitat de Valencia. Pp 7/13-7/14.
- [8] Khatib, O. (1986). "Real-time obstacle avoidance for manipulators and mobile robots". In: The International Journal of Robotics Research. 5, (1), pp 90-98.
- [9] Borestein, J. & Koren, Y. (1991). "The vector field histogram: fast obstacle avoidance for mobile robots". In: IEEE transaction on robotics and automation, 7(3), pp 278-288.
- [10] Beaufrere, B. and Zeghloul, S. A mobile robot navigation method using a fuzzy-logic approach. Robotica. 1995, vol-13, no-pt5, 437-448.
- [11] Parhi, D. R. (2008). Neuro-Fuzzy Navigation Technique for Control of Mobile Robots, Motion Planning, Xing-Jian Jing (Ed.), ISBN: 978-953-7619-01-5, InTech. Pp 415-416.
- [12] MARCOS M, GUZMAN R, ROGRIGUEZ R., Autoguiado de robots móviles mediante redes neuronales. XXV Jornada de automática. 2004.
- [13] Berrio, J. S. (2009). Navegación planificada de un robot móvil. Cali: Universidad Autónoma de Occidente.



S. Cantillo. (Barranquilla, Colombia; 30 de diciembre de 1991). Egresado del Colegio Alemán de Barranquilla, título obtenido: bachiller. Cursos o talleres realizados: Técnico de Electrónica de la Fundación Educativa del Caribe Actualmente es estudiante de Ingeniería Mecatrónica de la Universidad Autónoma del Caribe.