

Navegación autónoma de robots utilizando herramientas de inteligencia computacional

Ing. Guido M. Sánchez

Facultad de Ingeniería y Ciencias Hídricas
Universidad Nacional del Litoral

Resumen En esta tesis se pretende estudiar alternativas para resolver el problema de navegación de robots autónomos móviles en ambientes desconocidos. Se tendrá en cuenta la dinámica de un robot Lego Mindstorms® y se realizará un simulador que se comporta de manera realista. Se desarrollará un sistema de navegación haciendo uso de lógica difusa. El sistema de navegación combina datos obtenidos mediante sensores de ultrasonido para planear el movimiento del robot. Para ello, hace uso de tres componentes: *i*) seguir objetivo: que determina el conjunto de direcciones deseadas a partir de las posiciones actuales del robot y del objetivo, *ii*) evitar obstáculos: determina el conjunto de direcciones posibles a partir de los datos obtenidos por los sensores de ultrasonido del robot y *iii*) dirección posible: combina la información provista por los dos componentes anteriores para determinar una dirección en que se podrá mover el robot. Dada una dirección posible, el controlador modulará la velocidad en función de la proximidad a obstáculos y traducirá esta velocidad deseada en términos de voltajes de entrada para el sistema. El rendimiento y comportamiento del sistema de navegación propuesto fue evaluado mediante simulaciones computacionales.

1. Introducción

El objetivo de esta tesis es desarrollar un *sistema de navegación autónoma para robots móviles en ambientes desconocidos*. Se requiere que el robot sea capaz de dirigirse desde un punto inicial A hasta un punto final B, encontrando un trayecto que una dichos puntos sin intervención humana ejecutando sus acciones de manera correcta y confiable a pesar de la dinámica del sistema y las incertidumbres inherentes del ambiente de trabajo [3], [4]. El ambiente es completamente desconocido y la única fuente de datos son los sensores de ultrasonido. Además de la información de carácter local provista por los sensores, el robot conoce su posición y orientación en todo momento así como la posición del punto final B y con estos datos debe ser capaz de llegar a su objetivo.

Se tendrá en cuenta la dinámica del robot, donde se simulará un robot Lego Mindstorms®. Esto implica que el robot se comporta como un cuerpo con una masa y dimensiones propias, lo que restringe algunos movimientos tales como giros bruscos, frenar de manera instantánea, entre otros. Por ello, es necesario que el sistema de navegación incorpore la dinámica del sistema, garantice la controlabilidad del robot en todo momento y que sea capaz de generar los comandos de

control en tiempo real, siendo potencialmente problemático cualquier retraso en la generación de dichos comandos, lo cual podría llevar a situaciones que pongan en peligro al robot o al entorno del mismo.

Al intentar cumplir con los objetivos expuestos, realizamos las siguientes restricciones: *i)* se considera la navegación de un robot móvil único, *ii)* el robot no posee memoria y no genera mapas, por lo cual no puede saber si pasó más de una vez por un mismo lugar, y *iii)* se considerarán ambientes de interiores con terreno llano y se supone que no existen deslizamientos entre las ruedas del robot y el piso.

Esta tesis se estructura de la siguiente manera: en el capítulo 2 se presenta el robot a modelar y capturan datos del mismo, considerando la dinámica del sistema para desarrollar un sistema de navegación. Se considera al robot real como un conjunto de subsistemas y se estudian las ecuaciones de los mismos, desarrollando así un modelo matemático que se ajusta al robot real. Una vez validado el modelo, se presenta una introducción sobre lógica difusa y los sistemas difusos en el capítulo 3, prosiguiendo al desarrollo del sistema de navegación en el capítulo 4. El capítulo 5 muestra los resultados obtenidos por el sistema de navegación desarrollado. Finalmente, el capítulo 6 incluye las conclusiones y posibles trabajos a futuro.

2. Modelado

2.1. El robot

El robot que será modelado es un Lego Mindstorms NXT ®. Este robot puede controlar hasta cuatro sensores (pudiendo utilizarse más sensores mediante el añadido de un multiplexor) y tres motores. Los servomotores permiten la detección de giros de la rueda gracias a un encoder que posee una resolución de 360° por revolución. El sensor de ultrasonido es capaz de detectar objetos que se encuentran de 0 a 255 cm. Se utilizarán sólo los sensores de ultrasonido para medir distancias y los encoders incluidos en los servomotores, supondremos que poseemos cinco sensores de ultrasonido, los cuales serán ubicados tal como se ve en la figura 1a y se hará la suposición que cada sensor cubre un área de 36°, logrando de esta manera una cobertura total de 180°.

2.2. Control diferencial de dirección

El control diferencial de dirección es posiblemente una de las configuraciones más comunes utilizada en robots móviles. Tal como se aprecia en la figura 1b, consiste en dos ruedas principales montadas sobre un mismo eje, cada una de las cuales está conectada a su propio motor, logrando proveer conjuntamente las funciones de tracción y de dirección.

La figura 1b esquematiza la configuración de control diferencial, donde el robot rota alrededor de un punto que se encuentra en la intersección de la proyección de los ejes de las ruedas motrices y se denomina *Eje de Rotación Instantánea*

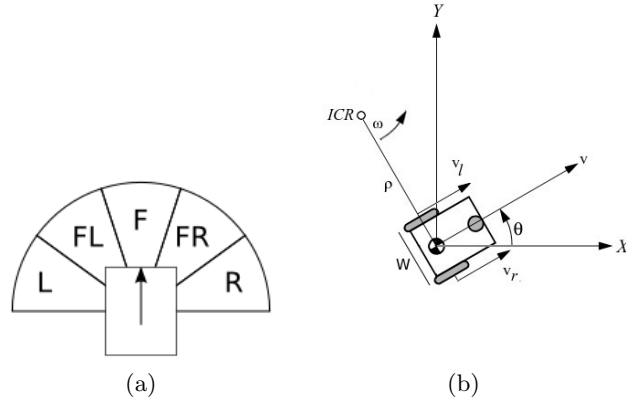


Figura 1: (a) Sensores del robot y (b) Parámetros de un robot con control diferencial de dirección.

o ICR (del inglés: Instantaneous Center of Rotation). Variando las velocidades de cada rueda se puede variar la posición del punto de rotación, determinando distintas trayectorias. En cada instante, se debe cumplir que la rueda izquierda y derecha sigan un trayecto que se mueve alrededor del ICR a la misma velocidad angular ω . De ésta manera, guiándonos por la figura 1b, podemos escribir:

$$v_r = \omega (\rho + w/2), \quad v_l = \omega (\rho - w/2) \quad (1)$$

donde w es la distancia del eje que une las dos ruedas, v_r y v_l son las velocidades de las ruedas derecha e izquierda, respectivamente y ρ es la distancia desde el ICR al punto medio entre las ruedas. En cualquier instante, se puede resolver para ρ , ω y v :

$$\rho = \frac{w}{2} \frac{v_r + v_l}{v_r - v_l}, \quad \omega = \frac{v_r - v_l}{w}, \quad v = \omega \rho = \frac{1}{2} (v_r + v_l) \quad (2)$$

Hay un número de casos de interés. Si $v_l = v_r$, entonces el radio ρ es infinito y el robot se mueve en línea recta. Si $v_l = -v_r$, entonces el radio es cero y el robot rota sobre su eje de rotación vertical. Para otros valores de v_l y v_r , el robot no se mueve en línea recta, sino que sigue una trayectoria curva alrededor de un punto a distancia ρ del centro del robot, cambiando tanto la posición como la orientación del robot.

2.3. Cinemática del robot

En la configuración de control diferencial, las variables que afectan la velocidad y dirección son las velocidades de las ruedas: v_l y v_r . Sea el robot un cuerpo libre moviéndose en el plano \mathbf{xy} , las coordenadas del punto (x, y) definen la posición global del robot con respecto al sistema de coordenadas. El ángulo θ que

forma la línea perpendicular al eje de las ruedas y que pasa por el punto (x, y) con el eje positivo x representa la orientación del robot. Las tres variables que definen la configuración geométrica del robot en cualquier momento son (x, y, θ) .

Supongamos que el punto (x, y) en el robot se mueve a una velocidad lineal v , mientras que el robot tiene una velocidad angular de ω . Podemos escribir las componentes de velocidad del punto (x, y) y el grado de cambio en la orientación como:

$$\dot{x} = v \cos \theta, \quad \dot{y} = v \sin \theta, \quad \dot{\theta} = \omega \quad (3)$$

Utilizando las ecuaciones 2 y 3, podemos expresar las ecuaciones cinemáticas del robot como sigue:

$$\dot{x}(t) = \frac{r}{2} [\omega_r(t) + \omega_l(t)] \cos(\theta(t)) \quad (4)$$

$$\dot{y}(t) = \frac{r}{2} [\omega_r(t) + \omega_l(t)] \sin(\theta(t)) \quad (5)$$

$$\dot{\theta}(t) = \frac{r}{w} [\omega_r(t) - \omega_l(t)] \quad (6)$$

donde r es el radio de las ruedas del robot. Con esta ecuación logramos expresar el movimiento del robot en función de las velocidades angulares ω_l y ω_r de las ruedas.

2.4. Movimiento rotacional

El movimiento de rotación nos permitirá modelar el comportamiento de las ruedas del robot. Cuando se aplica un torque a un cuerpo con momento de inercia J (ver figura 2a) teniendo en cuenta el efecto retardante de la fricción viscosa B , la ecuación de movimiento es

$$T(t) = T_J(t) + T_B(t) = J \frac{d^2\theta(t)}{dt^2} + B \frac{d\theta(t)}{dt} \quad (7)$$

Los servomotores suelen brindar una velocidad de rotación máxima pero un torque relativamente bajo. Para resolver esta falencia, puede utilizarse un tren de engranajes para intercambiar velocidad de rotación por torque. La figura 2b representa un tren de engranajes con fricción viscosa e inercia, donde T es el torque aplicado, T_1 y T_2 son los torques transmitidos y B_1 y B_2 son los coeficientes de fricción viscosa respectivos. Al ser la fricción viscosa B_1 despreciable, la ecuación del torque T puede ser escrita como

$$T(t) = T_1(t) = \frac{N_1}{N_2} T_2(t) = \frac{N_1}{N_2} J_2 \frac{d^2\theta_2(t)}{dt^2} + \frac{N_1}{N_2} B_2 \frac{d\theta_2(t)}{dt} \quad (8)$$

Donde observamos que la ecuación 8 del movimiento rotacional con un tren de engranajes no es más que la ecuación 7 multiplicada por la relación de engranajes

N_1/N_2 . Teniendo en cuenta la ecuación 3, podemos reescribir la ecuación 8 en función de la velocidad angular ω y despejar $d\omega_2(t)/dt$,

$$\frac{d\omega_2(t)}{dt} = \frac{N_2}{N_1} \frac{1}{J_2} T(t) - \frac{N_1}{N_2} \frac{B_2}{J_2} \omega_2(t) \quad (9)$$

donde se puede apreciar la relación entre la velocidad angular y el torque del tren de engranajes.

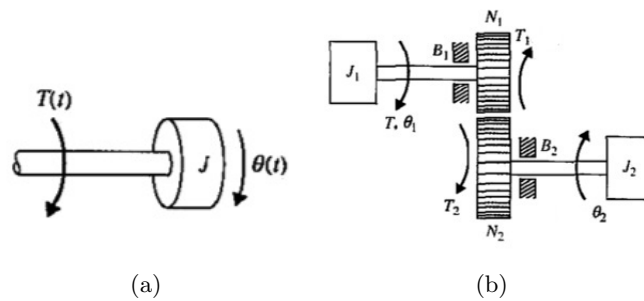


Figura 2: (a) Sistema torque-inercia, (b) Tren de engranajes con fricción e inercia.

2.5. Motor de corriente continua

Para el estado del motor de corriente continua se utiliza la ecuación del modelo de un circuito RL , tal como se ve en la figura 3 y cuya ecuación esta descripta por

$$V_{in}(t) = L \frac{dI(t)}{dt} + RI(t) \quad (10)$$

Teniendo en cuenta que el torque es $T(t) = k_T I(t)$, podemos reemplazar la intensidad de corriente por el torque y despejar $dT(t)/dt$

$$\frac{dT(t)}{dt} = \frac{k_T}{L} V_{in}(t) - \frac{R}{L} T(t) \quad (11)$$

la cual refleja la relación entre el torque producido por el motor y el voltaje de entrada.

2.6. Concatenación de sistemas

Utilizando las ecuaciones 4, 5, 6, 9 y 11 podemos armar el modelo global del comportamiento del robot en espacio de estado, el cual está descripto por

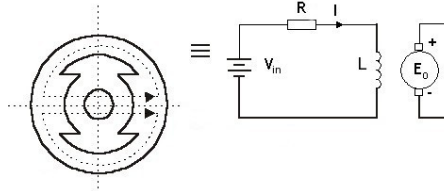


Figura 3: Circuito para un motor de corriente continua.

$$\begin{aligned}
 \dot{x}_1(t) &= \frac{r}{2} (x_5(t) + x_4(t)) \cos(x_3(t)) \\
 \dot{x}_2(t) &= \frac{r}{2} (x_5(t) + x_4(t)) \sin(x_3(t)) \\
 \dot{x}_3(t) &= \frac{r}{w} (x_5(t) - x_4(t)) \\
 \dot{x}_4(t) &= \frac{N_2}{N_1} \frac{1}{J_{2l}} x_6(t) - \frac{B_{2l}}{J_{2l}} x_4(t) \\
 \dot{x}_5(t) &= \frac{N_2}{N_1} \frac{1}{J_{2r}} x_7(t) - \frac{B_{2r}}{J_{2r}} x_5(t) \\
 \dot{x}_6(t) &= \frac{k_T}{L} V_l(t) - \frac{R}{L} x_6(t) \\
 \dot{x}_7(t) &= \frac{k_T}{L} V_r(t) - \frac{R}{L} x_7(t)
 \end{aligned} \tag{12}$$

donde $x_1 = x$, $x_2 = y$, $x_3 = \theta$, $x_4 = \omega_l$, $x_5 = \omega_r$, $x_6 = T_l$ y $x_7 = T_r$ son las variables de estado. Esta ecuación representa la dinámica completa del robot a partir de los voltajes de entrada V_l y V_r de cada motor y tiene como salida la posición x , y y la orientación θ del centro de masa del robot.

2.7. Identificación de parámetros

Teniendo en cuenta las ecuaciones 4, 5, 6, 9 y 11 vemos que hay parámetros que se pueden medir y otros que no. Para identificarlos se realizaron experimentos para medir la respuesta de los motores a una función escalón en V_l y V_r , registrando el ángulo descrito por las ruedas en el intervalo de tiempo. La dinámica de esta parte del robot es descrita por ecuaciones diferenciales de primer orden (ver secciones 2.4 y 2.5) que pueden ser escritas en su forma general de la siguiente manera

$$\tau \frac{dy}{dt}(t) + y(t) = kf(t), \tag{13}$$

donde τ es la constante de tiempo del sistema y k su ganancia. La función $f(t)$ es la entrada del sistema. Con la respuesta al escalón obtenida, se identificaron los parámetros τ y k utilizando un algoritmo basado en la minimización del error cuadrático acumulado entre la respuesta del sistema y la respuesta simulada del modelo.

Este procedimiento se realizó en dos etapas: *i*) para el motor sin carga a los fines de identificar únicamente los parámetros del motor (τ_M y k_M) y *ii*) con todo el robot a los fines de identificar los parámetros cinemáticos (τ_C y k_C). Los parámetros obtenidos fueron

$$\tau_M = 0,025\text{s}, k_M = 90^\circ/\text{V}, \tau_C = 0,04\text{s}, k_C = 1^\circ/\text{V} \quad (14)$$

Teniendo en cuenta las igualdades 14, el sistema de ecuaciones 12 puede ser reescrito de la siguiente manera

$$\begin{aligned} \dot{x}_1(t) &= \frac{r}{2} (x_5(t) + x_4(t)) \cos(x_3(t)) \\ \dot{x}_2(t) &= \frac{r}{2} (x_5(t) + x_4(t)) \sin(x_3(t)) \\ \dot{x}_3(t) &= \frac{r}{w} (x_5(t) - x_4(t)) \\ \dot{x}_4(t) &= \frac{k_C}{\tau_C} x_6(t) - \frac{1}{\tau_C} x_4(t) \\ \dot{x}_5(t) &= \frac{k_C}{\tau_C} x_7(t) - \frac{1}{\tau_C} x_5(t) \\ \dot{x}_6(t) &= \frac{k_M}{\tau_M} V_l(t) - \frac{1}{\tau_M} x_6(t) \\ \dot{x}_7(t) &= \frac{k_M}{\tau_M} V_r(t) - \frac{1}{\tau_M} x_7(t) \end{aligned} \quad (15)$$

donde se observa el empleo de los parámetros τ_M, k_M, τ_C, k_C junto con los valores conocidos r y w .

2.8. Validación de parámetros

El modelo fue validado considerando dos casos: *i*) movimiento en línea recta por un lapso de 10 segundos a velocidad máxima y *ii*) movimiento alrededor de uno de los ejes por un lapso de 5 segundos a velocidad máxima. Para la realización de ambos experimentos se programaron rutinas utilizando el lenguaje de programación NXT-G y se realizó el registro de los resultados añadiendo un fibrón al chasis del Mindstorms NXT. Una vez obtenidos estos datos, se programaron rutinas en MATLAB para comparar con datos simulados. Se utilizó la función `ode45` provista por MATLAB para integrar el sistema de ecuaciones diferenciales 15.

Para el caso del movimiento en línea recta a velocidad máxima, el robot se desplazó 2.15 m en el lapso de 10 segundos (figura 4). Al simular el mismo movimiento, observamos que los datos obtenidos se ajustan a los datos obtenidos experimentalmente, tal como se puede ver en la figura 4d.

En el segundo caso, el robot logra completar un giro de 360° sobre su eje izquierdo en 5 segundos, describiendo un radio de 13 cm. Los resultados de la simulación arrojan el mismo resultado en cuanto a la velocidad de giro, pero se observa una diferencia de 5 cm en el radio de giro. Esto se atribuye a que en el simulador se registra el radio de giro del centro de masa, mientras que

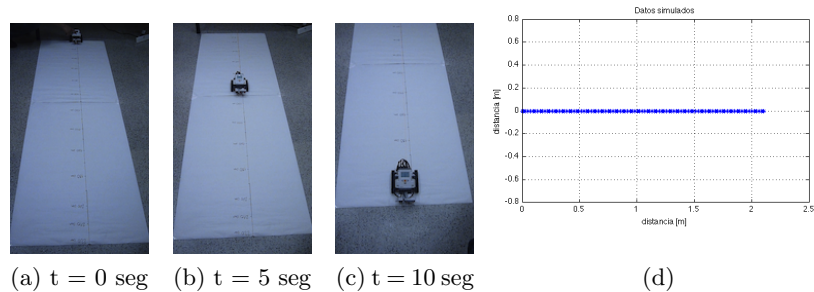


Figura 4: Desplazamiento del robot en línea recta por 10 segundos: (a-c) Datos reales, (d) Datos simulados.

en el experimento, el robot giró sobre la primer rueda, realizándose una traslación del centro de giro, dando lugar a la diferencia observada en el radio de la circunferencia.

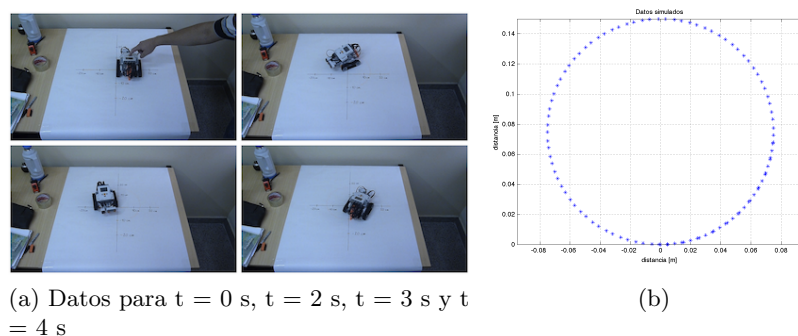


Figura 5: Desplazamiento del robot sobre una de sus ruedas por 5 segundos: (a) Datos reales, (b) Datos simulados.

3. Lógica difusa

La lógica difusa es reconocida como una herramienta adecuada para diseñar sistemas robustos que son capaces de lograr un rendimiento satisfactorio en presencia de perturbaciones en la entrada y de variabilidad en los parámetros. La clave de esta robustez se encuentra en el mecanismo interpolador utilizado por la

lógica difusa, donde se considera que entradas similares deben producir acciones similares. Sumado a esto, el formato de reglas y el uso de variables lingüísticas hacen que sea una herramienta de diseño adecuada para sistemas no lineales donde un modelo matemático preciso no puede ser obtenido fácilmente, pero existe conocimiento heurístico de manera que resulta sencillo escribir comportamientos simples y efectivos [5].

Estas características encajan con las necesidades de los robots autónomos móviles, donde: (i) generalmente no existe un modelo matemático del ambiente; (ii) los datos de los sensores son inciertos y en ocasiones, imprecisos; y (iii) la operación en tiempo real es esencial. Asimismo, el formato de reglas difusas hace que sea sencillo escribir comportamientos simples y efectivos para una gran variedad de tareas sin necesidad de utilizar modelos matemáticos complejos.

3.1. Sistema difuso

Un sistema difuso es un sistema experto ampliado que utiliza lógica difusa para poder operar con variables continuas. En la figura 6 podemos ver la estructura de un sistema difuso, que consta de tres bloques fundamentales: el fuzzificador, la máquina de inferencia y el defuzzificador. El fuzzificador calcula a qué conjuntos difusos pertenece la variable de entrada así como su grado de pertenencia. Al realizar este proceso se dice que se *etiqueta* dicha variable, asignándole un valor lingüístico a la misma. Una variable lingüística, como su nombre lo indica, es una variable cuyos valores son palabras, frases o conceptos. Por ejemplo, si la variable *velocidad* es tratada como una variable lingüística, sus valores pueden ser expresados como *rápido*, *un poco rápido*, *muy rápido*, *no muy rápido*, *muy muy rápido*, *algo rápido*, *más o menos rápido*, entre otros.

La máquina de inferencia utiliza un conjunto de reglas **if-then** para realizar una inferencia a partir de las variables lingüísticas. Estas reglas toman la forma

$$\text{if } x \text{ is } A \text{ then } z \text{ is } C \quad (16)$$

donde x es la variable de entrada, z es una variable de salida, A es una etiqueta definida en x y C es una etiqueta definida en z . Generalmente " x is A " se denomina el **antecedente**, mientras que " z is C " se denomina el **consecuente**. El antecedente describe una condición, mientras que el consecuente aplica una consecuencia. La máquina de inferencia relaciona los conjuntos difusos del antecedente con los del consecuente, a través de las etiquetas, realizando de esta manera un mapeo entre las entradas y salidas del sistema. A continuación, determina el grado de activación del consecuente de cada regla mediante el grado de pertenencia obtenido por el antecedente. Existen varios métodos para realizar este ajuste. En esta tesis se utilizará el método del mínimo, donde el conjunto difuso de salida es cortado a la altura correspondiente al mínimo grado de verdad del antecedente.

El defuzzificador recibe como entrada los conjuntos difusos que se obtuvieron al aplicar cada regla en la máquina de inferencia (consecuentes) y los combina para formar un único conjunto para cada variable de salida. Existen varios métodos

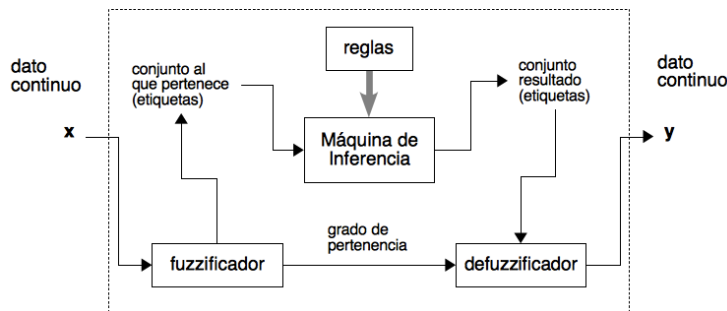


Figura 6: Modelo general de un sistema experto difuso

de composición. En esta tesis utilizaremos el método de máximos, que consiste en formar el conjunto tomando el máximo valor punto a punto a través de todos los conjuntos difusos asignadas a la variable de salida. Finalmente, una vez obtenido el conjunto combinado, resta convertirlo en un número. Existen varios métodos para realizar esta tarea, optando en este caso por el método de centroides, el cual genera la salida numérica calculando el centro de masa del conjunto combinado.

4. Sistema de navegación

El ambiente en el que se mueve el robot contiene obstáculos desconocidos, cuyas posiciones, dimensiones y formas se desconocen. El robot se desplaza sin deslizamientos, lo cual permite utilizar métodos de odometría para calcular su posición, de modo que el robot su posición en todo momento así como la posición del objetivo al que intenta llegar. Además de eso, la única otra fuente de información que posee el robot acerca del entorno en que se encuentra son sus sensores.

El sistema de navegación del robot consta de tres componentes principales: *i)* seguimiento del objetivo, *ii)* evitar obstáculos y *iii)* obtener dirección posible.

4.1. Seguimiento del objetivo

Este módulo es determina un conjunto de direcciones deseadas a partir de la posición actual del robot (x_r, y_r) y la posición del objetivo (x_t, y_t) . A partir de estos datos, calcula el ángulo mínimo ángulo de giro θ_d requerido para alinear el robot con el objetivo. Una vez obtenido el ángulo θ_d , este se traduce mediante reglas difusas (ver apéndice A) a un conjunto direcciones deseadas denominadas derecha, frente-derecha, frente, frente-izquierda e izquierda, siendo sus etiquetas R, FR, F, FL y L, respectivamente. Las reglas que realizan esta traducción son:

- IF ángulo-cercano-a -90, THEN dirección-deseada IS R.

- IF ángulo-cercano-a -45, THEN dirección-deseada IS FR.
- IF ángulo-cercano-a 0, THEN dirección-deseada IS F.
- IF ángulo-cercano-a 45, THEN dirección-deseada IS FL.
- IF ángulo-cercano-a 90, THEN dirección-deseada IS L.

4.2. Evitar obstáculos

El componente **evitar obstáculos** determina en qué dirección debe dirigirse el robot a fin de lograr evitar los obstáculos que aparezcan en el entorno. Para ello utiliza los datos obtenidos por los sensores y representa en qué grado una dirección está habilitada o no, midiendo el grado de membresía de cada distancia registrada a un conjunto difuso que denominaremos **cercanía** y que puede verse en la figura 7a. Debido a que cada sensor esta asociado con una etiqueta de dirección de (R: derecha, FR: frente-derecha, F: frente, FL: frente-izquierda, L: izquierda), se logra escalar cada etiqueta del conjunto de direcciones de acuerdo a la cercanía de obstáculos, determinando de esta manera el grado de *transitabilidad* de las cercanías del robot en cada instante.

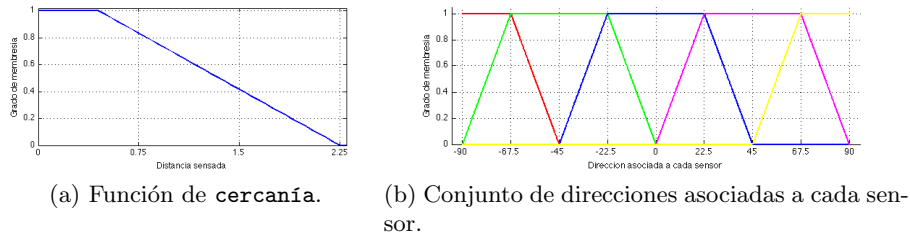


Figura 7: Funciones de membresía para el componente **evitar obstáculos**

4.3. Fusión de comandos y defuzzificación

Este módulo combina las conclusiones difusas de los componentes **seguimiento de objetivo** y **evitar obstáculos**. Se desea que la dirección final tenga en cuenta tanto la dirección deseada provista por el módulo de **seguimiento de objetivo** como la no deshabilitada provista por el módulo de **evitar obstáculos**. La fusión de comandos se realiza de la siguiente manera:

$$\theta_p = \theta_d \text{ AND NOT } \theta_{deshabilitada} \quad (17)$$

$$\theta_p = \theta_d \cap (1 - \theta_{deshabilitada}) \quad (18)$$

4.4. Modulación de velocidad

Debido a que este trabajo tiene en cuenta la dinámica propia del modelo real y sus limitaciones, tenemos como consecuencia: *i*) debido a la presencia de la inercia, no se puede frenar inmediatamente o realizar giros muy bruscos, *ii*) debido a la presencia de restricciones en las velocidades máximas ($0 \leq v \leq v_{max}$), el robot no puede ser frenado ni acelerado instantáneamente. En consecuencia, es necesario modular la velocidad máxima a fines de preservar la controlabilidad del robot, lo que se realiza en función de la distancia al obstáculo d_o o al objetivo d_t

$$d = \min[d_o, d_t] \quad (19)$$

Esta distancia es utilizada con una función difusa que define qué fracción de la velocidad máxima posible se puede aplicar en ese instante de tiempo. De modo que la velocidad máxima aplicada en cada instante de tiempo está dada por $\tilde{v}_{max} = \mu v_{max}$.

La forma de la función tiene en cuenta que una vez fijada una distancia *segura*, la velocidad puede ser la máxima. En tanto y en cuanto el robot estemos ante la presencia de obstáculos próximos al robot o cerca del objetivo, el robot deberá reducir progresivamente su velocidad para poder garantizar la capacidad de maniobra o de frenado del mismo.

4.5. Cálculo de voltajes de entrada

Dada una dirección θ_p y una velocidad máxima \tilde{v}_{max} deseadas, necesitamos saber qué combinación de voltajes V_l y V_r a aplicar. Para ello partiremos de la ecuación 6, asumiendo que

$$\dot{\theta} \approx \frac{\Delta\theta}{\Delta t}. \quad (20)$$

Teniendo en cuenta que el sistema difuso de navegación nos da como resultado cuánto debemos girar con respecto a la posición actual, podemos considerar que $\theta_p = \Delta\theta/\Delta t$. Entonces, la ecuación 6 puede ser reescrita como sigue

$$\theta_p = \frac{1}{w} [v_r(t) - v_l(t)]. \quad (21)$$

Como queremos que el robot se desplace a la máxima velocidad posible, una de las dos velocidades será fijada a su valor \tilde{v}_{max} , mientras que la otra será calculada para lograr el ángulo de giro θ_p . Esta condición conduce a las siguientes ecuaciones

$$\begin{aligned} \text{IF } \theta_p > 0, \quad \text{THEN } v_r &= \tilde{v}_{max} \text{ y } v_l = v_r - \theta_p w \\ \text{IF } \theta_p < 0, \quad \text{THEN } v_l &= \tilde{v}_{max} \text{ y } v_r = v_l + \theta_p w \end{aligned} \quad (22)$$

Una vez obtenidas las velocidades necesarias para ejecutar el movimiento, el paso siguiente es calcular los voltajes para obtener dichas velocidades. Como el período de muestreo del sistema de navegación es mucho mayor que la

dinámica del motor, podemos considerar que ésta está en estado estacionario ($T_S = 1\text{ s}$, $\tau_c = 0,04\text{ s}$), entonces

$$V_l = \frac{v_l}{\kappa}, \quad V_r = \frac{v_r}{\kappa} \quad (23)$$

donde $\kappa = \frac{\pi}{2} r$ es la ganancia del motor y r es el radio de la rueda del robot.

5. Resultados

En este capítulo se presentan resultados obtenidos mediante simulaciones del robot. A tal efecto, se desarrolló un simulador en MATLAB utilizando programación orientada a objetos, el cual permite cargar mapas en dos dimensiones y probar el comportamiento del robot y el sistema de navegación. Cada mapa consiste en una imagen en formato BMP de una resolución de 640×480 pixels. Se estableció que la zona coloreada de negro representa obstáculos que el robot debe sortear para llegar al objetivo, el cual está representado por una cruz de color rojo. Cada simulación ubica al robot en una posición inicial $(x_r; y_r; \theta_r)$ y al objetivo en una posición $(x_t; y_t)$. El trayecto recorrido por el robot se marca con puntos de color azul. Se midió el tiempo transcurrido entre que el robot comienza a moverse hasta que llega al objetivo con las funciones `tic` y `toc`.

5.1. Ejemplo 1

En este mapa de tipo *laberinto*, la configuración inicial del robot corresponde a $(x_r; y_r; \theta_r) = (5; 0,5; 0)$ y la posición del objetivo es $(x_t; y_t) = (0,5; 4,5)$. El sistema de navegación no presenta mayores inconvenientes, el robot llegó a la posición objetivo en 242,26 segundos como puede verse en la figura 8a.

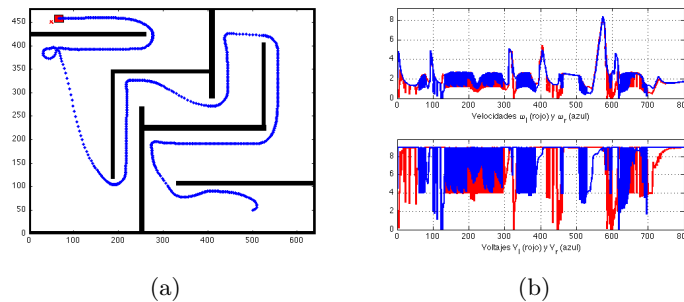


Figura 8: (a) Resultado del ejemplo 1, (b) Velocidades ω_l y ω_r y voltajes V_l y V_r .

La figura 8b muestra las velocidades ω_l y ω_r junto con los voltajes V_l y V_r aplicados a los motores, donde se nota claramente la modulación de voltajes

para girar abruptamente en el momento que se encuentra con obstáculos y puede observarse que las velocidades ω_l y ω_r son máximas en ausencia de obstáculos y disminuyen en presencia de ellos.

5.2. Ejemplo 2

En este caso se diseñó un mapa de paredes largas y alta simetría. La configuración inicial del robot corresponde a $(x_r; y_r; \theta_r) = (0,5; 2,4; 0)$ y la del objetivo a $(x_t; y_t) = (5; 2,4)$. La longitud de las paredes y las condiciones de simetría entre la forma del mapa, la posición del objetivo y la posición inicial del robot determinan un comportamiento cíclico, tal como se ve en la figura 9a.

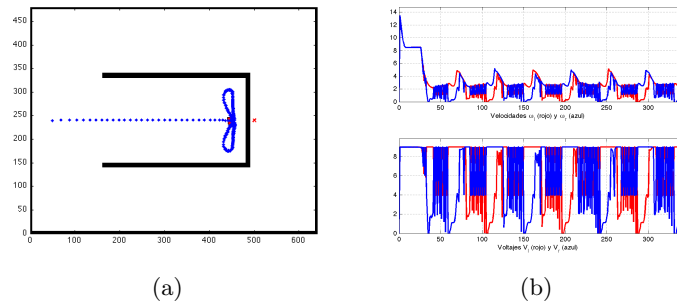


Figura 9: (a) Resultado del ejemplo 2, (b) Velocidades ω_l y ω_r y voltajes V_l y V_r .

En la figura 9b podemos observar que en conjunto con el comportamiento cíclico que presenta el robot, los voltajes aplicados se vuelven periódicos. La tendencia a entrar a bucles infinitos no está dada únicamente por el mapa, sino por una combinación entre la posición del robot, la posición del objetivo y la forma del obstáculo que el robot tiene enfrente en un tiempo determinado, lo cual genera mínimos locales que *atrapan* al robot en la zona.

6. Conclusiones

El objetivo principal de la tesis era desarrollar un sistema de navegación autónoma en ambientes desconocidos. Para ello se desarrolló un simulador realista que incorpora la dinámica un robot real. Esto trae como consecuencia que el robot a controlar no será capaz de realizar ciertas maniobras debido a las restricciones propias del sistema. Existen una gran cantidad de trabajos que se centran en un enfoque cinemático para desarrollar algoritmos de control y de navegación [1], [2], [7] y poca literatura tiene en cuenta la dinámica propia del sistema. En este aspecto, se considera positivo haber podido realizar simulaciones realistas

con resultados satisfactorios, lo que facilitará el paso a la implementación del sistema de navegación en un sistema real.

Gracias a la utilización de lógica difusa para desarrollar el sistema de navegación autónoma tenemos un controlador que es fácil de entender y simple de implementar. El sistema de navegación se comporta de manera determinística, lo cual es una característica sumamente deseable. No presenta inconvenientes en mapas donde se observa la presencia de espacios abiertos, mostrando dificultades en mapas del estilo de laberintos o en situaciones donde la geometría es muy particular. Sin embargo, teniendo en cuenta que el sistema no utiliza memoria, podemos afirmar que brinda buenos resultados, incluso en mapas donde inicialmente no se pensó que podría llegar a encontrar el objetivo.

El desempeño del sistema podría mejorarse si se añadiese una memoria de corto plazo, a fin de que tenga algún tipo de “recuerdo” de donde estuvo a fin de evitar bucles infinitos. Asimismo, podrían añadirse comportamientos, tal como: **rodear obstáculo, seguir pared, retroceder**, entre otros, a fin de poder atacar el problema de llegar al objetivo con más de una estrategia en caso de que el robot se encuentre atascado.

Referencias

1. W. Han, S. Baek, and T. Kuc. Genetic algorithm based path planning and dynamic obstacle avoidance of mobile robots. In *Systems, Man, and Cybernetics, 1997. 'Computational Cybernetics and Simulation'*, 1997 IEEE International Conference on, volume 3, pages 2747–2751. IEEE, 2002.
2. G. Huang, C. Tung, and J. Ciou. To achieve the path planning of mobile robot for a correct destination and direction using fuzzy theory. In *Industrial Electronics, 2009. ISIE 2009. IEEE International Symposium on*, pages 1737–1742. IEEE, 2009.
3. J.-C. Latombe. *Robot motion planning*. Kluwer international series in engineering and computer science: Robotics. Kluwer Academic Publishers, 1990.
4. V. Lumelsky. *Sensing, intelligence, motion: how robots and humans move in an unstructured world*. Wiley-Interscience, 2006.
5. A. Saffiotti. The uses of fuzzy logic in autonomous robot navigation. *Soft Computing-A Fusion of Foundations, Methodologies and Applications*, 1(4):180–197, 1997.
6. R. Siegwart and I. Nourbakhsh. *Introduction to autonomous mobile robots*. Intelligent robotics and autonomous agents. MIT Press, 2004.
7. X. Yang, M. Moallem, and R. Patel. A layered goal-oriented fuzzy motion planning strategy for mobile robot navigation. *Systems, Man, and Cybernetics, Part B: Cybernetics, IEEE Transactions on*, 35(6):1214–1224, 2005.
8. L. Zadeh. Fuzzy sets. *Information and Control*, 8(3):338 – 353, 1965.

A. Conceptos adicionales de lógica difusa

A.1. Conjuntos difusos

A pesar que los conjuntos clásicos son adecuados para varias aplicaciones y han probado ser una herramienta importante para las matemáticas y las ciencias de la computación, no reflejan la naturaleza de los conceptos y pensamientos

humanos, que tienden a ser abstractos e imprecisos. Por ejemplo, si deseáramos expresar matemáticamente el conjunto de velocidades rápidas como una colección de velocidades que superen los 120 km/h, podríamos escribir

$$A = \{x \mid x > 120\}, \quad (24)$$

esta definición no es intuitiva para representar nuestro concepto de “*velocidad rápida*”, ya que la naturaleza de los conjuntos clásicos clasificarían a una velocidad de 120.001 km/h como rápida, pero no a una velocidad de 119.999 km/h. Este tipo de distinción tan estricta contradice a la lógica que las personas estamos acostumbradas a utilizar día a día.

En un conjunto difuso la transición de “*pertenece al conjunto*” a “*no pertenece al conjunto*” es gradual y está caracterizada por *funciones de membresía* que brindan flexibilidad en el modelado de expresiones lingüísticas comúnmente utilizadas, tal como “el agua está caliente” o “la temperatura está muy alta”. Como señaló Lotfi Zadeh en su trabajo “Fuzzy Sets” [8], estas expresiones juegan un rol importante en el pensamiento humano, particularmente en los dominios del reconocimiento de patrones, la comunicación de información y la abstracción. Esencialmente, los conjuntos difusos proveen una manera natural de hacer frente al tipo de problemas en los que la fuente de imprecisión es la ausencia de criterios claros de pertenencia a conjuntos o clases.

A.2. Definiciones

Sea X un espacio de puntos (objetos), con un elemento genérico de X denotado por x . Por lo tanto, $X = \{x\}$. Un *conjunto difuso* A en X es caracterizado por una *función de membresía* $\mu_A(x)$, la cual asocia cada punto en X con un número real en el intervalo $[0, 1]$. El valor de $\mu_A(x)$ representa el “*grado de membresía*” de x en A . De esta manera, a medida que el valor de μ_A sea más cercano a la unidad, mayor será el grado de pertenencia de x en A . Cuando A es un conjunto clásico, su función de pertenencia puede tomar sólo dos valores: 0 y 1, con $\mu_A(x) = 1$ o 0 según x pertenezca o no pertenezca a A y en este caso, $\mu_A(x)$ se reduce a la *función característica* de un conjunto clásico A , [8].

A continuación introduciremos definiciones que involucren a los conjuntos difusos, las cuales son extensiones obvias de las definiciones correspondientes para conjuntos clásicos:

- **Igualdad:** Dos conjuntos difusos A y B son *iguales*, es decir $A = B$, si y sólo si $\mu_A(x) = \mu_B(x)$ para todo x en X .
- **Complemento:** El *complemento* de un conjunto difuso A , denotado por \tilde{A} es definido como

$$\mu_{\tilde{A}} = 1 - \mu_A(x) \quad (25)$$

- **Inclusión:** A está *incluido* en B (o, equivalentemente, A es un *subconjunto* de B , o A es *menor o igual* que B) si y solo si $\mu_A(x) \leq \mu_B(x)$. En símbolos

$$A \subset B \iff \mu_A(x) \leq \mu_B(x) \quad (26)$$

- Unión:** La *unión* de dos conjuntos difusos A y B con sus respectivas funciones de membresía $\mu_A(x)$ y $\mu_B(x)$ es un conjunto difuso C , escrito como $C = A \cup B$, cuya función de membresía está relacionada con las de A y B de la siguiente manera

$$\mu_C(x) = \text{Max}[\mu_A(x), \mu_B(x)], \quad x \in X \quad (27)$$

Una manera más intuitiva de definir la unión es la siguiente: La unión de A y B es el conjunto difuso más pequeño que incluye tanto a A como a B .

- Intersección:** La *intersección* de dos conjuntos difusos A y B con sus respectivas funciones de membresía $\mu_A(x)$ y $\mu_B(x)$ es un conjunto difuso C , escrito como $C = A \cap B$, cuya función de membresía está relacionada con las de A y B de la siguiente manera

$$\mu_C(x) = \text{Min}[\mu_A(x), \mu_B(x)], \quad x \in X \quad (28)$$

Como en el caso de la unión, una manera más intuitiva de definir la intersección de A y B es: La intersección de A y B es el conjunto difuso más grande que está incluido tanto en A como en B .

La figura 10 presenta gráficamente las operaciones de complemento, unión e intersección para dos conjuntos difusos A y B .

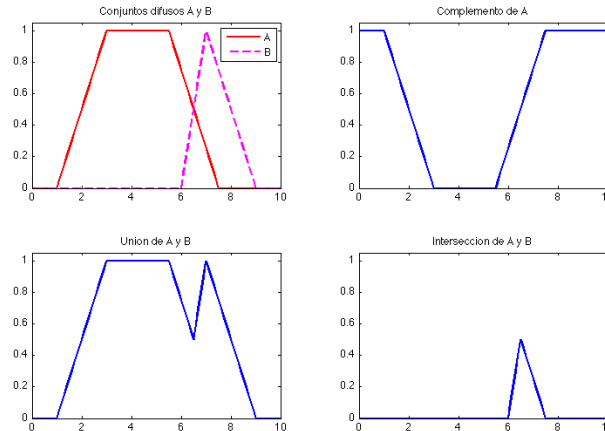


Figura 10: Operaciones con conjuntos difusos.

A.3. Funciones de membresía

Como se mencionó en la sección A.2, una función de membresía caracteriza a un conjunto difuso $A \subset X$ y está representada por una función cualquiera $\mu_A(x)$,

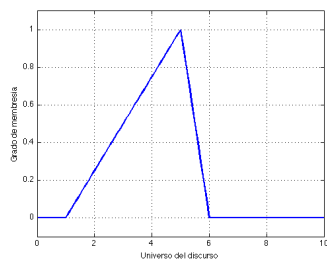
la cual asocia cada punto en X con un número real en el intervalo $[0, 1]$. El valor de $\mu_A(x)$ representa el “grado de membresía” de x en A y caracteriza a los miembros del conjunto. Esta definición nos permite utilizar funciones de membresía de cualquier forma, siempre y cuando satisfaga la condición de mapear un conjunto de números reales en el intervalo $[0, 1]$. Entre los tipos de funciones de membresía más utilizados están las triangulares, trapezoidales, Gaussianas, campanas y sigmoides. En esta tesis se utilizarán funciones de membresía triangulares y trapezoidales, por lo que pasaremos a detallarlas. Las figura 11a y 11b ilustran ejemplos de estos tipo de funciones de membresía.

- **Funciones de membresía triangulares:** Una función de membresía triangular está definida por tres parámetros $\{a, b, c\}$

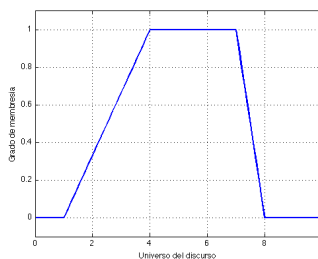
$$\text{triángulo}(x; a, b, c) = \begin{cases} 0, & x \leq a. \\ \frac{x-a}{b-a}, & a \leq x \leq b. \\ \frac{c-x}{c-b}, & b \leq x \leq c. \\ 0, & c \leq x. \end{cases} \quad (29)$$

- **Funciones de membresía trapezoidales:** Una función de membresía trapezoidal está definida por cuatro parámetros $\{a, b, c, d\}$

$$\text{trapecio}(x; a, b, c, d) = \begin{cases} 0, & x \leq a. \\ \frac{x-a}{b-a}, & a \leq x \leq b. \\ 1, & b \leq x \leq c. \\ \frac{d-x}{d-c}, & c \leq x \leq d. \\ 0, & d \leq x. \end{cases} \quad (30)$$



(a)



(b)

Figura 11: (a) Función de membresía triangular, con $a = 1$, $b = 5$ y $c = 6$. (b) Función de membresía trapezoidal con $a = 1$, $b = 4$, $c = 7$ y $d = 8$.