

Universidad de Ingeniería Y Tecnología

Global Academy



Introduction to Intelligent Control Systems

Proyecto 1

Palma Rodríguez, Diego Alonso V.

Instructor: Diego Benítez, Ph.D.

Lima - Peru

2021-1

1. Introducción

La navegación autónoma es un área importante de la robótica, pues involucra el conocimiento de diversas disciplinas, tales como mecánica, electrónica y ciencias de la computación, con el fin de controlar un robot móvil en un entorno definido con cierto grado de autonomía. En ese sentido, me resulta de gran interés proponer un proyecto enfocado en esa rama.

El presente proyecto busca diseñar e implementar un sistema de lógica difusa para determinar las velocidades lineales de las ruedas izquierda (v_l) y derecha (v_r) de un robot móvil (ver Fig.1), a partir de la distancia al punto deseado (d) y el ángulo entre el punto deseado y la orientación del robot (α). De este modo, el robot podría alcanzar cualquier objetivo de su entorno, previamente definido.

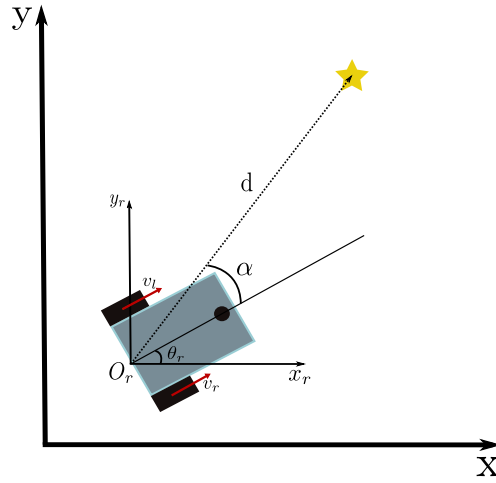


Figura 1: Robot móvil 2D

2. Sistema de Lógica Difusa

Para abordar este problema desde el punto de vista de la lógica difusa se debe seguir un proceso de inferencia. En primer lugar, se definen las variables y términos lingüísticos. En este caso se tiene dos entradas, distancia (d) y ángulo (α); y dos salidas que corresponden a la velocidad lineal de la rueda izquierda (v_l) y derecha (v_r). Por lo tanto, tenemos 4 variables lingüísticas: 2 de entrada y 2 salida. A continuación, en la Tabla 1 se definen los términos lingüísticos de cada variable.

Tabla 1: Variables y Términos lingüísticos

Variables Lingüísticas	Términos Lingüísticos						
Distancia (d)	Very Small (VS)		Small (S)	Medium (M)	Far (F)	Very Far (VF)	
Ángulo (α)	Negative Large (NL)	Negative Medium (NM)	Negative Small (NS)	Zero (Z)	Positive Small (PS)	Positive Medium (PM)	Positive Large (PL)
Vel. izquierda (v_l)	Zero (Z)		Slow (S)	Medium (M)	Fast (F)	Very Fast (VF)	
Vel. derecha (v_r)	Zero (Z)		Slow (S)	Medium (M)	Fast (F)	Very Fast (VF)	

Asimismo, en la Tabla 2 se muestra el rango de cada una de las variables lingüísticas.

Tabla 2: Rango de las variables lingüísticas

Variables Lingüísticas	Distancia (mm)	Ángulo ($^{\circ}$)	Vel. izquierda (v_l)	Vel. derecha (v_r)
Rango	[0, 1000]	[-180, 180]	[0, 60]	[0, 60]

La lógica que seguirá el robot es relativamente sencilla, pues el robot básicamente tenderá a girar hasta que α sea 0, es decir, deberá girar a la izquierda o la derecha dependiendo de si el ángulo es negativo o positivo, respectivamente. Para girar a la izquierda v_l debe ser 0 y v_r debe tomar cualquier otro valor positivo; para girar a la derecha, v_r debe ser 0 y v_l debe tomar cualquier otro valor positivo; y en caso α sea 0, tanto v_l como v_r deben tener un valor positivo. En base a lo descrito, se procede a definir las reglas del sistema difuso. Para ello se utilizan solamente operadores difusos del tipo AND, se establecen los antecedentes y se agregan los consecuentes para todas las reglas. En la Tabla 3 se definen las reglas del sistema difuso y en la figura 2 se muestra el entorno del toolbox Fuzzy Logic Designer, en el cual se implementan las 35 reglas.

Tabla 3: Reglas del sistema Fuzzy

		Ángulo (α)													
		NL		NM		NS		Z		PS		PM		PL	
		v_l	v_r	v_l	v_r	v_l	v_r	v_l	v_r	v_l	v_r	v_l	v_r	v_l	v_r
Distancia (d)	VS	Z	F	Z	M	Z	S	S	S	S	Z	M	Z	F	Z
	S	Z	VF	Z	F	Z	M	S	S	M	Z	F	Z	VF	Z
	M	Z	VF	Z	VF	Z	F	M	M	F	Z	VF	Z	VF	Z
	F	Z	VF	Z	VF	Z	VF	F	F	VF	Z	VF	Z	VF	Z
	VF	Z	VF	Z	VF	Z	VF	VF	VF	VF	Z	VF	Z	VF	Z

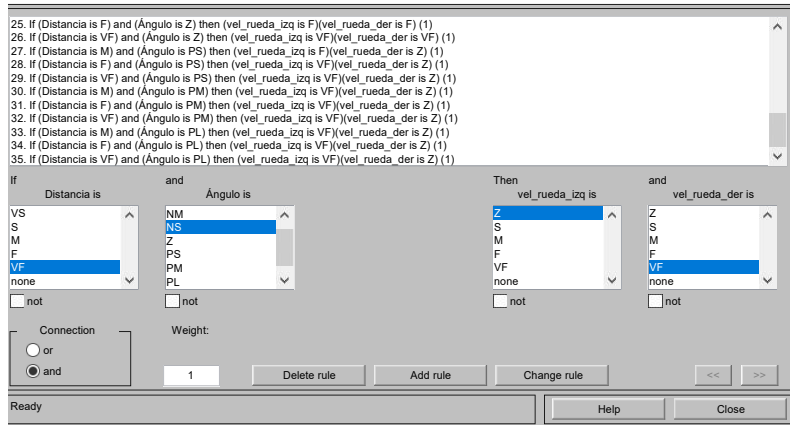


Figura 2: Implementación de las reglas en el entorno del toolbox Fuzzy Logic Designer

Una vez definidas las reglas, se tiene que definir las funciones de pertenencia que se utilizarán para cada término lingüístico. En este proyecto se realizan pruebas con las funciones de pertenencia triangular y trapezoidal para las entradas. En el caso de las salidas se mantendrá una función impulso para cualquiera de los casos, la cual se construirá a partir de una función triangular. El motivo de esto es que se requiere que la velocidad lineal o angular del robot sea constante

(obtenidas a partir de v_r y v_l), dado que tener un rango de valores para un mismo caso podría llevar a resultados erróneos. Finalmente, se utiliza el método de defuzzificación del *centroide*, debido a que es uno de los más utilizados.

3. Resultados

En esta sección se presenta la implementación del sistema de lógica difusa descrito en la sección 2. Asimismo, se muestran los resultados de dicha implementación, para cada una de las funciones de pertenencia a evaluar. A continuación, se detalla los resultados de ello en dos subsecciones.

3.1. Función de pertenencia triangular

En la figura 3 se puede observar los gráficos de pertenencia para cada una de las entradas y salidas. Como se puede observar en las figuras 3a y 3b, se tiene funciones de pertenencia triangulares. Dada la naturaleza del problema, se trató de que estas sean lo más simétricas posibles. Por otro lado, en las figuras 3c y 3d se trató de simular funciones de pertenencia impulso, pero por limitaciones del toolbox se emplearon funciones de pertenencia triangulares con un ancho pequeño. Cabe recalcar que las funciones de pertenencia de las salidas es la misma en ambos casos.

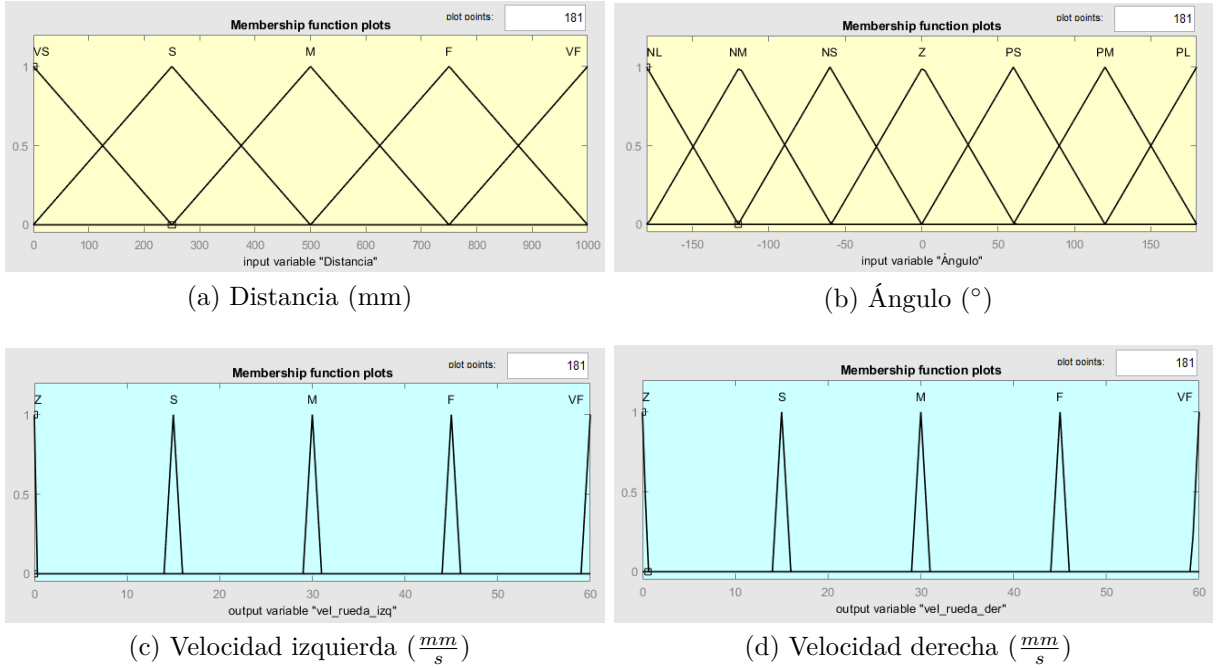


Figura 3: Gráficos de pertenencia

En la figura 4 se pueden observar algunas gráficas relacionadas a las reglas descritas en la sección 2. En las figuras 4a y 4b se muestra los gráficos *surface* que relacionan v_r y v_l con la distancia y el ángulo, respectivamente. Por otro lado, en la figura 4c se puede visualizar el conjunto de reglas, así como los valores de salida que se obtienen para las entradas dadas.

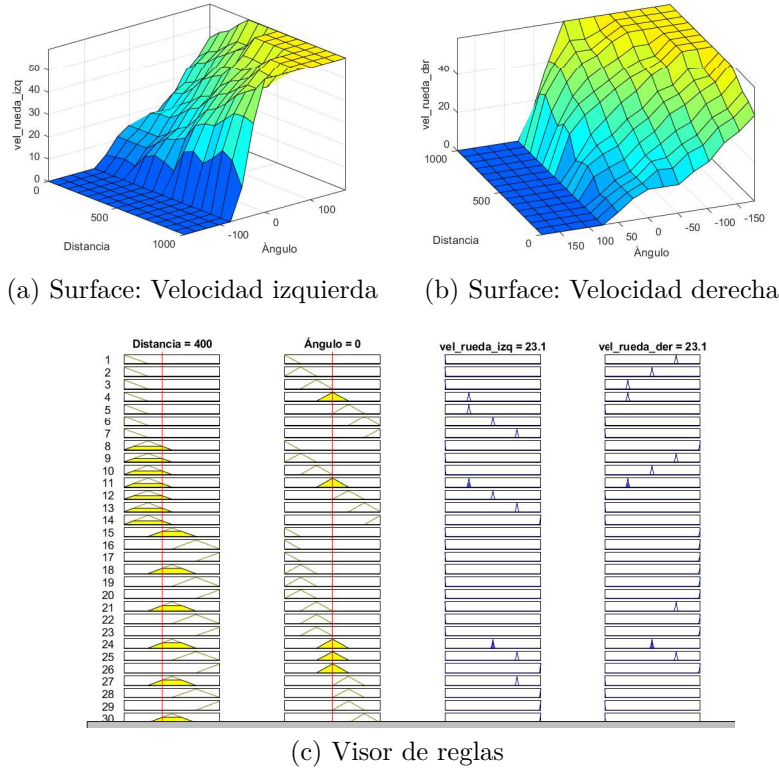


Figura 4: Reglas con función de pertenencia triangular

Finalmente, en la tabla 4 se muestran los resultados que se obtendrían para ciertos casos de distancia y ángulo. En dicha tabla se puede observar que los resultados son consistentes con lo que se espera. Por ejemplo, cuando el ángulo tiene un valor de 0 se espera que ambas ruedas tengan la misma velocidad lineal, de tal modo que se desplacen directamente a su objetivo deseado. Asimismo, cuando el ángulo es positivo v_r es 0 y solo se tiene velocidad en v_l de tal modo que el robot tiende a girar a la izquierda. Por otro lado, cuando el ángulo es negativo sucede lo contrario, v_l es 0 y solo se tiene velocidad en v_r de tal modo que el robot tiende a girar a la derecha. También se tiene un caso particular cuando el ángulo es pequeño (primera fila). En dicho caso se esperaría que v_r sea 0 y en este caso se tiene valores positivos para ambas velocidades. Sin embargo, se tiene que v_l es mayor a v_r , por lo que el robot tendería a girar a la derecha mientras avanza.

Tabla 4: Resultados con función de pertenencia triangular

Distancia (mm)	Ángulo (°)	Vel. izquierda ($\frac{mm}{s}$)	Vel. derecha ($\frac{mm}{s}$)
500	10	33.5	27.2
400	0	23.1	23.1
300	120	47.9	0
200	-150	0	46.1
100	0	15	15

3.2. Función de pertenencia trapezoidal

En la figura 5a y 5b, se puede observar los gráficos de pertenencia trapezoidal para las entradas distancia y ángulo, respectivamente. Como se mencionó anteriormente, las salidas mantienen las funciones de pertenencia impulso debido a la naturaleza del problema.

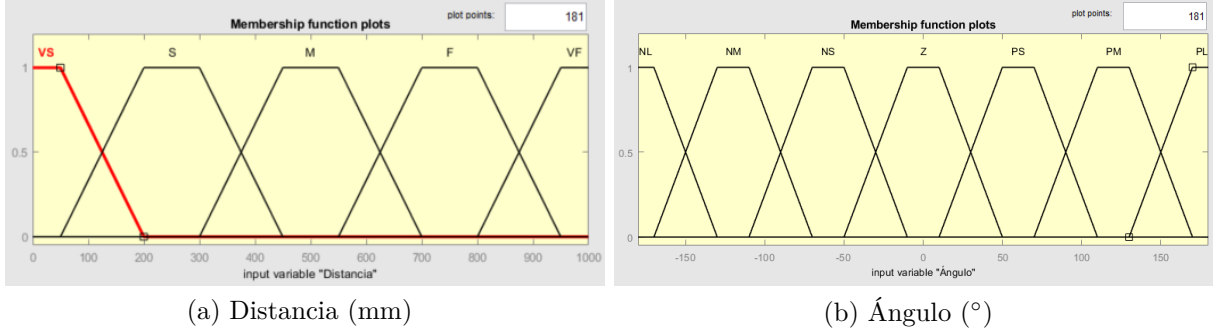


Figura 5: Gráficos de Pertenencia

En la figura 4 se pueden observar las gráficas relacionadas a las reglas descritas en la sección 2. En las figuras 6a y 6b se muestra los gráficos *surface* que relacionan v_r y v_l con la distancia y el ángulo, respectivamente. En este caso, se puede observar una superficie ligeramente más suave que en el caso trapezoidal. Por otro lado, en la figura 6c se puede visualizar el conjunto de reglas, así como los valores de salida que se obtienen para las entradas dadas.

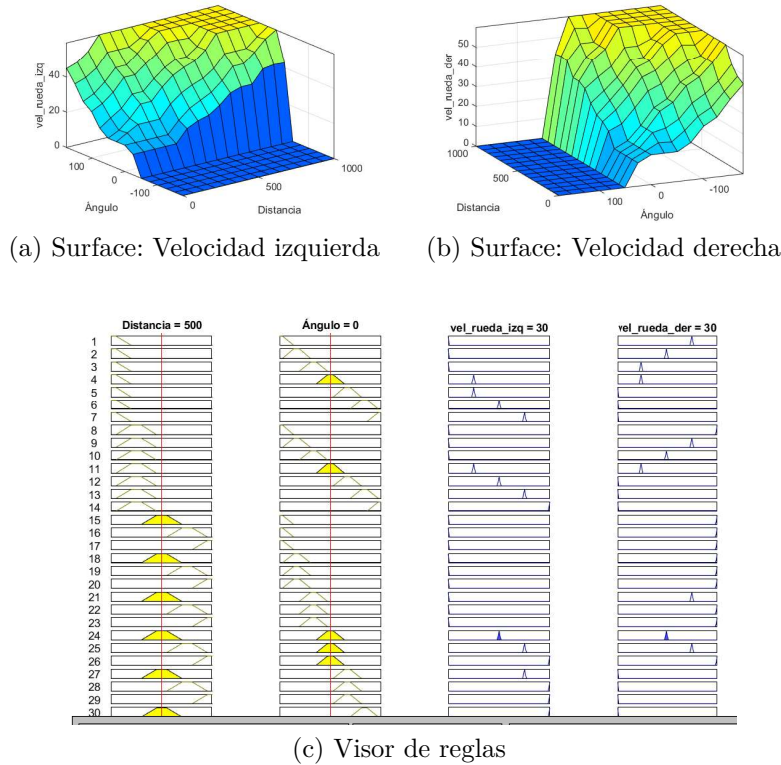


Figura 6: Reglas con función de pertenencia trapezoidal

Finalmente, en la tabla 5 se muestran los resultados que se obtendrían para ciertos casos de distancia y ángulo. Como se puede observar, los resultados son muy similares para los 4 últimos casos, ya que solo hay una pequeña variación en la velocidad. Sin embargo, se tiene un caso no deseado en la primera fila. Cuando la distancia es de 500 mm y un ángulo de 10° , se esperaría que el robot gire a la derecha o que avance mientras gire, pero el resultado nos dice que ambas velocidades lineales son las mismas, por lo que el robot solo avanzará sin girar y, por ende, no llegará al punto deseado.

Tabla 5: Resultados con función de pertenencia trapezoidal

Distancia (mm)	Ángulo ($^\circ$)	Vel. izquierda ($\frac{mm}{s}$)	Vel. derecha ($\frac{mm}{s}$)
500	10	30	30
400	0	23.9	23.9
300	120	45	0
200	-150	0	50.6
100	0	15	15

4. Conclusiones

El presente proyecto presenta el diseño e implementación de un sistema de lógica difusa para la navegación de un robot móvil. Para lograr ello se definió a la distancia y el ángulo hacia el punto deseado como variables lingüísticas de entrada, y a las velocidades lineales de sus ruedas como variables lingüísticas de salida. Asimismo, se definieron los términos lingüísticos correspondientes a cada variable. Paso seguido, se definieron las reglas del sistema utilizando únicamente el operador AND, y definiendo correctamente los antecedentes y consecuentes del sistema.

En la implementación del sistema se trabajó con funciones de pertenencia triangular y trapezoidal. Para cada una de ellas se obtuvo las superficies que relacionan las variables lingüísticas de entrada con las de salida. Asimismo, se obtuvo la visualización de los resultados en base a las reglas. Apartir de ello, se puede concluir que el sistema con funciones de pertenencia triangular obtiene resultados consistentes con lo esperado en una aplicación real. Por otro lado, el sistema con funciones de pertenencia trapezoidal presenta algunos casos en los cuales se obtiene un comportamiento no deseado. Para poder corregir ello se podría hacer que las funciones de pertenencia tengan un menor *overlap* y un menor rango, pues esto funcionó para el caso anterior.

En general, se puede concluir que el sistema de lógica difuso diseñado cumple con el objetivo de brindar al robot móvil las velocidades necesaria para llegar a un punto deseado. Como futuros trabajos se podría aumentar la complejidad del problema agregando obstáculos al entorno del robot. Para ello se tendría que agregar nuevas variables lingüísticas de entrada que representen la lectura de los sensores del robot.