BENEMERITA UNIVERSIDAD AUTÓNOMA DE PUEBLA.

FACULTAD DE CIENCIAS DE LA COMPUTACIÓN.

Tesis presentada para obtener el grado de:

Licenciado en Ciencias de la Computación.

“Control de un robot móvil para mapeo con lógica difusa”

Presenta:

David Luna Ramos.

Director de Tesis:

Dr. Gustavo Trinidad Rubín Linares.

Puebla, Pue Abril del 2021.

Agradecimientos.

Primeramente, agradezco a la Benemérita Universidad Autónoma de Puebla por permitirme ser parte de la institución también darme acceso a las puertas de su seno científico para lograr estudiar mi carrera, del mismo modo a los docentes que me brindaron sus conocimientos de igual manera su apoyo para seguir adelante día a día.

Agradezco a mi asesor de tesis al Doctor Gustavo Trinidad Rubín Linares por brindarme la oportunidad de recurrir a su capacidad por otro lado de su conocimiento científico, así también a su dedicación en su trabajo hay que mencionar, además haber tenido toda la paciencia del mundo para guiarme durante todo el desarrollo de la tesis.

Dicho lo anterior, también agradezco a mis compañeros de clase en cada nivel de la Licenciatura, gracias a su compañerismo, amistad más aun apoyo moral que ha sido un alto porcentaje a mis ganas de seguir adelante en mi carrera profesional.

Dedicatoria.

Dedico con todo mi corazón mi tesis a mi madre, pues sin ella no lo había logrado. Tu bendición a diario a lo largo de mi vida me protege además me llevó por el camino del bien. Por eso te doy mi trabajo en ofrenda por tu paciencia así mismo amor madre mía, te amo.

María Ocotlán Ramos Mancilla (q.e.p.d.)

Contenido.

Relación de ecuaciones.

Relación de tablas.

Relación de figuras.

Resumen.

En este trabajo de investigación se describe el proceso para lograr el control de un robot móvil **Scribbler®** con lógica difusa. Al inicio se estableció el marco de teórico en el cual se utilizaron la inteligencia artificial y lógica difusa Posteriormente se estudian la estructura del robot **Scribbler®**, ademas se desarrollo un simulador para este proyecto, en el cual se implementaron los algoritmos en amplitud y primero el mejor; así mismo se utilizaron estos algoritmos al ser de búsquedas sin información así mismo con información del SLAM. Paso siguiente, se postula una poda de los nodos cercanos a los objetos del mapa utilizando una función triangular. Posteriormente se utilizan las variables independientes distancia y velocidad para crear un par de conjuntos difusos que se implementan en el modelo de Sugeno [1], se obtiene la variable dependiente frenar, finalmente se realizó el proceso del controlador difuso para desarrollar un algoritmo de búsqueda con lógica difusa.

Capítulo 1.

Introducción.

1.1. Introducción.

Inicialmente los robots móviles tienen como precedente los dispositivos electromecánicos; tales como el denominado “micro mouse”, creado para desarrollar funciones como descubrir caminos en laberintos. Asimismo, este tipo de dispositivos eran guiados por cables bajo el suelo, con la finalidad de seguir rutas establecidas dentro de una planta de producción [2]. Alrededor de los años setentas aparecieron robots móviles dotados de autonomía, basados en sistemas de visión [3]. Por otro lado, se desarrollaron dispositivos móviles para interiores, este tipo de dispositivos fueron desarrollados por Bares [4]. Mientras la autonomía de un robot se basaba en sistemas de navegación automática, centrándose en la planificación, percepción también del control. Más tarde se postularon las bases de la exploración que consiste en la planificación de trayectoria, con la finalidad de evitar obstáculos no esperados. Después, los conceptos de localización por otro lado mapeo se formularon por primera vez en la conferencia de IEEE de robótica y automatización que tuvo lugar en San Francisco, California. USA en el año de 1986 [5]. Mientras se desarrollaron los conceptos del problema de mapeo también la acuñación del acrónimo SLAM (del inglés Simultaneous Localization and Mapping); estos conceptos se presentaron por primera vez en un panfleto sobre robots móviles en el simposio internacional que tuvo el nombre de localización de vehículos guiados de 1995 [6]. Posteriormente se alcanzó un acercamiento entre el SLAM basado en el filtro Kalman así mismo los métodos probabilísticos para la localización y mapeo introducidos por S. Thurn, en la conferencia internacional de robótica y automatización (ICRA) del Institute of Electrical and Electronics Engineers (IEEE) de 1999. Por consiguiente, se atrajo a 15 investigadores que buscaban soluciones para la complejidad, la asociación de la información por otro lado en retos de la implementación [7]. Por último, en el año de 2012 se ha utilizado el Kinect (sistema de video

juego) en el SLAM, se utilizó este dispositivo en la localización igualmente generación de mapas del entorno de un robot móvil, este proyecto de investigación se construyó en la Universidad Politécnica de Valencia [8]. Por otra parte, el concepto de lógica difusa fue propuesto por el matemático Lofti Ali Asker Zadeh profesor de la Universidad de California, en Berkeley [9], en el año de 1965 publica el primer artículo definiendo el concepto de conjunto difuso cuyo título fue “Fuzzy Sets” [10]. Alrededor del año de 1985 los científicos Takagi y Sugeno [1] logran aportar a la teoría del control difuso el método llamado Takagi, Sugeno también Kang (TSK), por el aporte de este método es una alternativa del método propuesto por Mamdani [11]. Por consiguiente, se ha logrado que la lógica difusa se aplique en el área del control automático. De hecho, se ha aplicado está rama en el frenado del metro de la ciudad de Sendai, Japón [12]. Del mismo modo la lógica difusa se ha aplicado en el análisis de datos y en los métodos basados en el aprendizaje automático además reconocimiento de patrones, “clustering” así mismo en los algoritmos genéticos o en redes neuronales artificiales [13]. Solo entonces, estudiando los inicios del mapeo igualmente la lógica difusa; se investigará la aplicación de la lógica difusa en la inteligencia artificial. Por consiguiente, se determina que esta investigación es de carácter cuantitativo de observación experimental y fundamental [14].

En el primer capítulo se realizó un breve resumen de este trabajo de investigación, además se establece el marco teórico, en el segundo capítulo se plantea la inteligencia artificial, también se define esta ciencia, de igual importancia se establecen las bases del SLAM, asimismo se investigan e implementan los algoritmos en amplitud del mismo modo el de primero el mejor en un simulador [15]. De igual importancia en el capítulo tercero se describe la lógica difusa, después de esto en el capítulo cuatro se implementan los capítulos anteriores; se parte del robot móvil se describen el sistema de frenado del móvil, a posteriori se describe el simulador que se

desarrolló con la finalidad de implementar los algoritmos de búsqueda de amplitud asimismo primero el mejor [16]. Posteriormente en el capítulo quinto se observa la aplicación efectiva de los conjuntos difusos, los cuales se implementan en Matlab, además se utilizan los resultados del controlador difuso como base para desarrollar un nuevo algoritmo de búsqueda con lógica difusa. En las conclusiones se describe el proceso de investigación así mismo se mencionan los hallazgos que se encontraron en esta investigación, de igual forma se plantea el trabajo actual del mismo modo trabajo futuro de este proyecto. Consecuentemente a lo largo de la investigación documental por otro lado práctica, se puede tomar este trabajo de estudio para dar sustento a la ciencia llamada inteligencia artificial.

Capítulo 2.

Marco teórico de la inteligencia artificial.

En este capítulo se describen los conceptos de la inteligencia artificial que se utilizaron para desarrollar de este trabajo de investigación, inicialmente, se describen las técnicas que se aplican en un robot móvil para lograr realizar la exploración de un mapa; también se define la inteligencia artificial, por otra parte, se describe la teoría de exploración sin información además la exploración con información que es parte del SLAM, de igual importancia se describe el desarrollo de un simulador para este trabajo; finalmente se describen por otra parte se implementan los mejores algoritmos de búsqueda con información además sin información los cuales son probados en el simulador.

2.1. ¿Qué es la inteligencia artificial?

John McCarthy organiza una conferencia en Dartmouth, en la cual se acuñada por primera vez el concepto de inteligencia artificial, por otra parte, se define como la ciencia de hacer máquinas inteligentes [17]. Recientemente se ha considerado esta ciencia como la base de la tercera revolución industrial formulada por Jeremy Rifkin [18] que es el antecedente de lo que se ha denominado como industria 4.0. En otras palabras, la inteligencia artificial consiste en la aplicación de las ciencias de la computación con la finalidad de ser implementadas en el software además en el hardware de un dispositivo con la finalidad de lograr su autonomía.

2.2. SLAM.

En forma general el problema de localización y construcción de mapas en simultáneo (del inglés simultaneous localization and mapping, en adelante SLAM) se basa en descubrir si es posible para un robot móvil navegar a través de un entorno desconocido, además, lograr construir de manera incremental un mapa determinando al mismo tiempo su posición. A la vez los trabajos que han ayudado en la solución de este problema se han considerado “el santo grial” por los especialistas en la robótica móvil, por consiguiente, dotaría a los robots con las herramientas

necesarias para lograr la autonomía. Por otra parte, esta técnica ha sido formulada del mismo modo ha logrado resolver varios aspectos desde la parte teórica. Asimismo, se ha implementado en diversos campos de la robótica móvil, como en los robots de interiores, en exteriores, robots subacuáticos del mismo modo en robots voladores. Además, en un nivel teórico además conceptual, el mapeo se puede considerar un problema resuelto. Sin embargo, aún quedan varios problemas para lograr aplicar soluciones generales de la exploración. Por esto se está trabajando en la utilización del mismo modo en la construcción de mapas como lugar de aplicación del algoritmo de localización adempas mapeo simultaneo [19].

Historia del problema del SLAM.

El origen del SLAM se dio en la conferencia sobre robótica y automática del IEEE, en la ciudad de San Francisco, en 1986 [4]. En primer lugar, los métodos probabilísticos fueron introducidos en la robótica además en la inteligencia artificial. De la misma forma en esta conferencia un grupo de asistentes mantuvieron una conversación sobre la aplicación de métodos de estimación a los problemas de mapeado y localización. Mejor dicho, se reconoció que el mapeado probabilístico era un problema fundamental de la robótica el cual merecía ser tratado. Por lo tanto, el descubrimiento más importante del mapeado y la localización, fue formulado como un único problema de estimación, el cual era de naturaleza convergente. En otras palabras, se reconoció las correlaciones entre los objetos de referencia, que otros científicos intentaban minimizar o eliminar, siendo la clave del problema [19].

Mapeado.

El objetivo principal del mapeado es representar la información adquirida del entorno, que se utilizará para lograr planificar la navegación igualmente la resolución de tareas. También existe el enfoque topológico en el cual se consideran “lugares” de igual importancia las conexiones

entre ellos. De igual importancia los mapas se representan en forma de grafos, así también los nodos se utilizan para representar distintas posiciones alcanzables por el robot además las aristas, siendo el camino de acceder desde un nodo hacia otro nodo. Mejor dicho, las coordenadas de los posibles nodos visitados son elegidas con precisión. Ésta es la representación preferida al resolver el problema del SLAM. Al utilizar este esquema se logran las siguientes ventajas: El algoritmo es robusto del mismo modo su implementación se consigue de forma sencilla. No obstante, se presenta la incertidumbre de la geométrica de los elementos presentes en el entorno. Ya que se alcanza distinguir entre zonas ocupadas de manera análoga vacías, consiguiendo una partición además la descripción completa del espacio explorado. Por este motivo es popular en tareas de navegación facilitando la planificación igualmente generación de trayectorias empleando métodos convencionales. Como es lógico, esto va en detrimento del rendimiento computacional del algoritmo. Así mismo se permite una extensión conceptualmente simple al espacio tridimensional en la zona que se va a explorar. Después de todo es estimada la posición del robot, por consiguiente, se puede elegir la estrategia para explorar el camino que habrá de seguir el robot durante su exploración. Mejor dicho, se conoce a este problema como “planificación de movimientos” o “problema de navegación” [19].

Navegación.

Una forma básica de representar la forma de conectar una posición inicial por otra parte la posición final, evitando colisionar con los obstáculos que existan en el entorno (paredes, muebles, escaleras, etc). Por consiguiente, existen casos de complejidad espacial, también la investigación exacta de la planificación de caminos puede considerarse con costos computacionales altos. En esta problemática, se han creado algoritmos basados en el muestreo, los cuales han demostrado ser bastante eficaces. A la inversa no es posible detectar

los casos en los que no existe un camino entre el inicio además del final, por consiguiente, la probabilidad que tienen estos algoritmos de fallar, decrece al invertir demasiado tiempo en la navegación. Eventualmente los algoritmos de muestreo se consideran extremadamente buenos en lograr la planificación de movimientos en espacios multidimensionales, por otra parte, se han aplicado a problemas con cientos de dimensiones (manipuladores robóticos, moléculas biológicas, personajes animados digitales, robots humanoides) [19]. En otras palabras, al haber planteado el mapeo asimismo la navegación podremos centrar este trabajo de investigación en la exploración de un mapa. En realidad, la navegación se clasifica en búsqueda sin información también búsqueda con información [19]. Con todo esto para poder realizar el mapeo es necesario tener presentes las siguientes definiciones:

Estado: Es la representación de un problema en cualquier instante. Asimismo, se debe definir el conjunto de todos los estados, de la misma forma no es necesario realizar una completa enumeración de todos los estados válidos, lo más adecuado es definirlo de manera general.

Estado inicial: Lugar inicial para realizar la exploración del mapa.

Estado objetivo: Consiste en una o varios metas que se consideran solución aceptable.

Reglas: Son las acciones que se deberán aplicar a un conjunto de estados. Igualmente, una regla se desarrolla en dos partes: Inicialmente se determina la forma en que se aplicará la regla, es decir, se describen los estados a los que se podrá aplicar la regla. Posteriormente se describe el proceso que se llevará a cabo.

Heurística: Es el conjunto de técnicas para resolver un problema. En primer lugar, su origen es griego **εὑρίσκειν** que significa “hallar, inventar”. Por lo tanto, es el arte de inventar por parte de los seres humanos, con la intención de procurar estrategias, métodos, que permitan resolver problemas a través de la creatividad [20].

Camino Solución: Es el árbol dirigido de los nodos que se deben de seguir desde el inicio hasta la meta. También se han aplicado en diversos problemas de juegos; como son la solución del cubo Rubik, en problemas de navegación de robots, también secuenciación automática de ensamblaje, en la búsqueda en el internet [19].

A continuación, se describe el proceso mediante el cual se aplicarán los conceptos anteriores en el juego de rompecabezas de 8 piezas.

Descripción del juego: El siguiente punto trata del rompecabezas de 8 piezas el cual se puede observar en la figura uno, además consta de un tablero de 3 x 3 con ocho bloques numerados y un espacio en blanco. Del mismo modo tiene un bloque adyacente que es un espacio en blanco, el cual se podrá utilizar para desplazar los bloques [21].

Se describe el juego a continuación:

- **Espacio de estados:** Ubicación de cada uno de los ocho mosaicos y el espacio en blanco.
- **Estado inicial:** Cualquier estado puede ser designado como estado inicial. Además, se establece que cualquier meta, se podrá alcanzar desde los posibles estados iníciales.
- **Estado final:** Se establece al inicio del juego, el cual se muestra en la figura uno.
- **Reglas:** Una vez que se establece el conjunto de estados permitidos se deberá de probar los movimientos permitidos que son blanco, izquierda, derecha, arriba o abajo.
- **Heurística**: Son los lugares donde se puede mover cada pieza [20].
- **Camino solución:** Secuencia de estados generados desde el estado inicial al estado final.

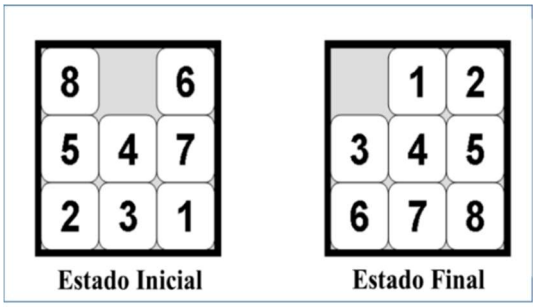


Figura 1. Ejemplo de rompecabezas de 8.

El rompecabezas 8 pertenece a la familia de problemas de búsqueda, en la tabla uno, se muestran el tipo de juegos que se pueden resolver mediante esta técnica, de igual forma los números de estados asimismo el tiempo de solución.

Tabla 1. Problemas de prueba de algoritmos.

Dominio	Número de estados	Tiempo (10^7 nodos/s)
8-puzzle	$\left(\frac{N^2!}{2}\right)\Big\vert_{N=3} = 181{,}440$	0.01 segundos
15-puzzle	$\left(\frac{N^2!}{2}\right)\Big\vert_{N=4} = 10^{13}$	11,5 días
24-puzzle	$\left(\frac{N^2!}{2}\right)\Big\vert_{N=5} = 10^{25}$	$31,7 \times 10^9$ años
Hanoi (3,2)	$(3^n)\vert_{n=2} = 9$	9×10^{-7} segundos
Hanoi (3,4)	$(3^n)\vert_{n=4} = 81$	$8,1 \times 10^{-6}$ segundos
Hanoi (3,8)	$(3^n)\vert_{n=8} = 6561$	$6,5 \times 10^{-4}$ segundos
Hanoi (3,16)	$(3^n)\vert_{n=16} = 4,3 \times 10^7$	4,3 segundos
Hanoi (3,24)	$(3^n)\vert_{n=24} = 2,824 \times 10^11$	0,32 días
Cubo de Rubik $2 \times 2 \times 2$	10^6	0,1 segundos
Cubo de Rubik $3 \times 3 \times 3$	$4,32 \times 10^{19}$	31.000 años

2.3. Estrategias y algoritmos de búsqueda.

A su vez para la solución de un problema aplicando el mapeo, se pueden utilizar las estrategias de búsqueda que se clasifican en búsqueda desinformada asimismo búsqueda informada o heurística.

Búsqueda desinformada.

Por otra parte, se utiliza el término de búsqueda desinformada, en particular en esta búsqueda no se cuenta con información adicional sobre los estados que se proporciona en la definición

del problema, considerando que consiste en generar sucesores asimismo distinguir un estado objetivo de un estado [19]. Brevemente la búsqueda sin información del domino, también llamada exhaustiva es:

- **Sistemática. -** No deja sin explorar ningún nodo además lo explora sólo una vez.
- **Objetiva. -** Pues no depende del dominio del problema.

En resumen, estas técnicas son poco eficientes, tiene un coste promedio, asimismo la generación del camino solución, también se caracterizan por la aplicación de los estados que se realizan de manera sistemática, tienen como estrategia de control no informada, explorar los nodos uno detrás de otro estado [19]. Son búsquedas por tentativas de la misma forma utilizan una búsqueda en el espacio de estados, además tienen como criterio para diferenciarlas, se utilizará el orden que se obtiene para recorrer los estados. A continuación, se describen los estados igualmente operaciones de las búsquedas:

- **Espacio de estados:** Los nodos donde se puede realizar la exploración del mapa.
- **Estado inicial:** Es la posición inicial donde se colocará el móvil dentro del mapa.
- **Estado final:** Es la meta del móvil, se especifica desde el inicio de la exploración.
- **Reglas:** Son los movimientos que el móvil puede realizar.
- **Heurística:** Se generarán tres nodos donde se puede desplazar el móvil [20].
- **Camino Solución:** Conjunto de nodos que se deben de recorrer para lograr el estado final.

La búsqueda en amplitud denominada también en anchura la cual se puede estudiar en la figura tres como se ha señalado es una estrategia sencilla en la que se expande primero el nodo raíz, posteriormente se realiza la tarea de expandir todos los sucesores del nodo raíz, también cada uno de sus sucesores, finalmente se genera el conjunto de nodos que guiara al estado final. En concreto, se expanden todos los nodos a una profundidad I en el árbol de búsqueda antes de

expandir cualquier trayectoria del próximo nivel. En efecto, no se expanden nodos del nivel I+1 hasta que no se hayan expandido todos los estados del nivel I. A su vez el proceso termina con error si no hay más nodos que expandir sin encontrar una meta [19]. En segundo término, la búsqueda primero en amplitud se puede implementar utilizando una estructura de tipo cola (Glosario), se le denominará abierta donde se almacenarán todos los estados abiertos. De igual forma cada registro en esta estructura quedara almacenado el estado del mismo modo un apuntador a su padre. Por otra parte, las características del nodo, asegurándose que los primeros nodos visitados serán los primeros en expandirse. La principal desventaja de la búsqueda en anchura es el requisito de memoria para almacenar todos los nodos que no han sido expandidos durante la búsqueda. Por ejemplo, se muestra el árbol de búsqueda en la figura dos, donde se puede observar la evolución del algoritmo, indicado por las flechas.

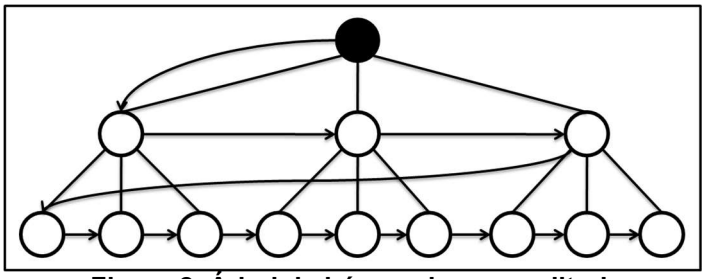


Figura 2. Árbol de búsqueda en amplitud.

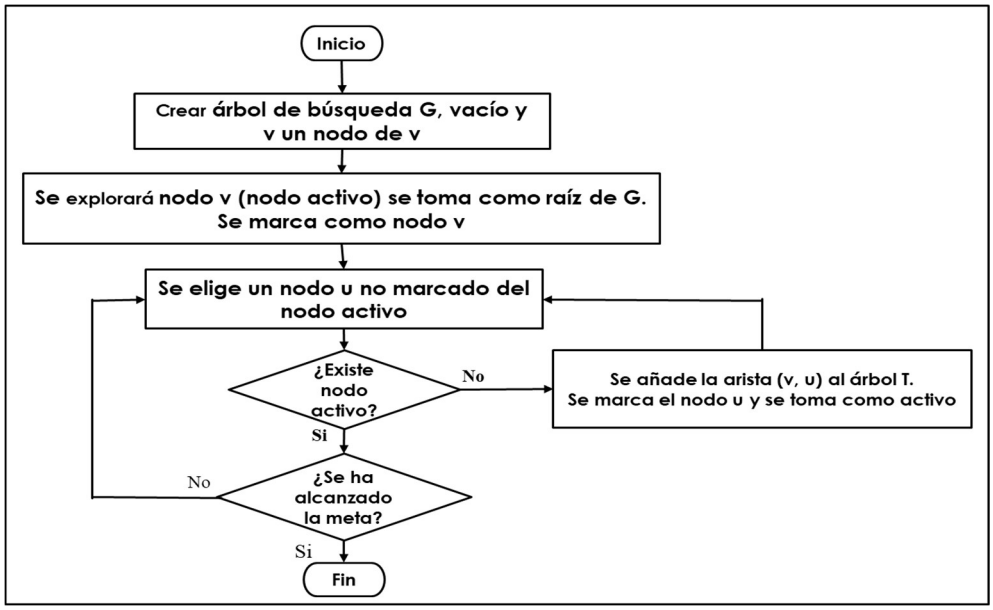


Figura 3. Algoritmo de búsqueda en amplitud.

Completo. - El objetivo de esta estrategia es encontrar el camino solución si hay una solución.

Optimo. - Hemos dicho si hay solución, esta estrategia la encuentra; pero además de menor costo, si consideramos que los operadores tienen el mismo costo unitario del mismo modo en todo caso la solución con el menor número de operaciones (la de menor nivel).

Complejidad temporal. - El tiempo de ejecución dependerá del factor de ramificación además de profundidad p, el tiempo necesario será $1 + i^1 + i^2 + i^3 + \ldots + i^p$, el coste temporal está en **$O(i^p)$**.

Complejidad espacial. - Al terminar la expansión de los estados de un nivel de profundidad (p-1) se deberán almacenar en la estructura abierta, todos los estados del nivel p. Además, si el número de operadores es n asimismo estamos a profundidad p, tendremos almacenados i^p estados. En realidad, el peor de los casos es cuando la meta este en el último estado de un nivel p. Así el costo espacial está en **$O(i^p)$**.

Búsqueda informada o heurística.

En primer lugar, la búsqueda informada consiste en añadir información, basándose en el espacio estudiado hasta ese momento, de igual forma con esto se logra restringir drásticamente

esa búsqueda [19]. También tiene una función de evaluación heurística [20]. Además, se utilizan las estrategias de control para realizar este tipo de búsqueda, que se denominará fev. Se definen los siguientes conceptos:

- **ℜ:** Conjunto de números reales.
- **Estados:** Nodos en los que se realiza la exploración.
- **Estado j:** Es el nodo donde se está realizando la exploración.
- **f:** Función de evaluación, mide la calidad del nodo.
- **nj:** Es el número de nodos que la función genera.

Son la aplicación del conjunto de espacio de estados en ℜ.

f: {estados} →ℜ tal que **f(Estado j)** = **nj**.

El valor de esta función depende exclusivamente del estado que se está evaluando, es decir, el valor de la función se basa en la información disponible en el momento de la búsqueda. Por regla general, el sistema experto almacenará esta información mediante el aprendizaje. Además, este valor numérico realiza una estimación de lo bueno que puede ser el estado para llegar a la meta. En concreto esta estimación, suele ser de mayor coste pero que también puede ser de otro parámetro [19].

A continuación, se describen estos dos valores:

- **Coste:** El valor mínimo o máximo de la función, se obtiene al encontrar la meta.
- **Parámetro:** Se pretende que el valor calculado, por esta función sea óptimo durante todo el camino. En este caso se dice que la **fev** es óptima, por tanto, se obtendrá la mejor solución: la búsqueda se simplifica a una secuencia de acciones. Acorde con el caso real no se sabe cuál es el camino óptimo, se pretende encontrar la meta.

Hay que tener en cuenta que la **fev**, aunque utilice el conocimiento del dominio para el cálculo de su valor, forma parte del solucionador también de las tareas de búsqueda; las que se usan en los métodos de búsqueda no informada. Encontrar una **fev** óptima podría ser fácil si no fuese por un detalle: su cálculo también tiene un coste asociado, puede darse el caso de que el coste del algoritmo con esa **fev** sea mayor que el coste del algoritmo sin ella. Para identificar una **fev**, lo que hacemos es identificar todas las restricciones que ha de cumplir el problema. Posteriormente simplificamos el modelo matemático, es decir, relajamos estas restricciones, de manera que se convierte en el mismo problema, pero menos estricto. También se consigue que la **fev** sea más simple, pero lo suficientemente útil para determinar que el estado nos sirva para llegar a la meta en función de las restantes restricciones, con la finalidad de poder simplificar el modelo, se analizará al dividirlo en sub problemas más sencillos [19]. Por supuesto que un problema el valor calculado por una **fev**, que estima lo bueno que es el estado para llegar a la meta se toma en cuenta todas las restricciones, ya que va a ser mejor que el calculado por la **fev** que estime lo mismo, pero cumpliendo solo algunas restricciones. Como sucede cuanto más simple sea el modelo relajado, menor costo tendrá la **fev**, al ser más simple, en contrapartida, dirigida como peor búsqueda. Así mismo una **fev** que solo mida un parámetro del problema, además de ser demasiado simple para estimar la distancia a la meta, también puede producir otros problemas [19].

- **Máximos (mínimos) locales. -** Es un estado que se estima como el mejor en el conjunto de estados que puede explorar el móvil, pero que se estima peor que alguno que está más alejado.
- **Altiplanicies o mesetas. -** Son un conjunto de estados que tienen el mismo valor de fev, por lo que no se tiene información del camino que se deberá de seguir.

Estos inconvenientes se pueden solventar, aunque no definitivamente, con **fev´s** que midan más de un parámetro por otro lado eligiendo adecuadamente el método de búsqueda. Además, las funciones heurísticas [20] son la forma más común de transmitir el conocimiento adicional del problema estudiado. Otra vez la búsqueda primero el mejor, trata de expandir el nodo más prometedor, alegando que probablemente conduzca rápidamente a una solución. En otras palabras, esta técnica pertenece a las estrategias de búsqueda informada, ya que utiliza conocimiento específico del problema, por tal razón se puede encontrar soluciones de manera más eficiente que una estrategia de búsqueda no informada. Asimismo, la búsqueda voraz primero el mejor, se parece a la búsqueda primero en profundidad, a su vez prefiere seguir un camino hacia el objetivo, pero volverá atrás cuando llegue a un callejón sin salida. También la búsqueda primero el mejor selecciona un nodo para la expansión; por otra parte, se selecciona el nodo con la evaluación más baja, porque la evaluación mide la distancia estimada hasta la meta. En particular se puede implementarse con una estructura de datos tipo cola con prioridad (Glosario), los cuales son ordenados por su valor de evaluación en orden ascendente [21]. Sin embargo, el nombre de esta técnica parece incorrecto, debido que, al expandir primero el mejor nodo se realizaría un conjunto solución directo por que se genera una búsqueda en absoluto. En otras palabras, al momento de que se elige el mejor nodo se tendría como consecuencia su valor mínimo será 0. Asimismo, esta distancia podrá ser el coste estimado a la meta, lo cual también puede medir cualquier otro parámetro. En efecto se busca primero un camino a la vez, pero puede cambiarse a otro camino que se tenga mejores características para encontrar la meta, del que se estaba siguiendo. De igual importancia se puede considerar que el algoritmo primero el mejor realiza la búsqueda en un grafo de tipo **O** (Glosario), por la razón de que todas las ramas dentro del árbol es una alternativa de solución el cual se puede investigar en la figura

cuatro. Además, para su operación el algoritmo necesita una lista de nodos además una función heurística [20] que a su vez la función evalúa los nodos que se generen. Como sucede en tomar sentido en la reordenación de la cola llamada abierta, debido a que se hace en función del valor calculado con la **fev**. Mientras el valor de la **fev** sea óptimo, la exploración será en profundidad, dado que tras ordenar la cola abierta esta va a tener siempre en la cima el mejor sucesor en la rama que estamos siguiendo. Si la rama que se está siguiendo se aleja de la meta, tenemos en la cola abierta otro estado con mejor estimación, la ordenación colocara este último nodo en la cima de la estructura de datos, tomándose entonces esta nueva rama. A su vez en la búsqueda en profundidad exhaustiva si llegábamos a un callejón sin salida o al límite de profundidad se retoma la exploración en profundidad desde otro estado hermano del predecesor al callejón sin salida. Ahora se toma el estado más prometedor de los que hay en abierta, como consecuencia podemos expandir un nodo en cualquier nivel así mismo una rama del árbol de exploración. Se debe comprobar si el nuevo estado generado está en las colas abierta o cerrada para lograr el estado más cercano a la meta [19].

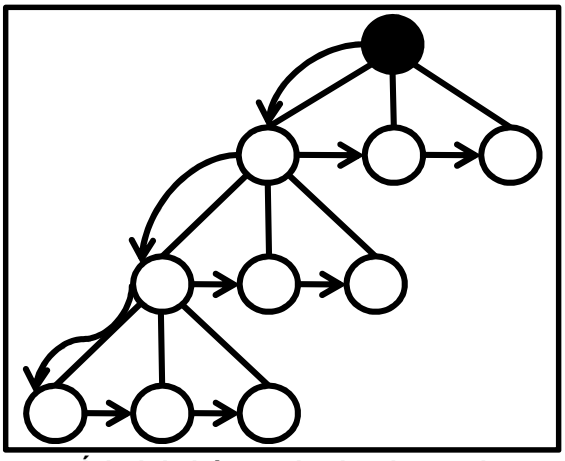


Figura 4. Árbol de búsqueda algoritmo primero el mejor.

Algoritmo primero el mejor.

Para lograr la implementación del algoritmo primero el mejor que está representado en la figura cinco, se definen los siguientes términos:

1. Abiertos: Es una variable que contiene los nodos que han sido generados. También la función heurística [20] ha sido aplicada para estos estados, pero aún no han sido examinados, es decir no se han generado sus sucesores abiertos, puede considerarse como una estructura de datos llamada cola (Glosario) con prioridad en la que los elementos con mayor prioridad son los que tienen los valores más prometedores, los cuales se obtienen sus valores por la función heurística.

2. Función heurística: Permite que el algoritmo busque primero por senderos que son más prometedores para muchas aplicaciones del mismo modo es conveniente definir esta función f', como la suma de dos funciones, que se les asignaran los nombres de g y h'. De nuevo la función g es una medida del costo de llegar desde el nodo inicial al nodo actual. De la misma forma la función h' es una estimación del costo adicional para llegar desde el nodo actual al estado objetivo. Aquí es donde se utiliza el conocimiento que se dispone sobre el dominio del problema. Es decir, la función combinada f' representa una estimación del costo de llegar desde el estado inicial hasta el estado final, siguiendo el camino que ha generado el nodo actual. Por otra parte, si el nodo actual ha generado más de un camino, el algoritmo deberá registrar sólo la mejor ruta. El algoritmo fue formulado para ser aplicado a las estructuras de datos de grafos. De la misma manera se ha trabajado en simplificarlo al implementarse en estructuras de datos llamadas árboles. A pesar de no realizar la comprobación del nuevo nodo en cola abiertos o en la cola de cerrados. Acelerando la generación de nodos también la búsqueda, para estos casos en la cual es poco probable que se repitan los nodos. Usualmente, el costo de ir de un nodo a su sucesor g, se fija en una constante igual a 1, cuando se desea encontrar un sendero a la

solución, que involucre el menor número de pasos. Si por el contrario nos interesa encontrar el camino de menor costo además algunos operadores cuestan más que otros, se asume un valor de g, que refleje estos costos. De igual forma un valor g igual a cero significará que simplemente nos interesa llegar a la solución, de cualquier manera. También si h' es un estimador perfecto de h, hará que converja inmediatamente al objetivo, sin búsqueda. Mientras mejor sea h', más cerca se estará de alcanzar esta aproximación directa. Por lo tanto, si h' vale cero, la búsqueda será controlada por g. También si el valor de g es cero, se realizará la búsqueda aleatoria. Asimismo, si el valor de g es siempre 1, hará que la búsqueda se realice primero en anchura. Mientras para los casos donde h' no sea perfecto ni cero, nunca llegará sobrestimado el valor de h, se garantiza que al realizar la búsqueda se logre un camino óptimo, en tal caso debe existir una solución, se concluye que es admisible [19].

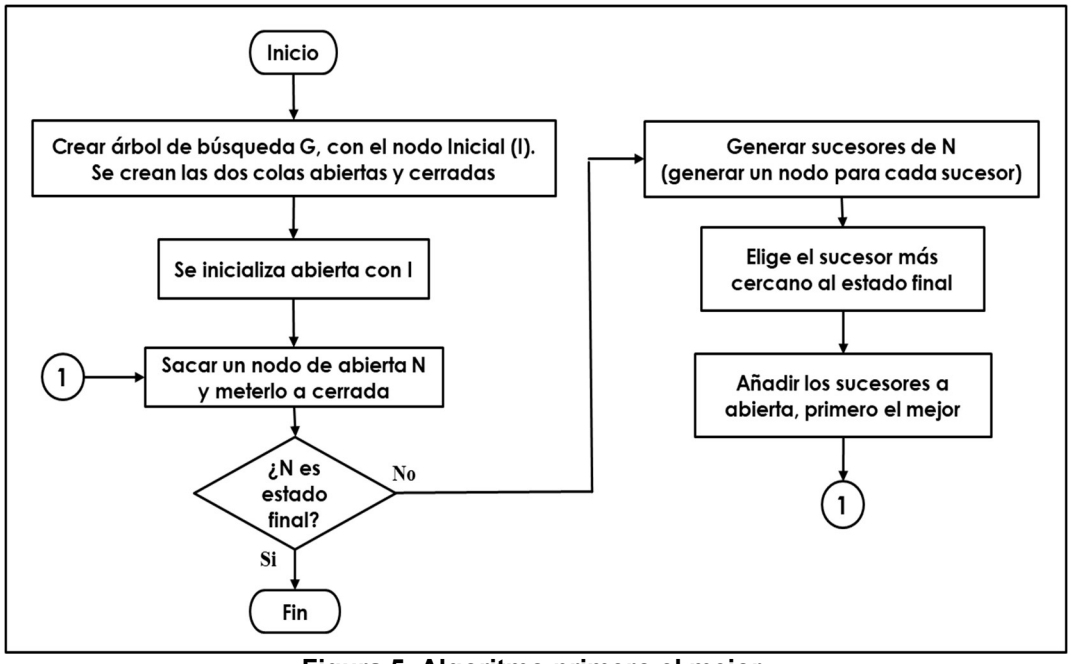


Figura 5. Algoritmo primero el mejor.

El algoritmo primero el mejor soluciona el problema de los mínimos locales además de las mesetas. Si el nodo llega a una de estas situaciones se permite que continuar con la búsqueda

además no permitir la repetición de estados, finalmente se saldrá de ellas. Por otra parte, en el mínimo local se logra continuar con el camino más prometedor asimismo en las mesetas se logra salir después de generar todos los estados de esta [19].

Completo. - Si el estado es meta, su **fev** son cero. Luego siempre encontrará una rama que tenga mejor fev además finalmente encontrará la meta, por lo tanto, se puede definir como completo.

Óptimo. - No es óptimo, pues la **fev** solo estima la cercanía a la meta por otra parte no se tiene en cuenta el coste del camino ya recorrido. Asimismo, se puede abandonar una rama cuya **fev** estima que está más lejos de la meta que otra, pero que el coste total del camino de la raíz a la meta sea inferior a esta segunda [19].

Supongamos el problema representado por el árbol de la derecha de la figura seis, donde los arcos del mismo modo además los nodos se etiquetan con sus costes asociados. Por consiguiente, se puede estudiar que el camino menos costoso de la raíz a la meta es A, C, D en el árbol de exploración de la derecha, asimismo al expandir la raíz se generan los nodos B también C, el siguiente nodo a expandir es B, pues es el que tiene el camino menos costoso a la meta. Al contrario, si lo que buscamos es el camino solución menos costoso, el siguiente nodo a expandir debería ser C, lo anterior se podría hacer si el algoritmo tuviese en cuenta el camino recorrido hasta B y C.

Coste computacional. - En el peor de los casos tendrá el mismo coste que la búsqueda en profundidad, ya que la solución podrá estar en el estado final de la última rama. Además, está en $\mathbf{O(n^p)}$. Mejor dicho, al ser implementada la **fev** el comportamiento promedio puede ser mucho mejor.

Coste espacial. - En la estructura abierta se logra tener almacenado el árbol de exploración, es decir en el último estado de la rama padre podremos observar los estados almacenados; también el coste estará en **O(n^p)**. Además, el coste computacional puede ser mucho mejor dependiendo de la calidad de la **fev** seleccionada.

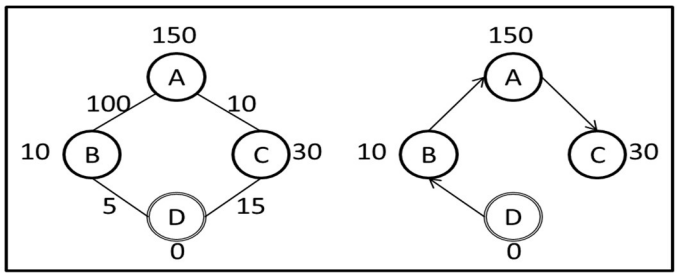


Figura 6. Árboles con diferentes costos en los recorridos.

Para acabar en el capítulo presente se planteó en el marco teórico la inteligencia artificial, de igual forma se define el SLAM, además las estrategias finalmente los algoritmos de búsqueda como elementos fundamentales para realizar el mapeo de una región.

Capítulo 3.

Lógica Difusa.

En este capítulo se describe la lógica difusa que se utilizó para el desarrollo de este capitulo, en primer lugar, se realiza la definición matemática del conjunto difuso, la cual es la base de la lógica difusa, de igual importancia se describe el proceso que se debe de seguir para generar un conjunto difuso finalmente se definen los controladores que existen.

3.1. Definición matemática del conjunto difuso.

Hablaremos de la lógica difusa, la cual es una alternativa que establece un grado de percepción, de igual forma en el mundo real existe mucho estudio no perfecto, es decir conocimiento perceptivo, el cual es incierto, a su vez es ambiguo o borroso por naturaleza. De manera análoga el razonamiento a su vez el pensamiento humano se basa en la percepción [16]. No obstante, en la lógica clásica los elementos tienen un grado de pertenencia él cual tiene un valor binario, en contraste al utilizar esta herramienta no se representan las características reales de un parámetro. En concreto supongamos que tenemos un conjunto de personas las cuales se clasificaran de acuerdo a su estatura; definiéndolos como bajos o altos. Además, todas las personas que midan 1.80 o más serán altas, en cambio las demás personas serán bajas. En otras palabras, según esta definición, alguien que mida 1.79 será tratado igual que otro que mida 1.50, a ambos individuos se clasificaron como personas bajas. Mejor dicho, si dispusiéramos de una herramienta para clasificar las alturas en forma suave, se representaría la realidad más fielmente [16]. Asimismo, no hay un valor cuantitativo que defina el término joven. De igual forma alguna gente, que tenga la edad de veinticinco años es joven, para otros con treinta y cinco años es joven. Para ilustrar un presidente de gobierno de treinta y cinco años es joven, mientras que un futbolista con la misma edad es viejo. Después de todo hay cosas que están claras: una persona de un año es joven, mientras que una de cien años es vieja. En este caso una persona de treinta y cinco años tiene algunas posibilidades de ser joven.

Asimismo, definiremos el conjunto joven de modo que cada uno de sus elementos pertenezca al grupo con cierto grado. De un modo más formal, un conjunto difuso se caracteriza por una función de pertenencia, la cual es representada en la ecuación uno:

$$\mu A : U \rightarrow [0,1] \qquad \text{Ecuación 1.}$$

La ecuación uno se asocia a los elementos x de U, también un número μ A(x) en el intervalo [0,1], en particular representa el grado de pertenencia de x al conjunto difuso A. Del mismo modo U se le llama universo de discurso. Así que el término difuso joven se define mediante su respectivo conjunto difuso representado en la tabla dos.

Tabla 2. Conjunto de hombre joven.

Edad	Grado de pertenencia.
≤25	1.0
30	0.8
35	0.6
40	0.4
45	0.2
≥50	0.0

Es decir, la función de pertenencia del conjunto difuso joven viene dada por la ecuación dos:

$$\mu A(x) = 1 \text{ si } x \leq 25,\ \mu A(30) = 0.8,\ ....,\ \mu A(x) = 0 \text{ si } x \geq 50. \qquad \text{Ecuación 2.}$$

Si el universo de discurso es continuo, entonces tendremos la función de pertenencia continua que se pueden ver sus valores en la figura siete.

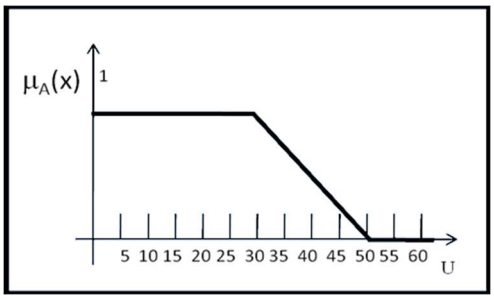


Figura 7. Función de pertenencia de joven si U es continuo.

En consecuencia, en la función de pertenencia se especifican los valores del conjunto discreto que pertenece al universo de discurso, también el valor asociado al resto de los elementos se

puede obtener por interpolación (utilizando la ecuación de la recta que unen los dos puntos) [22]. De modo que en esta obra se establecen los fundamentos de la lógica difusa, así también se llega a la definición de conjunto difuso habría que decir también sus propiedades.

3.2. Conjunto difuso.

Términos lingüísticos.

Otro punto es el proceso de utilizar una inconstante de investigación, que se utiliza para generar un conjunto difuso. Inicialmente convirtiéndolo en una función de pertenecía, consideremos que este proceso es fundamental en el desarrollo de este trabajo. Más tarde se deberán de tomar en cuenta los términos, se debe de realizar una adecuada asociación de los elementos para lograr una función de pertenecía, por consiguiente, se deberá de implementar la experiencia del mundo real. Aun cuando no existen reglas solamente las restricciones propias de la lógica difusa. Así, se elegirá la estrategia para establecer los acuerdos lingüísticos sobre el lenguaje, de ahí tendrán la cualidad de modificar la función de pertenencia, es decir los convenios lingüísticos se representarán las acciones que realizara el controlador, estas deberán ser representadas con valores como “casi”, “mucho”, “ampliamente”, etc [16].

En efecto se logra tomar la familia de funciones de la forma:

$$h_{\propto}(x) = x^{\propto} \qquad \text{Ecuación 3.}$$

De esta manera se pueden asociar valores de **α** determinados a las formas de matizarlos.

Por ejemplo:

$$\alpha(casi) = 2 \qquad \text{Ecuación 4.}$$

$$\alpha(mucho) = 6 \qquad \text{Ecuación 5.}$$

$$\alpha(extremadamente) = \frac{1}{2} \qquad \text{Ecuación 6.}$$

Para modificar el comportamiento de la función de pertenecía “delgado” para obtener “muy delgado” basta considerar lo siguiente:

$$f_{muy\ delgado}(x) = h_{muy}\left(f_{delgado}(x)\right) \qquad \text{Ecuación 7.}$$

Variables lingüísticas.

En otras palabras, a la variable que se representa mediante un término lingüístico es una variable lingüística. Por supuesto que las variables difusas permiten una transición gradual de estados, asimismo tienen la capacidad de expresar valores, también se trabaja con observaciones. Finalmente, utilizar este tipo de inconstantes radica en ajustar la forma real de representar una variable de investigación [16]. Por ejemplo, se genera el conjunto de hombres altos con todos los valores de la estatura del individuo que está constituido en un conjunto difuso el cual lo podemos observar en la figura ocho.

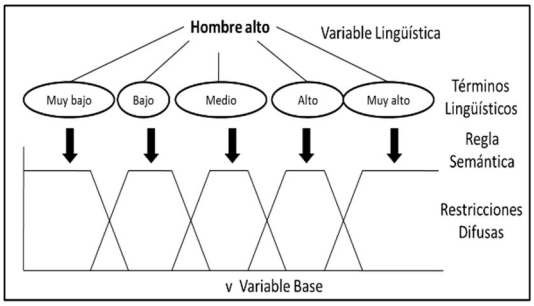


Figura 8. Conjunto difuso hombres altos.

3.3. Tipos de Controladores.

Controlador Difuso.

¿Por qué usar un controlador difuso? El modelo difuso se aplica en procesos en los cueles se utiliza la automatización de un proceso. El desarrollo del controlador difuso consiste en trabajar en la emulación de decisiones del sistema, además se clasifican los valores en forma lingüística, por ejemplo “alto”, “lento”, “rápido”, por otra parte, en el control convencional no maneja este tipo de entradas, mientras que el control difuso se describe de forma natural. De la misma forma

otra ventaja del controlador difuso, es verlo como una caja negra, a la cual se proporcionan valores de entradas, que son procesados por el modelo del mismo modo lograra generar el resultado. No obstante, en el control convencional es necesario conocer el modelo matemático del sistema, a la inversa al utilizar el control difuso no es necesario conocer el modelo matemático [16].

Definiendo un controlador difuso.

Otro punto es la estructura del controlador difuso el cual está constituido por cuatro partes: En primer lugar, la interfaz de difusificación, en segundo lugar, la base de conocimiento, en tercer lugar, la lógica de decisiones finalmente la interfaz de desdifusificación. Naturalmente estas partes interactúan entre los procesos del controlador difuso, cada una de sus elementos los podemos identificar en la figura 9.

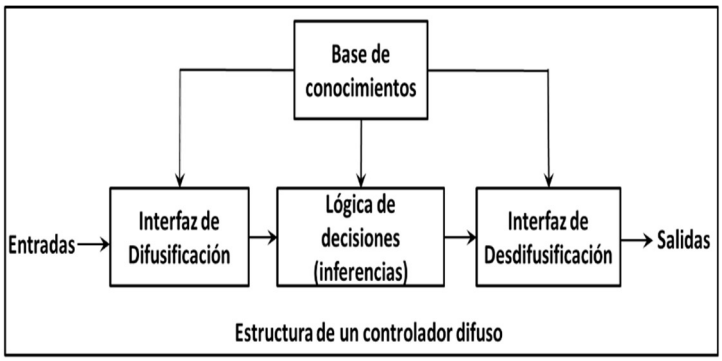


Figura 9. Estructura de un controlador difuso.

Interfaz de defusificación.

Mide las variables de entrada para ser transferidas al universo de discurso. Asimismo, la defusificación convierte los datos de entrada en valores lingüísticos, siendo las etiquetas de la función de pertenencia. Igualmente, la representación de información a través del conjunto difuso se puede realizar en forma discreta. De nuevo al segmentar el universo en un número definido, es posible definir un conjunto difuso asignando un grado de pertenencia a cada elemento genérico del conjunto universo. Al lado de los niveles de “discretización” tienen

influencia en la obtención del control. Por otra parte, para la “discretización” es necesario realizar un mapeo para transformar el valor medido a valores del universo discreto, ya sea de manera uniforme y/o no uniforme. Igualmente, una variable lingüística se asocia a un conjunto de términos, definido en el mismo universo de discurso, En otras palabras, para representar los números difusos está relacionado con la complejidad del controlador, además éstos tienen un significado lingüístico como “grande”, “bueno”, “pequeño”. De hecho en la figura diez se muestran dos particiones difusas en el mismo universo, normalizado de -1 a +1 [16].

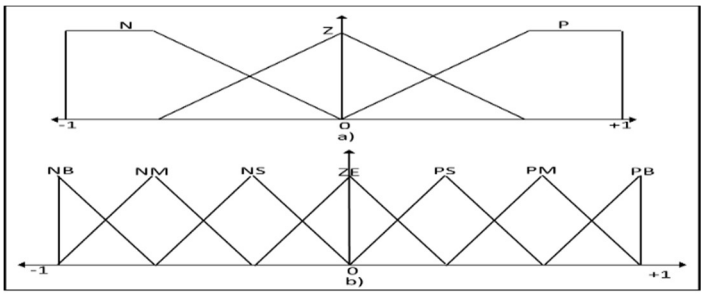


Figura 10. Participaciones difusas con distinto número de términos:
a) Tres términos N, Z y P; b) Siete términos NB, NM, NS, ZE, PS, PM y PB.

Base de conocimiento.

La base de conocimiento es la información que se obtiene de la aplicación que se va a controlar, así también de las metas del controlador. Entonces está formada por una base de datos que es de igual forma una base de reglas lingüísticas para controlar la variable de entrada. De igual importancia la base de datos proporciona definiciones para el establecimiento de reglas igualmente la manipulación de datos difusos. Otra vez la base de reglas se caracteriza en las metas de control al lado de la política que utilizan los expertos para lograr el control; que se representa empleando proposiciones. Asimismo, el control difuso debe ser capaz de inferir una acción de control correspondiente para cada estado del proceso que se va a controlar. Por otra parte, la estrategia de la base de datos comprende la definición de los conjuntos difusos. Luego los modos de derivación de las reglas difusas de control se contemplan en la experiencia de

expertos, de igual forma el conocimiento del control además de las acciones de control del operador se establecen en forma de proposiciones condicionales que se relacionan con las variables de estado en el antecedente del proceso interactuando con las variables de control en el proceso obteniendo un conjunto de consecuencias [16].

En general el conjunto de reglas se define como:

$$(*_1 \text{ es } A_{x1}^{(k)}) \text{ y } (\ldots) \ldots (\ldots) \text{ y } (*_n \text{ es } A_{xn}^{(k)}),\ k = 1, 2, \ldots, m \qquad \text{Ecuación 8.}$$

Donde:

X: Dominio físico actual donde tienen sentido los valores lingüísticos.

A_x: Conjunto de valores lingüísticos que * puede tomar, un elemento arbitrario es Lx.

*: Nombre simbólico.

μx: Función semántica de interpretación de un valor lingüístico en términos cuantitativos.

μx: A*→Ãx.

Lógica de decisiones.

Otro punto es la lógica de un controlador difuso en su núcleo, se emula el conocimiento perceptivo utilizado para la toma de decisiones, asimismo se emplean implicaciones también reglas establecidas; según la base de conocimiento [16]. De la misma forma el resultado de disparar estas reglas con valores de entrada físicos nítidos X_1 k, X_2 k,…, X_n k será m conjuntos difusos recortados, los cuales se denotaran en la ecuación nueve.

$$C_{\tilde{A}u}^{(1)},\ldots,C_{\tilde{A}u}^{(m)} \qquad \text{Ecuación 9.}$$

En otras palabras, se pueden tener m conjuntos escalados, los cuales se representan en la ecuación diez.

$$S_{\tilde{A}u}^{(1)},\ldots,S_{\tilde{A}u}^{(m)} \qquad \text{Ecuación 10.}$$

Es decir, los conjuntos recortados también conjuntos escalonados son representados en la figura once.

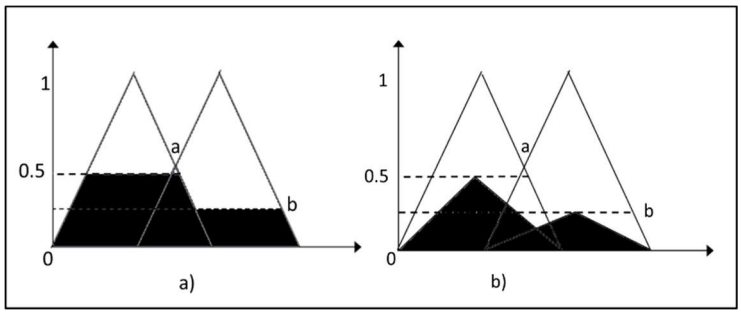


Figura 11. a) Conjuntos recortados. b) Conjuntos escalados.

Interfaz de defusificación.

Por lo que se refiere a la interfaz de defusificación siendo la encargada del proceso de control, igualmente se convierte en el rango de valores de las variables de salida a su respectivo universo de discurso. Por consiguiente, la defusificación se logra obtener de la acción del control clásico que se genera a partir de una acción de control difusa. En otras palabras, la variable de salida de control se obtiene como la unión de las salidas de control cortadas, que se expresa en la ecuación once [16].

$$\text{ũ} = u \; c \; \tilde{A} \; u \qquad \text{Ecuación 11.}$$

Asimismo, el valor nítido se denota por u* y el área del conjunto se define en la ecuación doce.

$$\int_u \mu_u(u)\, du \qquad \text{Ecuación 12.}$$

Por consiguiente, los parámetros principales de un controlador difuso se enlistarán, no obstante, al existir esta lista no existen procedimientos establecidos para el desarrollo del controlador difuso:

1. Estrategias de defusificación: además de la interpretación de un operador.
2. Base de datos: Participación del espacio, elección de las funciones de pertenencia.

3. Base de reglas: Elegir las variables de entrada, las variables de salida, derivación de las reglas, consistencia, interactividad y abarcar todos los casos.

4. Lógica de decisiones: Definir las implicaciones difusas, interpretación de los conectores, definición de las composiciones, mecanismo de inferencia.

5. Estrategias de defusificación y la interpretación de un operador de defusificación. u~=u c Ãum k.

Los controladores difusos se clasifican en clásicos con PID convencional como respaldo, sintonizador de PID convencional programador de ganancias para PID [16].

Reglas de control.

Para diseñar un sistema difuso, el primer paso consiste en definir las reglas de control, de la misma forma se pueden observar en la figura doce, así también se definen las siguientes operaciones:

$$e(k) = r(k)-w(k).\quad w = \text{velocidad.} \qquad \text{Ecuación 13.}$$

$$de(k) = e(k)-e(k-1) \qquad \text{Ecuación 14.}$$

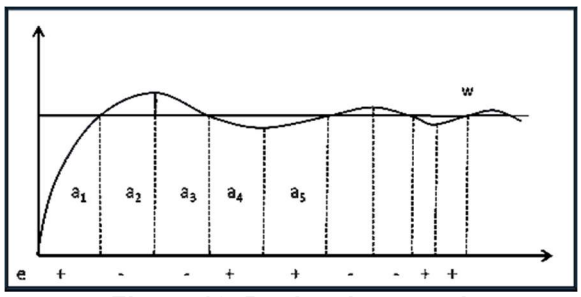


Figura 12. Reglas de control.

a) Se definen las zonas generales que se pueden ver en la ecuación quince.

$$a_1: \{e>0,\ de<0\} \ldots a_n: \{e>0,\ de<0\} \qquad \text{Ecuación 15.}$$

b) Además, se deben de tomar en cuenta los casos donde e=0 los cuales se pueden estudiar en la figura trece y las ecuaciones dieciséis, diecisiete y dieciocho.

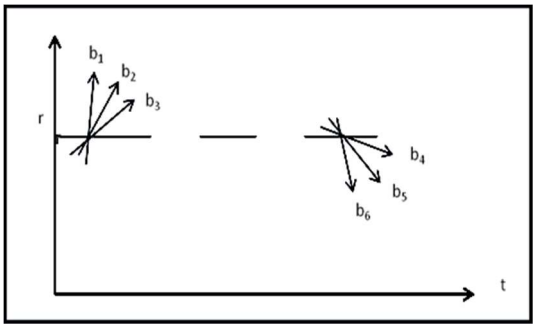


Figura 13. Casos e = 0.

b_1: {e = 0, de <<< 0}; de muy menor que 0. Ecuación 16.

b_2: {e = 0, de << 0}; de bastante menor que 0. Ecuación 17.

b_3: {e = 0, de < 0}; de menor que 0. Ecuación 18.

En forma similar b_4, b_5, b_6 >,>>,>>> 0 respetivamente

c) También se deben de tomar en cuenta los casos que toma el valor de cero la variable de, la cual podemos ahondar en cada caso en la figura catorce.

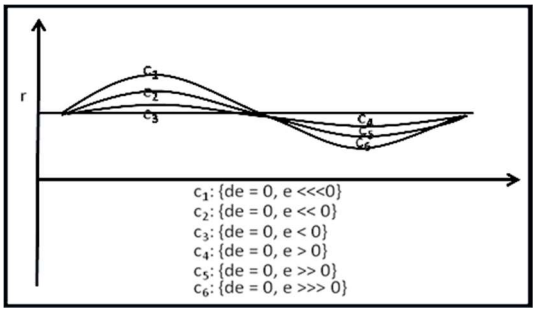


Figura 14. Casos de = 0.

Plano de estados.

Con el objetivo de definir los estados N: Negativo, P: Positivo, M: Mediano, S: Pequeño, ZE: cero, estos estados son importantes en la estructura del controlador difuso, representados en la tabla tres.

Tabla 3. Plano de estados.

de(k) \ e(k)	NB	NM	NS	ZE	PS	PM	PB
NB				$.b_1$			
NM		$.a_1$		$.b_2$		$.a_1$	
NS				$.b_3$			
ZE	$.c_6$	$.c_5$	$.b_4$	ZE	$.c_3$	$.c_2$	$.c_1$
PS				$.b_4$			
PM		$.a_3$		$.b_5$		$.a_4$	
PB				$.b_6$			

Partiendo de puntos arbitrarios en la figura doce, para el eje x se seleccionan funciones de pertenencia cuyos máximos (con pertenencia igual a uno) corresponden a los puntos. Así mismo, en el eje y también se seleccionan funciones de pertenencia con máximos en las interacciones [16]. Se eligen las reglas para realizar la aproximación que se observan en la tabla cuatro, establecidas usualmente en la figura quince para obtener la aproximación.

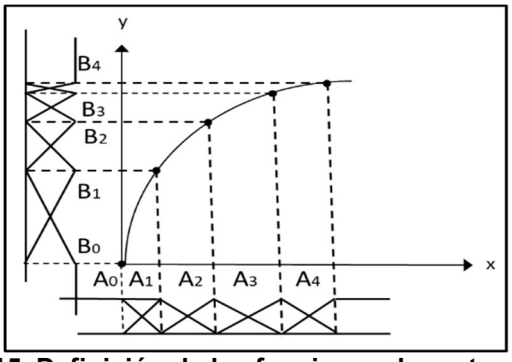


Figura 15. Definición de las funciones de pertenencia.

Tabla 4. Reglas de control.

Si ***x*** pertenece al dominio A_0, entonces ***y*** pertenece al dominio B_0
Si ***x*** pertenece al dominio A_1, entonces ***y*** pertenece al dominio B_1
Si ***x*** pertenece al dominio A_2, entonces ***y*** pertenece al dominio B_2
Si ***x*** pertenece al dominio A_3, entonces ***y*** pertenece al dominio B_3
Si ***x*** pertenece al dominio A_4, entonces ***y*** pertenece al dominio B_4
Si ***x*** pertenece al dominio A_5, entonces ***y*** pertenece al dominio B_5
Si ***x*** pertenece al dominio A_6, entonces ***y*** pertenece al dominio B_6
Si ***x*** pertenece al dominio A_7, entonces ***y*** pertenece al dominio B_7

Controlador P.

En particular la ley de control en un acoplador tipo P se define como $u = K_P \, Ɛ$, donde $Ɛ=r-y$. A partir de esto, se proponen las siguientes reglas, representadas de forma matricial en la tabla cinco denominada matriz de asociación difusa o FAM [16].

Tabla 5. Matriz de asociación difusa Controlador P.

Ɛ	N	Z	P
U	N	Z	P

Controlador PD.

Para ilustrar la ley de control en un acoplador tipo PD se define como $u = K_p \, Ɛ + K_p \, Ɛ$, donde $Ɛ = r - y$, se propone la FAM en la tabla seis [16].

Tabla 6. Matriz de asociación difusa Controlador PD.

Ɛ	N	Z	P
N	N	N	P
Z	N	Z	P
P	N	P	P

Controlador PI.

A su vez la ley en un acoplador tipo PI se define como $u = K_p \, Ɛ + K_I \int Ɛ \, d \, Ɛ$, donde $Ɛ=r–y$. Siendo que la derivada de la ley de control es $\dot{u} = K_p \, \dot{ɛ} + KI \, Ɛ$, se integra la salida de un controlador PD es posible obtener un controlador PI empleando la FAM del controlador PD que se representa en la figura dieciséis [16].

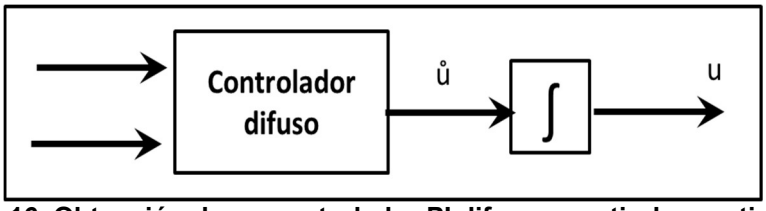


Figura 16. Obtención de un controlador PI difuso a partir de uno tipo PID.

Controlador PID difuso.

El siguiente punto trata de dos maneras de armar un controlador PID difuso. La primera propuesta es colocando en un bloque las variables de error, la derivada del error así mismo la doble derivada del error el cual podemos profundizar en la figura diecisiete [16].

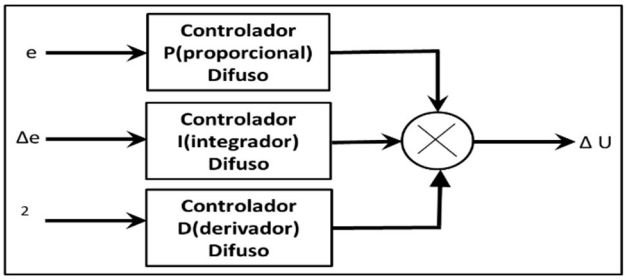


Figura 17. Controlador PID difuso, primera propuesta.

La segunda propuesta es creando cada variable por separado, sumando los bloques así mismo obtener así el PID el cual podemos sondear en la figura dieciocho [16].

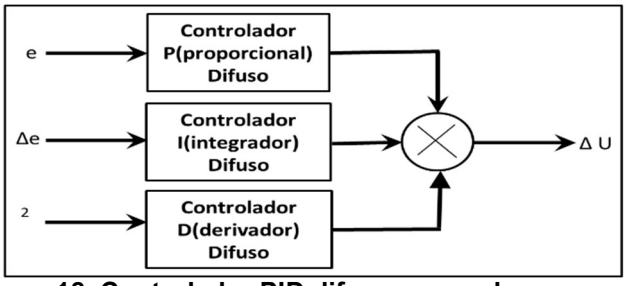


Figura 18. Controlador PID difuso, segunda propuesta.

Diseño con bases en Sugeno.

El siguiente punto trata de la propuesta que realizó Sugeno [10] en el diseño de controlador difuso al implementar un modelo, por otra parte, se deberán de relacionar las variables de entrada con sus variables de salida. También una vez desarrollado este modelo, es posible generar un controlador difuso que este interactuando con las variables de entrada con las variables de salida del sistema, con un proceso de deducción inversa [16]. A continuación, se explica un método propuesto por Takagi y Sugeno [1] para la identificación difusa de sistemas,

su aplicación en modelado por otra parte del control. En otras palabras, es un método matemático para hacer una implicación difusa (Z):

$$Z: \text{Si } f(x_i \text{ es } A_i, \dots, x_i \text{ es } A_k) \text{ entonces } y = g(x_1, \dots, x_k) \qquad \text{Ecuación 19.}$$

Donde:

y: Variable de la consecuencia, cuyo valor es inferido.

$x_1, \dots, x_k$: Variables de la premisa, además de que también aparecen en la consecuencia.

$A_1, \dots, A_k$: Conjuntos difusos con funciones de membrecía lineales, representando un sub espacio difuso, cuya implicación Z puede ser usada para el razonamiento.

f: Función lógica que conecta las proposiciones en la premisa.

g: Función que implica el valor de y cuando $x_1, \dots, x_k$ satisface la premisa.

En el presente capítulo se describe la lógica difusa, de igual manera los conjuntos difusos y finalmente los diferentes tipos de controladores que existen.

Capítulo 4.

Plataforma de implementación y optimización.

En este capítulo se describe la plataforma de desarrollo del proyecto de igual importancia la optimización que se ha utilizado en este trabajo de investigación, en primer lugar, se describe el sistema de frenado del robot **Scribbler®**, de igual importancia se especifica el desarrollo de la implementación de los algoritmos de amplitud además primero el mejor, asimismo se delinea una poda que se plantea de los nodos del árbol, de la misma forma se han implementado los dos algoritmos con este proceso para realizar la poda, también se empieza a formular el seudocódigo del algoritmo primero el mejor con lógica difusa, finalmente se desarrolla el diseño del controlador difuso.

4.1 Robot Scribbler®.

En cuanto al robot tiene la capacidad de realizar movimientos en función de la información que obtiene de sus sensores. Por otra parte, el móvil es un sistema de control de lazo cerrado al realizar un cambio de posición de sus elementos en función de la información recibida de su entorno mediante sus sensores [23]. Se define la ecuación de lazo cerrado, la cual la podemos estudiar en la ecuación veinte.

$$M(s) = G(s)/(1 + G(s)\,) \qquad \text{Ecuación 20.}$$

Además, el robot **Scribbler®** [24] es un móvil perfecto para aprender por otra parte enseñar el cual es ideal para iniciarse en la robótica, es lo suficientemente simple para poderlo configurar en minutos, también es lo suficientemente potente como para desarrollar actividades interesantes. Al lado de la programación, también la robótica por otro lado la integración de las matemáticas igualmente el arte es posible con este móvil. Además, sus bloques entrelazados con código legible permiten ingresar en sus programas de una manera intuitiva. En concreto los programas pre configurados se han incluido todos sus sensores de igual importancia los motores del robot. El robot **Scribbler®** pertenece a los robots móviles que operan en ambientes

no estructurados, debe enfrentarse con la incertidumbre en la posición e identificación de objetos. Además, se utiliza para lograr explorar regiones desde un punto inicial hasta un punto final; este modelo se utilizó para este proyecto de investigación, también se analizaron las características del móvil, de igual manera se determinó que el móvil cuenta con tres sensores de luz en la parte frontal, además tres ruedas que son parte de su sistema de locomoción. A continuación, se mostrarán los resultados de un reporte del frenado del móvil, realizado por Jiménez [25]. En primera instancia se realizaron los procesos para encontrar la función de control para el movimiento de los motores del móvil. De igual forma se utilizaron los sensores de luz de la estructura física del móvil así mismo el micro basic stamp 2 [26]. Inicialmente se propuso realizar un programa con la finalidad de leer la luz a través de un cartón cortado en una reja para obtener las velocidades que tiene el robot móvil al pasar por cada tramo de luz, esta idea se puede estudiar en la figura diecinueve.

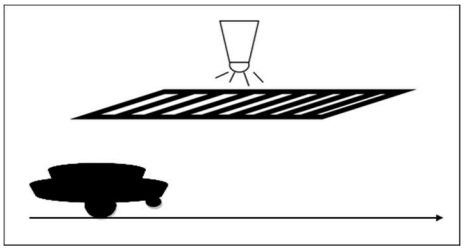


Figura 19. Sensores de robot.

En efecto se realizaron varias pruebas en la trayectoria del robot, a continuación, se describe el frenado del móvil. Por otra parte, para este proceso, se debe de proporcionar una velocidad de entrada que se indica como tiempo cero, se obtiene una muestra en diferentes instantes al realizar el frenado, estos resultados se pueden investigar en la tabla siete.

Tabla 7. Frenado del robot.

Tiempo	Sensor	Tiempo	Sensor	Tiempo	Sensor	Tiempo	Sensor
0 ms	235	15 ms	263	30 ms	359	45 ms	397
3 ms	245	18 ms	265	33 ms	371	48 ms	399
6 ms	246	21 ms	313	36 ms	371	51 ms	376
9 ms	237	24 ms	372	39 ms	401	54 ms	427
12 ms	244	27 ms	360	42 ms	351	57 ms	391

En este caso encontramos la gráfica del sensor, la cual se muestra en la figura veinte.

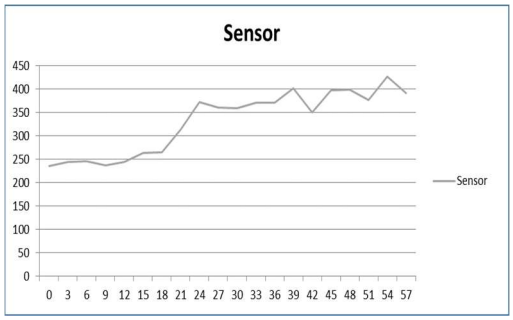


Figura 20. Sensor del móvil.

Al utilizar los datos anteriores, se logró ubicar la tarea de paro que se representan en la tabla ocho, con los datos obtenidos se generaron las gráficas de distancia en la figura veintiuno y la velocidad en la figura veintidós.

Tabla 8. Paro del robot.

Tiempo	Distancia	Velocidad	Tiempo	Distancia	Velocidad	Tiempo	Distancia	Velocidad
0 ms			21 ms	9.4	0	42 ms	9.4	0
3 ms	1.566666	0.522222	24 ms	9.4	0	45 ms	9.4	0
6 ms	3.133333	0.522222	27 ms	9.4	0	48 ms	9.4	0
9 ms	4.7	0.522222	30 ms	9.4	0	51 ms	9.4	0
12 ms	6.266666	0.522222	33 ms	9.4	0	54 ms	9.4	0
15 ms	7.833333	0.522222	36 ms	9.4	0	57 ms	9.4	0
18 ms	9.4	0.522222	39 ms	9.4	0		0	

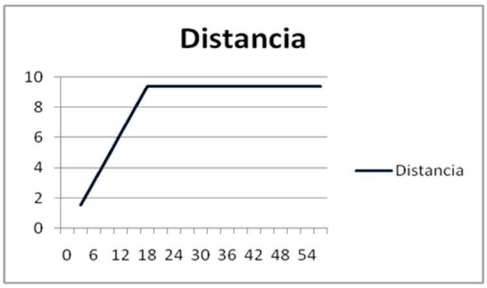


Figura 21. Distancia en el paro.

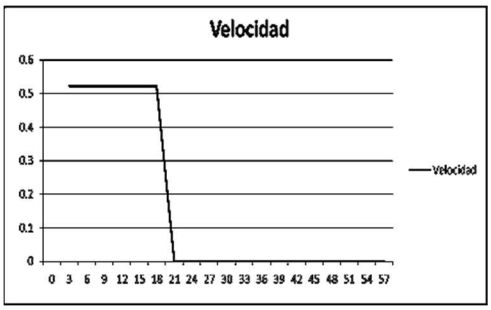


Figura 22. Velocidad en el paro.

En otras palabras, se determinar la función de velocidad, la cual se describe en la ecuación veintiuno.

$$v(t) = \frac{1}{2}u(t) - \frac{1}{2}u(t-18) - \left(\frac{t}{6} - \frac{7}{2}\right)u(t-18) + \left(\frac{t}{6} - \frac{7}{2}\right)u(t-24) \quad \text{Ecuación 21.}$$

Es decir, conseguir la transformada de Laplace para la velocidad que podemos estudiar en la ecuación veintidós.

$$L\{v(t)\} = V(s) = \frac{1}{2s}\left(1 + \frac{e^{-18s}+e^{-24s}}{3s}\right) \quad \text{Ecuación 22.}$$

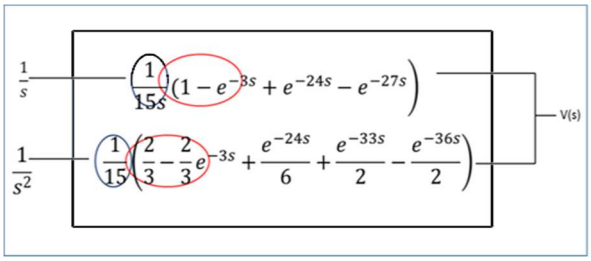
Suponiendo que la entrada es R = ⍜ 0 se obtienen las ecuaciones veintitrés, veinticuatro y veinticinco:

$$G(s) = \frac{V(s)}{R(s)} \quad \text{Ecuación 23.}$$

$$G(s) = sV(s) \quad \text{Ecuación 24.}$$

$$G(s) = \frac{1}{2}\left(1 + \frac{e^{-18s}+e^{-24s}}{3s}\right) \quad \text{Ecuación 25.}$$

Después de haber obtenido estas funciones, se modelan los bloques de la siguiente forma, encontramos diferencias entre las funciones debido a que la función de la rampa como en la función del escalón, no se pudieron alcanzar las mismas condiciones entre ellas, pues no se llegó a la misma velocidad también en la rampa se incrementó en el mismo tiempo, en tanto en el escalón se encontró una constante que se puede investigar en la ecuación veintiséis.



Ecuación 26.

Mejor dicho, en la ecuación veintiséis se detectan puntos en común, el factor es el mismo en la salida, en los primeros factores se pueden obtener el mismo valor, que se estudia en la figura veintitrés.

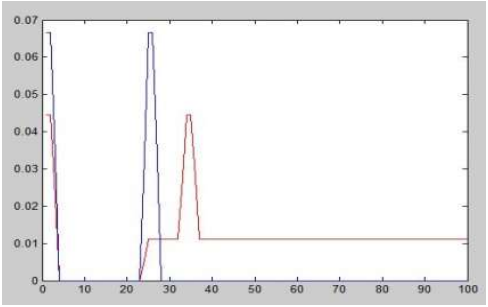


Figura 23. Representación de los sensores.

4.2. Desarrollo e implementación del simulador.

En relación con el desarrollo del simulador se tomaron en cuenta los tres sensores de luz que cuenta el móvil **Scribbler®** [24], de la misma forma se eligió la estructura de datos llamada árbol trinario el cual se puede observar en la figura veinticuatro para poder representar los tres sensores de luz con los que cuenta el robot móvil. Asimismo, la raíz del árbol es el punto de partida donde se colocará el móvil. Además, se implementó una estructura de datos llamada pila; por tanto, las definiciones de ambas estructuras se encuentran en el glosario.

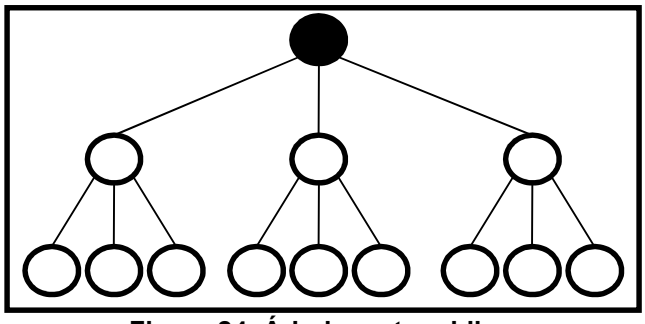


Figura 24. Árbol con tres hijos.

Por lo que se refiere al simulador se utilizó el lenguaje de programación orientado a objetos Java, además la interfaz gráfica del simulador se ha representado en la figura veinticinco. Finalmente se describen cada uno de los elementos de la interfaz del simulador:

Dibujo del mapa: Se ha representado el mapa al que se realizará la exploración, así también los objetos del mapa, son representados con figuras geométricas en color negro.

Botones de manipulación: Se implementaron botones como parte de la interfaz de usuario, los cuales se describen a continuación:

- **Generar ruta:** Este botón tiene la función de poner en funcionamiento la exploración del mapa con el algoritmo de búsqueda (figura veinticinco).
- **Pausar su ejecución:** Tiene la función de detener la exploración (figura veinticinco).
- **Obtener la traza:** Tiene la tarea de desplegar la ruta desde el estado inicial hasta lograr llegar al estado objetivo, así también se representa el recorrido del móvil en el camino solución (figura veinticinco).
- **Limpiar el mapa:** Este botón tiene la tarea de limpiar el mapa después de que se realiza una exploración (figura veinticinco).

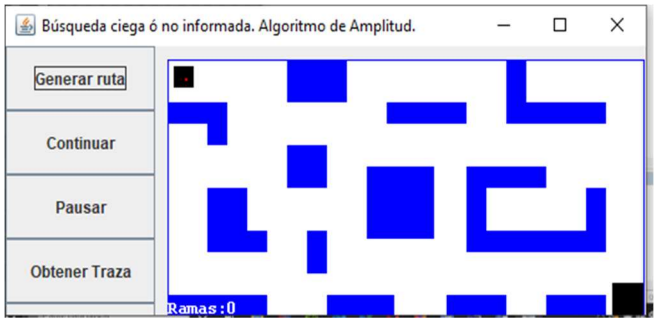


Figura 25. Interfaz de simulador.

4.3. Implementación en él simulador búsqueda en amplitud.

En cuanto a la exploración del mapa se puede estudiar el funcionamiento del algoritmo en la búsqueda de amplitud, asimismo se muestra el árbol de búsqueda con todos los nodos hijos con sus respectivos vértices. Por consiguiente, se despliega el camino solución además del robot móvil en movimiento explorando la ruta trazada, finalmente, lograr el mapeo representado

en el simulador logrando una longitud de cuatro mil setecientos seis, la cual podemos estudiar en la figura veintiséis.

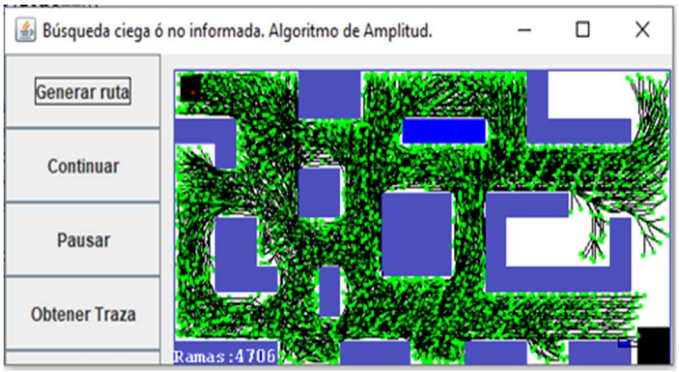


Figura 26. Simulador con amplitud.

De forma tal que el algoritmo de búsqueda en amplitud tiene como ventaja encontrar siempre la ruta solución si existe; además sea la de menor coste además en todo caso la de menor profundidad [19]. Al contrario, la desventaja está en el factor de ramificación que se genera de igual importancia se almacenan muchos nodos en el árbol.

4.4. Implementación en simulador algoritmo primero el mejor.

Desarrollo de la función heurística:

Otro punto es el desarrollo de esta función, en la cual se utiliza la ecuación de distancia entre dos puntos. Asimismo, se toman las coordenadas del nodo que se generará como parte del estado inicial también las coordenadas de estado final, esta tarea se describe en la ecuación veintisiete. En la última parte, se evalúan los tres nodos hijos del árbol, comparando las tres distancias por otra parte se elige la distancia más cercana al lugar del punto final.

$$d(p_1, p_2) = \sqrt{(x_1 - x_2)^2 + (y_1 - y_2)^2} \qquad \text{Ecuación 27.}$$

Por otra parte, en la implementación del algoritmo primero el mejor, se puede observar el funcionamiento de la función heurística del mejor nodo entre los tres nodos hijos. Finalmente

se muestra el árbol solución para el algoritmo primero el mejor, también el árbol solución logra llegar a la profundidad de longitud de veintiocho hijos este resultado se puede estudiar en la figura veintisiete.

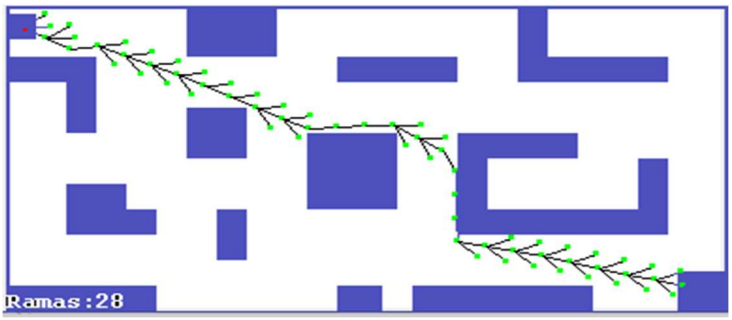


Figura 27. Simulador con primero el mejor.

Sin duda la principal ventaja del algoritmo que es completo, además supera el problema de los mínimos locales por una parte las mesetas que tenían el método en escalada. No obstante, su principal inconveniente está en la **fev**, como sucede con la evaluación lo cerca que esta del estado meta, pero sin tener presente el coste del camino ya recorrido lo cual implicaría en no ser la mejor. Conocer los algoritmos de búsqueda informada también de búsqueda desinformada los cuales se utilizan en el SLAM de robots móviles siendo vitales para lograr la autonomía del móvil; dado entonces la lógica difusa es una técnica que se desarrolla con el objetivo de lograr la toma de decisiones por parte de los robots móviles al lado de lograr contribuir con él SLAM.

4.5. Poda de nodos.

El siguiente punto que se trata es lograr una poda de los nodos que se generan próximos a los objetos, por tal razón el móvil no los puede visitar lo anterior es por la estructura física del móvil, al principio se realizó un estudio de los algoritmos de búsqueda sin información, así como los de información, con el objetivo de mejorar sus resultados. Por esto se logró determinar que existen nodos que se les dará el nombre de nodos no válidos. De igual importancia estos nodos

al ser explorados se crearían colisiones entre el robot del mismo modo con los objetos del mapa, por otra parte, para eliminar dichos nodos se plantea una poda en el árbol de búsqueda, asimismo esta idea se logra implementar con la ayuda del proceso explicado en la figura veintiocho.

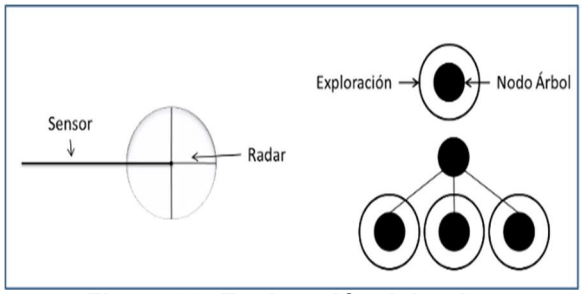


Figura 28. Exploración del radar.

Después para lograr el proceso de poda es necesario definir una función de membrecía triangular, la cual se describe en la ecuación veintiocho. Además, tomando en cuenta la ecuación veintiocho; se utilizaron treinta seis grados asimismo se asigna el valor de 0.1, también setenados grados también se le asigna el valor de 0.2 sucesivamente hasta llegar a 1.0. Entonces se logra esta convención la cual se muestra en la tabla nueve.

Tabla 9. Poda en el árbol.

Grados	Valor
0°	0.0
36°	0.1
72°	0.2
108°	0.3
144°	0.4
180°	0.5
216°	0.6
252°	0.7
288°	0.8
324°	0.9
360 °	1.0

Por otra parte, se implementa la función triangular definida en la ecuación veintiocho en los sensores del robot. De igual importancia, se puede ver la gráfica de la función de membrecía triangular con los valores establecidos que se puede estudiar en la figura veintinueve.

$$\mu_A(x) = \begin{cases} 0 & x < a \\ \frac{x-a}{b-a} & a \leq x < b \\ \frac{c-x}{c-b} & b \leq x \leq c \\ 0 & c < x \end{cases}$$ Ecuación 28.

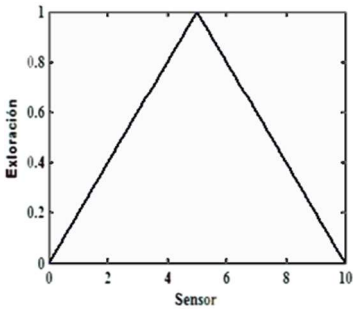


Figura 29. Función triangular de los datos del sensor.

Mejor dicho, la gráfica de membrecía se obtiene del número difuso, el cual se puede observar en la parte superior de la figura treinta, un pequeño punto de color negro.

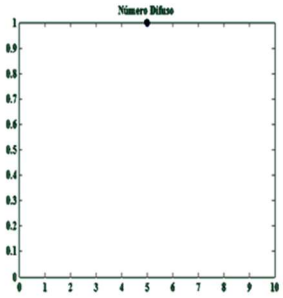


Figura 30. Número difuso.

La exploración en los sensores se puede observar en la figura treinta y uno, la cual se explica a continuación:

- **Sensor izquierdo:** Al realizar la exploración la función de membrecía genera el valor de uno, por lo anterior se genera el nodo.
- **Sensor central:** Al realizar la exploración la función de membrecía genera el valor de uno, por lo anterior se genera el nodo.

- **Sensor derecho:** Al realizar la exploración la función de membrecía genera el valor de 0.30, por lo anterior no se genera el nodo.

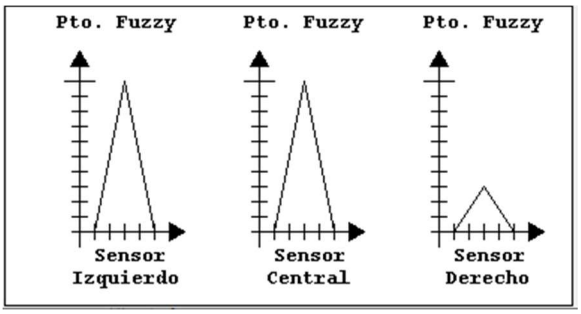


Figura 31. Graficas triangulares del funcionamiento de los 3 sensores.

4.6. Implementación del algoritmo de amplitud con poda.

En cuanto al ser implementada la poda de los nodos, se puede ahondar en el espacio que se ha generado entre el árbol de búsqueda también los objetos del mapa. Por tanto, en la figura treinta y dos se muestra el árbol del algoritmo en amplitud con la poda, el cual ha logrado tener la profundidad de mil setecientos para este mapa.

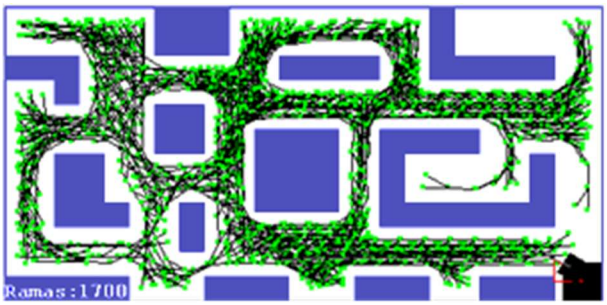


Figura 32. Árbol con 1700 ramas y graficas de los tres sensores del móvil.

También en la tabla diez se puede identificar la longitud del árbol en los algoritmos amplitud sin poda también con poda, como parte de este trabajo al implementar la poda se puede determinar que la longitud del árbol se ha reducido en la exploración del mapa.

Tabla 10. Comparación algoritmos con poda y sin poda.

Algoritmo	Longitud
Algoritmo de amplitud sin poda.	4,706
Algoritmo de amplitud con poda	1,700

4.7. Implementación de las variables lingüísticas.

El siguiente punto trata de formular las variables independientes llamadas distancia además velocidad las cuales se transformarán en variables lingüísticas. Igualmente, se realizará el procedimiento para generar el conjunto difuso, implementado cada una de sus partes. En una primera etapa se analiza la implementación de la variable lingüística. En una segunda etapa se definirán los términos lingüísticos de la inconstante. En la tercera parte se implementan las reglas semánticas en la variable, posteriormente se definen las restricciones difusas, finalmente se obtiene el conjunto difuso con cada uno de sus elementos como son la variable lingüística, los términos lingüísticos, regla semántica al lado de restricciones difusas. En otras palabras, podemos ver la variable difusa velocidad la cual se puede considerar en la figura treinta y tres.

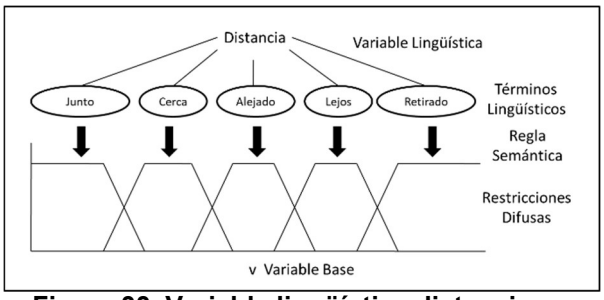


Figura 33. Variable lingüística distancia.

De este modo se realizará el procedimiento para obtener el conjunto difuso velocidad. En una primera etapa se implementa la variable lingüística, en la segunda etapa se definen los términos lingüísticos. Asimismo, se define la regla semántica, posteriormente se definen las restricciones difusas, con lo anterior se obtiene el conjunto difuso, con la variable lingüística, los términos lingüísticos, regla semántica igualmente restricciones difusas, el proceso descrito anteriormente se pueden observar en la figura treinta y cuatro.

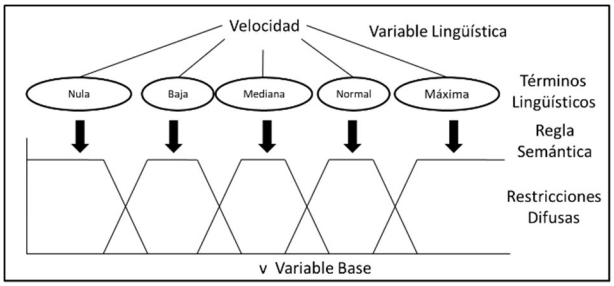


Figura 34. Variable lingüística velocidad.

4.8. Implementación en el simulador, algoritmo primero él mejor.

Por lo que se refiere a la implementación en el simulador, en la figura treinta y cinco se puede observar el resultado de la función de membrecía interactuando en la exploración del mapa. Al lado se puede estudiar el árbol solución del algoritmo primero el mejor con la poda, también el árbol tiene como longitud de veintinueve para este mapa.

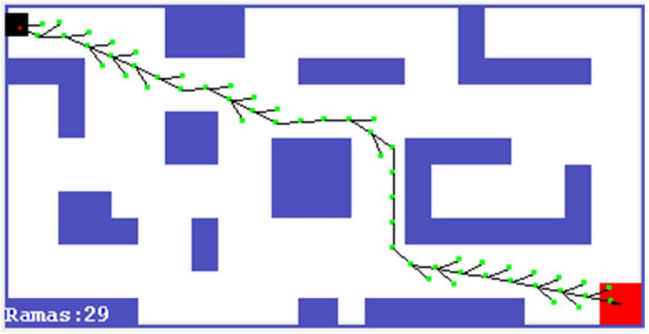


Figura 35. Algoritmo primero el mejor.

4.9. Algoritmo primero el mejor con lógica difusa.

En relación con el algoritmo primero el mejor la **fev**, que denominaremos d(n), también se va a medir la distancia estimada entre el nodo actual hasta la meta, al lado de su valor calculado para cada estado se almacenará junto con él igualmente el apuntador a su nodo padre dentro de la estructura llamada cola abierta. En este momento la cola cerrada es imprescindible para comprobar que no se repitan estados en el árbol de exploración. Otra vez al ser generado cada conjunto sucesor con la **fev**, se insertarán los valores del conjunto difuso, en cada conjunto de los sucesores.

A continuación, se describe dicho algoritmo de búsqueda:

- Crear árbol de búsqueda G, con el nodo inicial, I.
- Abierta = I.
- Éxito = Falso.
    - o Hasta que abierta esté vacía o éxito.
    - Quitar el primer nodo de abierta.
    - SI N es estado - final entonces éxito = Verdadero.
- SI NO Expandir N, generando el conjunto S de sucesores de N, que no son antecesores de N en el árbol.
- Se insertará el conjunto difuso en cada nodo en G por cada s de S.
- Establecer un puntero a N desde aquellos s de S que no estuvieran ya en G añadirlos a abierta.
- Para cada s de S que estuviera ya en abierta decidir si redirigir o no sus punteros hacia N.
- Reordenar abierta según f (n).
- Si éxito entonces solución = camino desde I a N a través de los punteros de G.
- Si no solución=fracaso.

FIN.

4.10. Diseño del controlador difuso.

Para empezar con el desarrollo del controlador difuso representado en la figura treinta y seis se toman las variables difusas distancia del mismo modo la de velocidad representadas en la figura treinta y siete, de la misma forma al realizar este proceso se genera la variable frenar (F), asimismo se define como variable dependiente que es necesaria para realizar la exploración del móvil, igualmente con los valores representados en la figura treinta y ocho, así también se obtiene su FAM que se muestra en la tabla once, el procedimiento de realizar este proceso en Matlab esta descrito en el anexo A.

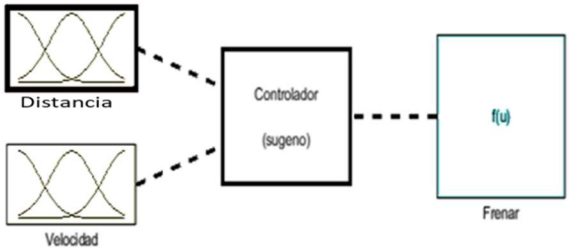


Figura 36. Controlador difuso.

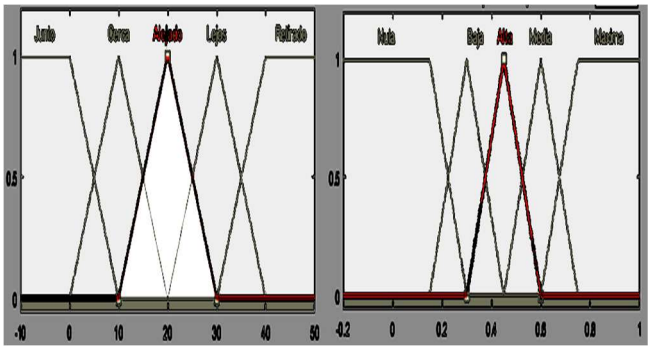


Figura 37. Variable difusa distancia y variable difusa velocidad.

Figura 38. Variable de salida frenar.

Tabla 11. FAM del Controlador Difuso.

Regla	Error	Derivada del error	Salida	Punto de referencia
1	PG	ZE	PG	A
2	PM	ZE	PM	E
3	PP	ZE	PP	I
4	ZE	NG	NG	B
5	ZE	NM	NM	F
6	ZE	NS	NS	J
7	NG	ZE	NG	C
8	NM	ZE	NM	G
9	NP	ZE	NP	K
10	ZE	PG	PG	D
11	ZE	PM	PM	H
12	ZE	PP	PM	L
13	ZE	ZE	ZE	Establecimiento

Capítulo 5.

Análisis de resultados.

5.1. Implementación en Matlab.

El siguiente punto describe la forma de generar el controlador difuso, de manera análoga, se utilizarán las variables distancia también la de velocidad; del mismo modo se han transformado estas inconstantes para generar un par de conjuntos difusos, en otras palabras, se da como resultado el controlador difuso el cual se puede estudiar en la figura treinta y nueve.

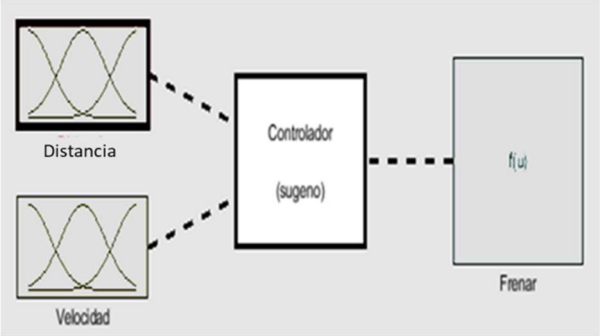


Figura 39. Parte inicial del controlador difuso.

Posteriormente se utiliza la formulación de los dos conjuntos difusos el de distancia también el de velocidad representados en la figura cuarenta. En otras palabras, se define la función frenar con los valores representados en la tabla doce.

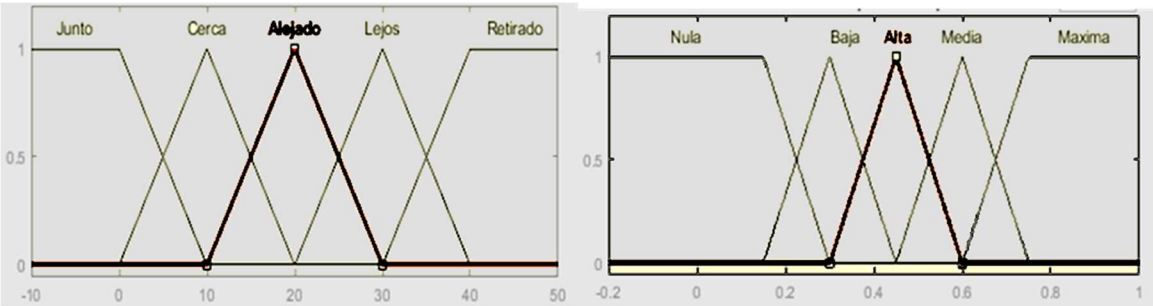


Fig. 40. Conjunto difuso de distancia. Conjunto difuso de velocidad.

Tabla 12. Función frenar.

Función	Valor
Frenar(Rápida):	3500
Frenar (Mediana más)	2750
Frenar (Mediana)	1500
Frenar (Mediana menos)	750
Frenar (Nulo)	0

Del mismo modo en la tabla trece se puede ahondar en la interacción entre las variables distancia del mismo modo la de velocidad con la variable frenar.

Tabla 13. Variables (distancia, velocidad) vs frenar.

		Distancia				
		Junto	**Cerca**	**Alejado**	**Lejos**	**Retirado**
Velocidad	**Nula**	F(nulo)	F(mediana-)	F(mediana)	F(mediana+)	F(rápida)
	Baja	F(nulo)	F(mediana-)	F(mediana)	F(mediana+)	F(rápida)
	Media	F(nulo)	F(mediana-)	F(mediana)	F(mediana+)	F(rápida)
	Alta	F(nulo)	F(mediana-)	F(mediana)	F(mediana+)	F(rápida)
	Máxima	F(nulo)	F(mediana-)	F(mediana)	F(mediana+)	F(rápida)

En la siguiente etapa se utilizan las reglas de la tabla trece para ser implementadas en el controlador difuso, lo anterior es fundamental para lograr el desarrollo del controlador difuso, posteriormente se utilizan las reglas de control que se muestran en la tabla catorce, con la finalidad de ser implementadas en el modelo Takagi – Sugeno [1].

Tabla 14. Reglas de control.

1. If (Distancia is Junto) and (Velocidad is Nula) then (Frenar is Nulo) (1)
2. If (Distancia is Junto) and (Velocidad is Baja) then (Frenar is Nulo) (1)
3. If (Distancia is Junto) and (Velocidad is Media) then (Frenar is Nulo) (1)
4. If (Distancia is Junto) and (Velocidad is Alta) then (Frenar is Nulo) (1)
5. If (Distancia is Junto) and (Velocidad is Maxima) then (Frenar is Nulo) (1)
6. If (Distancia is Cerca) and (Velocidad is Nula) then (Frenar is Mediano-) (1)
7. If (Distancia is Cerca) and (Velocidad is Baja) then (Frenar is Mediano-) (1)
8. If (Distancia is Cerca) and (Velocidad is Media) then (Frenar is Mediano-) (1)
9. If (Distancia is Cerca) and (Velocidad is Alta) then (Frenar is Mediano-) (1)
10. If (Distancia is Cerca) and (Velocidad is Maxima) then (Frenar is Mediano-) (1)
11. If (Distancia is Alejado) and (Velocidad is Nula) then (Frenar is Mediano) (1)
12. If (Distancia is Alejado) and (Velocidad is Baja) then (Frenar is Mediano) (1)
13. If (Distancia is Alejado) and (Velocidad is Media) then (Frenar is Mediano) (1)
14. If (Distancia is Alejado) and (Velocidad is Alta) then (Frenar is Mediano) (1)
15. If (Distancia is Alejado) and (Velocidad is Maxima) then (Frenar is Mediano) (1)
16. If (Distancia is Lejos) and (Velocidad is Nula) then (Frenar is Mediano+) (1)
17. If (Distancia is Lejos) and (Velocidad is Baja) then (Frenar is Mediano+) (1)
18. If (Distancia is Lejos) and (Velocidad is Media) then (Frenar is Mediano+) (1)
19. If (Distancia is Lejos) and (Velocidad is Alta) then (Frenar is Mediano+) (1)
20. If (Distancia is Lejos) and (Velocidad is Maxima) then (Frenar is Mediano+) (1)
21. If (Distancia is Retirado) and (Velocidad is Nula) then (Frenar is Rapido) (1)
22. If (Distancia is Retirado) and (Velocidad is Baja) then (Frenar is Rapido) (1)
23. If (Distancia is Retirado) and (Velocidad is Media) then (Frenar is Rapido) (1)
24. If (Distancia is Retirado) and (Velocidad is Alta) then (Frenar is Rapido) (1)
25. If (Distancia is Retirado) and (Velocidad is Maxima) then (Frenar is Rapido) (1)

Por una parte, se muestran las reglas de control de las variables independientes distancia del mismo modo la de velocidad; consecuentemente se logra que estén fuzzyficadas estas variables las cuales se utilizan para generar una variable dependiente llamada frenar, la interacción entre estas variables se puede identificar en la figura cuarenta y uno.

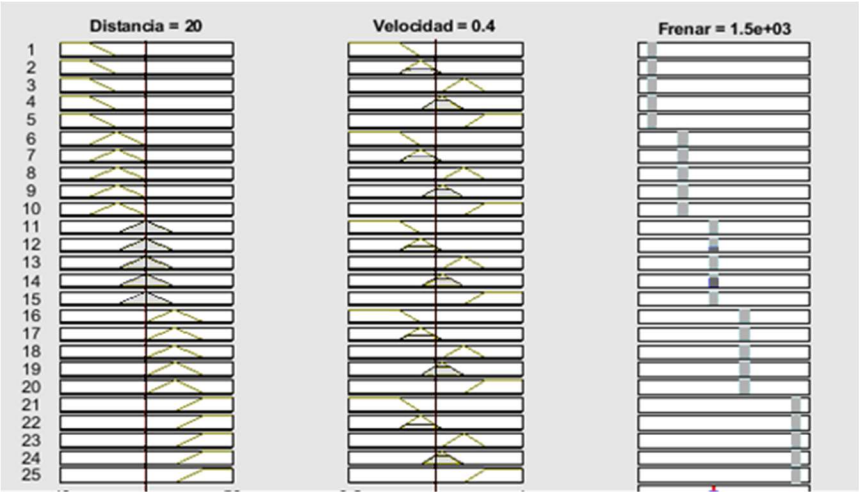


Figura 41. Reglas de salida de ambas variables.

De la misma forma se puede estudiar el controlador difuso en la figura cuarenta y dos. Por otra parte, se observa la forma de interactuar entre las variables independientes velocidad del mismo modo la de distancia con la variable dependiente frenar, finalmente se logra realizar el objetivo de este trabajo de investigación que es realizar el mapeo. En resumen, se puede observar que el sistema de control es de lazo cerrado de igual manera en el controlador difuso se logra determinar que el modelo matemático tiene estabilidad.

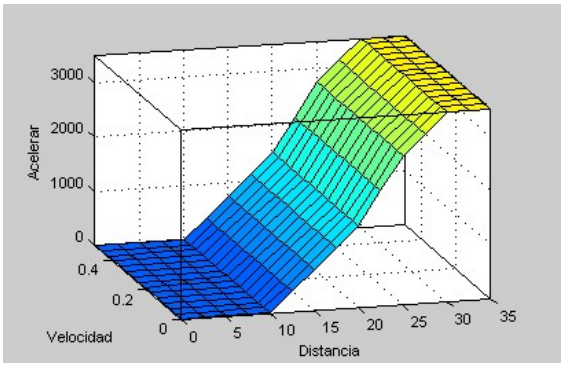


Figura 42. Curva de control.

5.2. Implementación de la lógica difusa en el simulador.

Otro punto importante es lograr la implementación de la lógica difusa que se estudió en la estructura del simulador, de nuevo, se logró determinar que la estructura de datos llamada pila no cumple con las operaciones de búsqueda del controlador difuso, por lo tanto, se llegó a la determinación de sustituir la pila por una cola en la estructura del simulador. De la misma manera al realizar esta operación en el simulador se logra implementar el nuevo algoritmo difuso. Finalmente se obtiene la misma ruta con el algoritmo de amplitud, con la diferencia que el árbol de búsqueda que se obtiene es con una profundidad de cuatro nodos, para concluir se ha logrado obtener la ruta en menor tiempo con respecto a la exploración inicial la cual se puede estudiar en la figura cuarenta y tres.

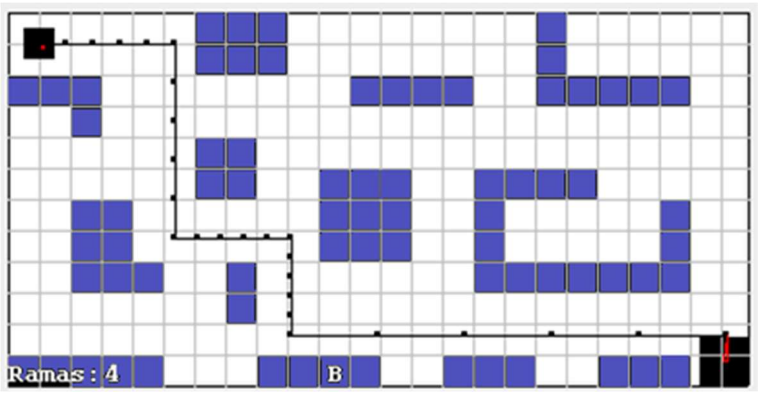


Figura 43. Implementación del simulador con el algoritmo difuso.

5.3. Algoritmo difuso.

En conclusión, utilizando las reglas del controlador difuso se formula el nuevo algoritmo.

Estructura del almacenamiento en la pila.

1. Detectar distancia de los tres sensores.

   I. Si central es igual a cero rotar el móvil e ir a 3).
   II. Si central es distinto a cero ir a 2).

2.

a) Si **velocidad** = nula y **distancia** = junto entonces **frenar** = nulo.
b) Si **velocidad** = nula y **distancia** = cerca entonces **frenar** = mediana menos.
c) Si **velocidad** = nula y **distancia** = alejado entonces **frenar** = mediana.
d) Si **velocidad** = nula y **distancia** = lejos entonces **frenar** = mediana más.
e) Si **velocidad** = nula y **distancia** = retirado entonces **frenar** = rápido.
f) Si **velocidad** = baja y **distancia** = junto entonces **frenar** = nulo.
g) Si **velocidad** = baja y **distancia** = cerca entonces **frenar** = mediana menos.
h) Si **velocidad** = baja y **distancia** = alejado entonces **frenar** = mediana.
i) Si **velocidad** = baja y **distancia** = lejos entonces **frenar** = mediana más.
j) Si **velocidad** = baja y **distancia** = retirado entonces **frenar** = rápido.
k) Si **velocidad** = media y **distancia** = junto entonces **frenar** = nulo.
l) Si **velocidad** = media y **distancia** = cerca entonces **frenar** = mediana menos.
m)Si **velocidad** = media y **distancia** = alejado entonces **frenar** = mediana.
n) Si **velocidad** = media y **distancia** = lejos entonces **frenar** = mediana más.
o) Si **velocidad** = media y **distancia** = retirado entonces **frenar** = rápido.
p) Si **velocidad** = alta y **distancia** = junto entonces **frenar** = nulo.
q) Si **velocidad** = alta y **distancia** = cerca entonces **frenar** = mediana menos.
r) Si **velocidad** = alta y **distancia** = alejado entonces **frenar** = mediana.
s) Si **velocidad** = alta y **distancia** = lejos entonces **frenar** = mediana más.
t) Si **velocidad** = alta y **distancia** = retirado entonces **frenar** = rápido.
u) Si **velocidad** = máxima y **distancia** = junto entonces **frenar** = nulo.
v) Si **velocidad** = máxima y **distancia** = cerca entonces **frenar** = mediana menos.
w) Si **velocidad** = máxima y **distancia** = alejado entonces **frenar** = mediana.
x) Si **velocidad** = máxima y **distancia** = lejos entonces **frenar** = mediana más.
y) Si **velocidad** = máxima y **distancia** = retirado entonces **frenar** = rápido.

3. Se ha llegado a la meta No Ir a 1). Si Éxito ir a 4.

4. Salir.

Conclusiones.

Conclusiones.

A continuación, se mencionan las conclusiones de este trabajo de investigación:

- Al realizar trabajos de indagación inicialmente en el algoritmo de amplitud además primero el mejor se logran detectar nodos cercanos a los objetos los cuales no se pueden visitar por la estructura física del móvil, por consiguiente, se plantea realizar una poda de los estos sitios.
- En el análisis del modelo se determina un conjunto de variables independientes que se denominan **distancia** también **velocidad**, por otro lado, se utilizaron estas variables para formular un par de conjuntos difusos partiendo de las variables independientes.
- Se formula un controlador difuso utilizando las variables independientes; basado en el modelo **Sugeno** de modo que se genera la variable dependiente llamada **frenar.**
- Al estudiar el controlador difuso se utilizaron las reglas de control para formular un nuevo algoritmo para él SLAM.
- Como resultado del estudio del modelo del controlador se puede determinar que es estable, en consecuencia, la lógica difusa influye de manera importante para lograr la exploración de un mapa.

Además de las conclusiones anteriores este trabajo de investigación se ha presentado en un simposio, más aún se logró publicar en una revista científica por otra parte actualmente se encuentra en revisión de una revista indexada, a continuación, se describe este proceso.

- En una primera etapa se realizó un reporte científico de este trabajo, el cual fue aceptado entonces presentado en noviembre de 2020, en el primer Simposio Nacional de Inteligencia Artificial e Industria 4.0 de la Universidad del Valle de Atemajac, Zapopan, Jalisco, para ilustrar en la figura cuarenta y cuatro se tiene el nombre de la presentación en el simposio, así como la fecha además la hora de la exposición.

Figura 44. Primer simposio Nacional de Inteligencia Artificial e Industria 4.0.

- Acto seguido ha sido aceptado por otro lado publicado el artículo Luna, D. “Metodología para controlar un robot móvil con lógica difusa”. Revista Research in Computing Science

(IPN), (Ciudad de México), (149), 261–270 ISSN: 1870-4069 [27]. Esta publicación se puede considerar en la figura cuarenta y cinco.

Figura 45. Research in Computing Science.

- Finalmente se encuentra en revisión, una versión extendida con el mismo tema, en la revista **Computación y Sistemas** (CyS) (UNAM) (IPN) la cual es una revista indexada por el padrón de revistas del **CONACYT**. Finalmente se puede identificar esta revista en la figura cuarenta y seis.

Figura 46. Computación y sistemas.

Trabajo futuro.

De este modo el trabajo de investigación se ha dirigido al mapeo en dos dimensiones el cual se representa en el plano cartesiano (x, y) también se realiza la exploración de forma horizontal, por lo tanto, se muestra en la parte izquierda de la figura cuarenta y siete, de la misma forma la exploración en tres dimensiones es la que se utiliza en los vehículos aéreos no tripulados (popularmente llamados drones), por consiguiente, este trabajo podrá servir al momento de aplicar el SLAM en este tipo de dispositivos. Luego, para lograr el mapeo en tres dimensiones, también se plantea ampliar este trabajo de investigación utilizando dos árboles de búsqueda. Además, estos árboles deberán interactuar de forma perpendicular. En primer lugar, utilizar un árbol para explorar el plano cartesiano (x, y); por otro lado, se realizará la exploración horizontal, igualmente el segundo árbol se deberá utilizar para realizar el mapeo en el plano cartesiano (y, z); por consiguiente, este deberá de trabajar perpendicularmente al eje (x, y). Entonces, cada árbol tendrá sus conjuntos difusos también estructuras de datos independientes. Para concluir al lograr unir los dos árboles, se alcanzará el objetivo deseado de realizar el mapeo en tres dimensiones en el plano cartesiano (x, y, z), parte derecha de la figura cuarenta y siete.

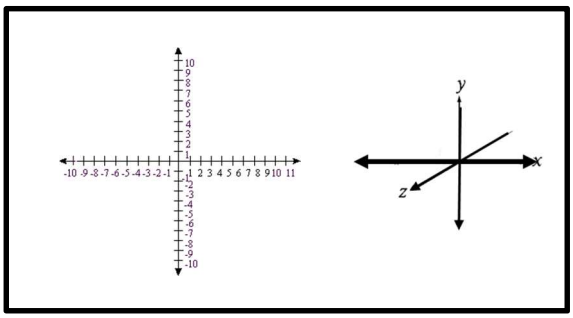


Figura 47. Plano cartesiano (x, y) y Plano cartesiano (x, y, z).

Bibliografía.

[1] Takagi, T.; Sugeno, M., IEEE Transactions on System, Man, and Cybernetics 1985, 15, 116-132.

[2] Baturone, A. O. (2001). Robótica manipuladores y robots móviles. Barcelona: MARCOMBO, S.A.

[3] Moravec, H. P. (1983). Robots and Intelligence. Obtenido de https://frc.ri.cmu.edu/~hpm/project.archive/robot.papers/1983/robage.mss.

[4] John Bares, W. (1988). cyberneticzoo.com. Recuperado el 23 de 06 de 2019, de http://cyberneticzoo.com/walking-machines/1988-91-ambler-john-bares-william-red-whittaker-american/

[5] IEE Xplore Digital Library. (01 de 02 de 1986). Recuperado el 23 de 06 de 2019, de https://ieeexplore.ieee.org/document/1087073

[6] Durrant-Whyte, H., Rye, D., & Nebot, E. (1995). Localization of auto-matic guided vehicles. The 7th International Symposium (ISRR'95).

[7] Corke, P. I., & Trevelyan:, J. (1999). dblp computer science bibliography. Sidney, Australia.

[8] Pons, J. V. (2012). Localización y generación de mapas del entorno (SLAM) de un robot por medio de una Kinect. Valencia, España: Universidad Politécnica de Valencia.

[9] Asker Zadeh, L. A. (1965). Fuzzy sets. Information and Control, 338-353.

[10] Olivas, J. (24 de 04 de 2018). Zadeh (D.E.P), la lógica borrosa y el análisis de datos masivos. Obtenido de Universidad Internacional de Valencia: https://www.universidadviu.com/zadeh-d-e-p-la-logica-borrosa-analisis-datos-masivos/

[11] Mamdani, E. H., Application of fuzzy algorithms for control of simple dynamic plant, Academic Press: Neva York, 1974.

[12] Metro de Sendai, Japón. (1987). Recuperado el 23 de 06 de 2019, de http://mapa-metro.com/es/japon/sendai/sendai-subway-mapa.htm

[13] Olivas, J. (24 de 04 de 2018). Zadeh (D.E.P), la lógica borrosa y el análisis de datos masivos. Obtenido de Universidad Internacional de Valencia: https://www.universidadviu.com/zadeh-d-ep-la-logica-borrosa-analisis-datos-masivos/

[14] Hernández Sampieri, R. (2014). Metodología de la investigación. Ciudad de México: Mc Graw Hill.

[15] Rusell, S., & Norving, P. (2004). Inteligencia Artificial. Un enfoque Moderno. Segunda Edición. Pearson & Prentice Hall: Madrid.

[16] Ponce Cruz, P. (2011). Inteligencia Artificial con aplicaciones a la Ingeniería. Ciudad de México: AlfaOmega.

[17] McCarthy, J. (1956). Padre de la Inteligencia Artificial. Recuperado el 21 de 06 de 2019, de https://www.bbvaopenmind.com/tecnologia/inteligencia-artificial/el-verdadero-padre-de-la-inteligencia-artificial

[18] Rifkin, J. (2011). The Third Industrial Revolution. New York: Palgrave Macmillan.

[19] Berzal, Fernando Recuperado (13 de junio de 2020). DECSAI Departamento de Ciencias de la computación e Inteligencia Artificial. Universidad de Granada.

[20] Real de León, R., Vargas Rubio, J., & Flores Enríquez, M. (26 de 07 de 2020). Heurística. Obtenido de www.arquepoetica.azc.uam.mx/escritos/heuristica.html

[21] Bueno, Antonio (13 de enero de 2020). www.portaleso.com Obtenido de www.portaleso.com/web_robot_3/robot_indice-html

[22] Klir, G. J., & Yuan, B. (1995). Fuzzy Sets and Fuzzy Logic. Theory and Applications. Prentice Hall PTR.

[22] uam. (26 de 07 de 2020). campusvirtual. Obtenido de www.campusvirtual.cua.uam.mx/material/tallerm/10_Distancia_entre_dos_puntos_html/index.html

[23] debé. (26 de julio de 2020). tecnología 4eso. Obtenido de www.edebe.com/educacion/documentos/830552-8-529-103947_UD07_Tecno%204%20Bessemer.pdf

[24] Scribbler®, R. (2019). Paralax inc. Recuperado el 21 de junio de 2019, de https://www.parallax.com/product/28333

[25] Morales Jiménez, F. J. (2012). Reporte de Robot Scribbler®. Puebla.

[26] Paralax inc. (29 de 06 de 2019). Recuperado el 29 de 06 de 2019, de Basic Stamp Discovery Kit USB: https://www.parallax.com/product/27807

[27] Luna, D., Zenteno, A., Santiago, M., Romero, Y., Pérez, J., & Rubín, G. (2020). Metodología para controlar un robot móvil con lógica difusa. Research in Computing Science, 261-270.

Anexos.

Anexo A.

Simulación de las reglas difusas de control de temperatura en Matlab.

Para utilizar el Toolbox de lógica difusa de Matlab, en primer lugar, escriba en la ventana de comandos **Fuzzy**. Posteriormente se abrirá el editor FIS en esta área del software se deben de implementar las funciones de membrecía tanto de entrada es decir temperatura como de salida sea válvula, que se muestra en la figura cuarenta y ocho.

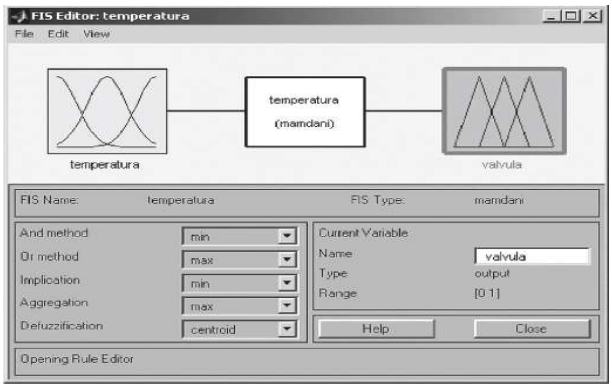


Figura 48. Editor FIS de Matlab.

El editor FIS tiene la opción para elegir el método de defuzzyficación que se desea utilizar, como se muestra en la figura cuarenta y nueve.

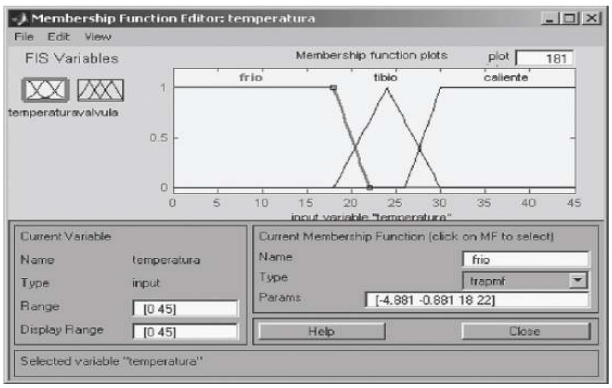


Figura 49. Función de membrecía.

Dando un doble click en la función de membrecía de entrada también de salida se pueden generar las funciones de membrecía, que se muestran en la figura cincuenta.

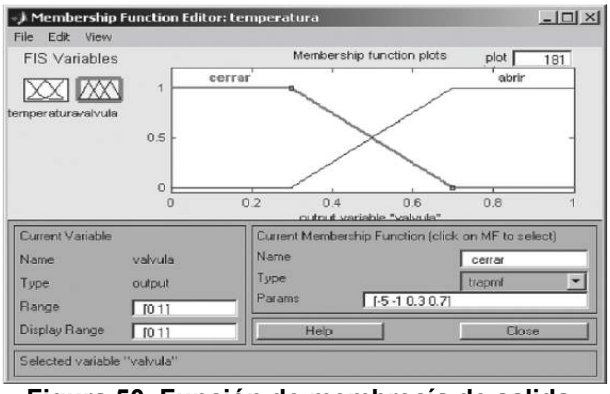


Figura 50. Función de membrecía de salida.

En el Editor FIS dando doble clic en la sección “Mandani” (se puede escoger el tipo de controlador que se desea trabajar) se introducen las reglas del controlador finalmente en **Ver Reglas** se puede ver la salida de fuzzyficada que finalmente se muestra en la figura cincuenta y uno.

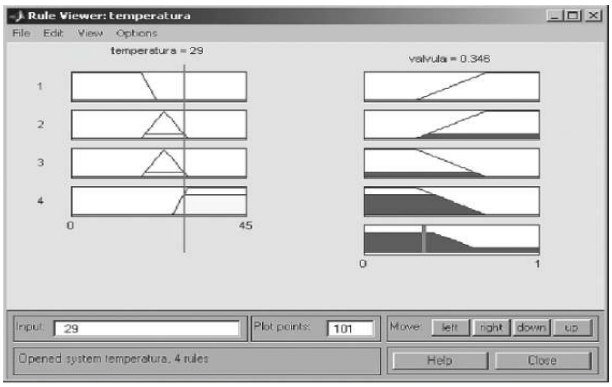


Figura 51. Reglas y defuzyficación.

Glosario.

Vértice o nodo: Se representará por una circunferencia, que podrá estar etiquetada. Vértice es el nombre formal en matemáticas. Para las técnicas que se utilizaran, normalmente se usará el término nodo.

Árbol trinario: Un árbol trinario tal que el grado de todos sus nodos no es superior a cuatro. Asimismo, se definen que cada nodo podrá tener tres hijos como máximo, se definen como hijo izquierdo, hijo central e hijo derecho.

Arista: Si A y B son dos nodos distintos,

- Una arista se define como un conjunto de dos nodos de la forma {A, B}. Su representación será con una línea que une los dos nodos.

Se dice que dos nodos unidos por un arco son **adyacentes**.

Grado de un vértice (nodo): Número de aristas (arcos) que inciden en el vértice (nodo).

Camino: Es una sucesión finita de nodos asimismo de aristas alternativamente, de manera que entre dos nodos consecutivos hay una arista.

Longitud de un camino: Número de aristas que tiene. Si la arista está etiquetada con números, a esta etiqueta se le suele llamar longitud de la arista, la longitud total del camino será de las etiquetas de las aristas.

Camino simple o ruta: Son aquellos que no pasan dos veces por el mismo nodo.

Camino cerrado: Si el último nodo del camino es el mismo que el primero. Formalmente, dados los nodos de un camino $(n_1,\ldots,n_k)$, donde $n_1 = n_k$ y $k \geq 1$, y todos los nodos son distintos n_1 y n_k.

Circuito: Camino cerrado en el que no se repite ninguna arista.

Camino abierto: Cuando en un camino, el último nodo no es el mismo que el primero.

Grafo conexo: Cuando entre dos nodos cualesquiera hay al menos un camino. Dicho de otra forma, hay un camino para cada dos nodos cualesquiera del grafo. Estos se pueden clasificar en:

- **Grafo simplemente conexo o poli árbol**: Cuando para cada dos nodos cualesquiera hay solo un camino y solo uno.
- **Grafo múltiplemente conexo**: Para algún par de nodos hay dos o más caminos entre ellos (como consecuencia de la definición, si hay un camino cerrado, es múltiplemente conexo).

Árbol: Un grafo conexo que no tiene ciclos.

Un árbol tiene las siguientes características:

- Para n nodos, hay exactamente n-1 aristas.
- Si se añade una arista, entonces se produce un ciclo.
- Si se elimina una arista, entonces deja de ser conexo.

Árboles (dirigidos): Son poli árboles en los que cada nodo tiene exactamente un padre. Sus propiedades son:

- Existe un único nodo que no tiene antepasados además es antepasado de todos los demás. Se denomina raíz.
- Esta raíz no tiene ningún padre. El resto de nodos tiene exactamente un padre. Los nodos que tienen el mismo padre son hermanos.

A partir de la definición se deduce que, para cualquier nodo del árbol, existe un camino desde este nodo hasta el nodo raíz. A este camino se le llama rama del árbol (recordar que es antepasado de todos los nodos igualmente ese camino tiene que existir).

Arboles con raíz: Un árbol con raíz es un árbol (no dirigido), donde uno de los nodos se considera como raíz. Como es un árbol especial en el que estamos definiendo un orden parcial

para los nodos, podemos utilizar los términos padres, hijo, hermano también terminal en el mismo sentido que en los árboles dirigidos. En la representación gráfica de un árbol dirigido a sí mismo un árbol con raíz, la única diferencia serán las flechas de sus arcos. Así se dibujará el nodo raíz en la parte superior. Por debajo de él se dibujarán todos sus hijos, unidos por una arista dirigida o no según el árbol que se trate. Esto es lo que forma un nivel del árbol. Además, se repetirá hasta que se llegue a todos los nodos terminales.

Pila: Una pila es una estructura de datos en la que el modo de acceso a sus elementos es de tipo **LIFO** (del inglés Last In First Out) que permite almacenar además recuperar datos.

Cola: Una cola es una estructura de datos, caracterizada por ser una secuencia de elementos en la que la operación de inserción **push** se realiza por un extremo también la operación de extracción **pop** por el otro. También se le llama estructura **FIFO** (del inglés First In First Out), debido a que el primer elemento en entrar será también el primero en salir.