|  |  |
| --- | --- |
|  | |
| **TALLER 5** | |
| **CARRERA:** Mecatrónica | **ASIGNATURA**: Sistemas Embebidos de Control Automático |
| **DOCENTE RESPONSABLE:** César Minaya | |
| **ESTUDIANTE:** | |
| **TÍTULO**: Control de un motor con un controlador difuso | |

**Materiales:**

* Placa Arduino
* Motor DC
* Driver LM298

Diagrama, Esquemático

Descripción generada automáticamente

Figura 1. Esquema de conexión

Introducción

La lógica borrosa (Fuzzy Logic) ha surgido como una herramienta lucrativa para el control de subsistemas y procesos industriales complejos, así como también para la electrónica de entretenimiento y hogar, sistemas de diagnóstico y otros sistemas expertos. Aunque la lógica borrosa se inventó en Estados Unidos el crecimiento rápido de esta tecnología ha comenzado desde Japón y ahora nuevamente ha alcanzado USA y también Europa. La lógica borrosa es todavía un boom en Japón, el número de cartas patentando aplicaciones aumenta exponencialmente. Principalmente se trata de aplicaciones más bien simples de lógica borrosa.

La base de la lógica difusa se basa en la simulación de las opiniones y percepciones de las personas para controlar cualquier sistema. Uno de los métodos para simplificar complejos sistemas es tolerar la imprecisión, la vaguedad y la incertidumbre hasta cierto punto, formando un diagrama de bloques.

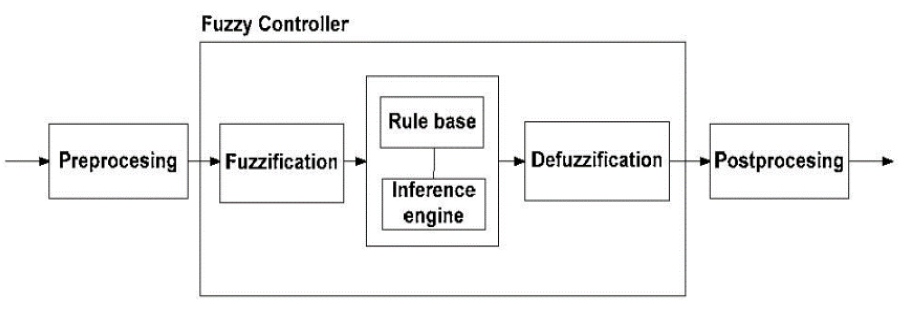


Figura 2. Bloque de proceso por control Fuzzy

El taller se centra en el estudio y aplicación de la teoría de lógica difusa en el área de control de procesos para el motor DC de forma práctica, con el fin de controlar la velocidad mediante un control difuso diseñando el controlador con lógica difusa a través de Fuzzy Toolbox de Matlab, simular e implementar el sistema de control de velocidad.

**Actividades**

**Estudie y analice el comportamiento de la planta a lazo abierto.**

1. Configurar datos del Arduino al Simulink, en la Figura 1 se realiza la medición de la velocidad del motor generando una comunicación serial y actúa como un receptor el Simulink.
2. Enviar datos desde Simulink al Arduino: En la Figura 2 se realiza la colocación del voltaje que va de 0 a 5 [v] considerando que la entrada del Arduino debe ser de 0 a 255 para la ejecución de los PWM.
3. En la Figura 3 se forma un bloque que contenga todo el proceso del motor.

**Diseñe un algoritmo de control basado en lógica difusa. La arquitectura del control es a su elección.**

1. Para el diseño del controlador se utilizó el Fuzzy toolbox de Matlab, a continuación, se describirán los pasos:

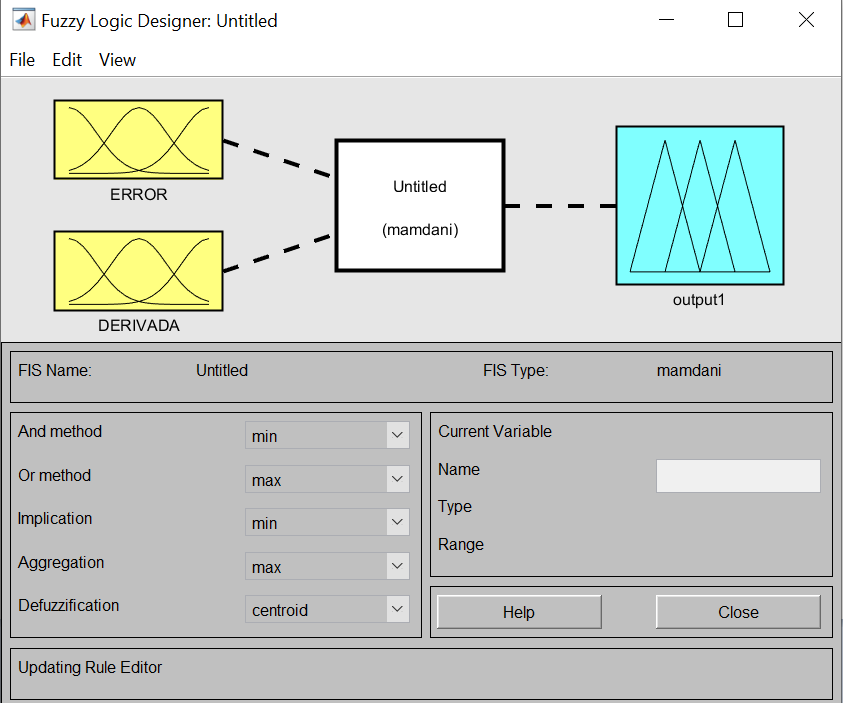


Figura 3. Entradas y salida controlador Fuzzy

Se definieron 2 entradas y 1 salida, donde la primera entrada pertenece al error, la segunda entrada corresponde a la derivada del error y la salida del controlador es el voltaje de entrada del motor, como se observa en la Figura 3.

1. Definir rangos de I/O:
2. Asignar funciones de pertenencia.
3. Asignaciones de variables lingüísticas.

La entrada del error consta de 7 funciones de pertenencia triangulares , distribuidos en el rango de [-1 ; 1] ,como se observa en la Figura 4.

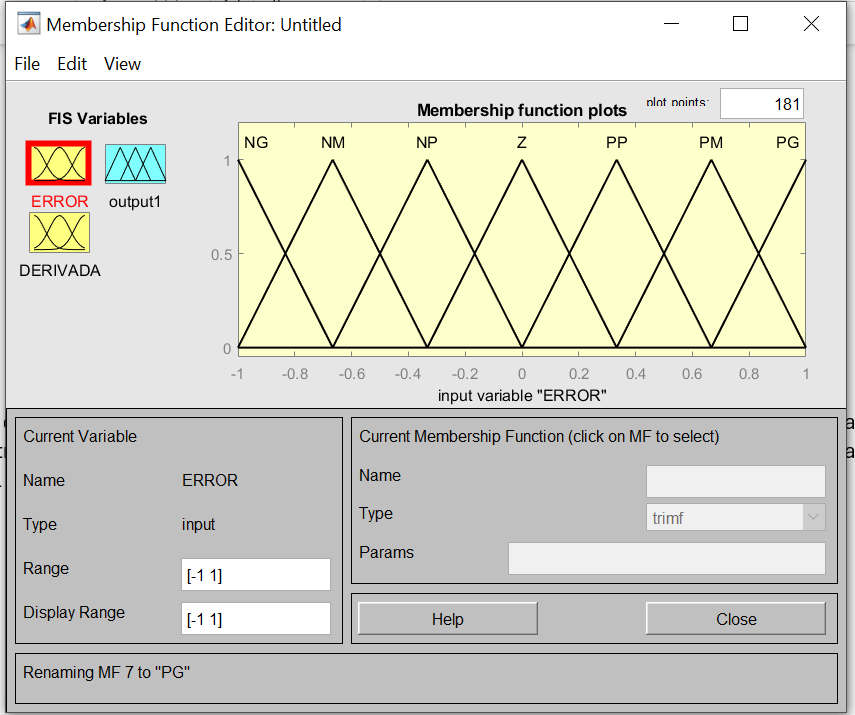


Figura 4. Conjuntos de Entrada (velocidad)

La entrada derivada Error consta de 7 funciones de pertenencia distribuidos en el rango de [-1,1], en el sistema el error es calculado, como se observa en la Figura 5.

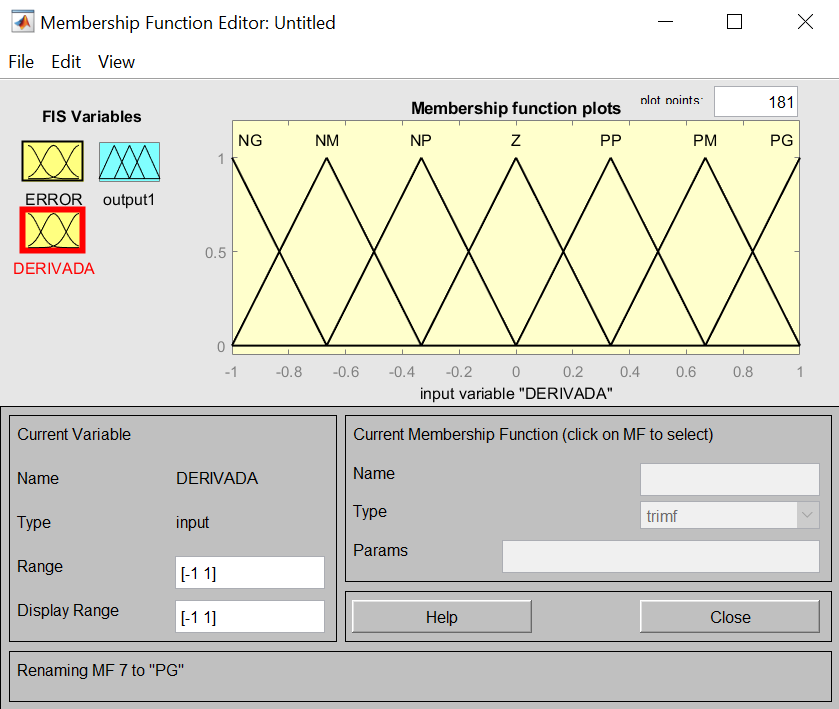
****

Figura 5. Conjuntos de Entrada (error)

En la Figura 6, la salida del controlador consta de 9 funciones de pertenencia, este está en el rango de [-1,1].

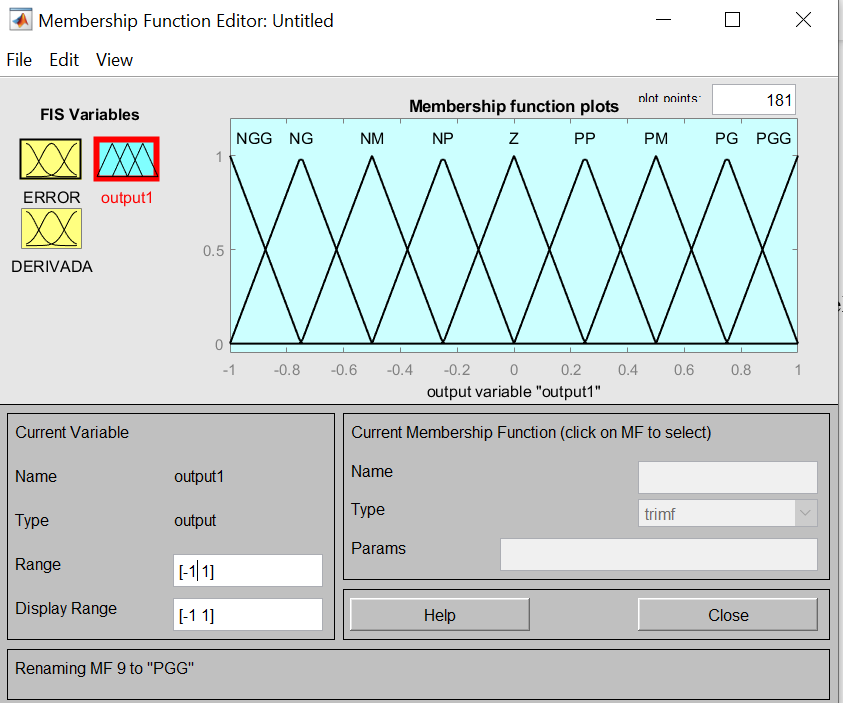


Figura 6. Conjuntos de Salida(Voltaje)

1. Reglas de pertenencia:

En el diseño del controlador se consideraron 49 reglas, estos fueron determinados:

Luego de realizar la matriz de reglas, procedemos a agregarlas en el Rule Editor, las reglas que se muestran en la Figura 7.

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| e/de | NG | NM | NP | Z | PP | PM | PG |
| NG | NGG | NGG | NGG | NG | NM | NP | Z |
| NM | NGG | NGG | NG | NM | NP | Z | PP |
| NP | NGG | NG | NM | NP | Z | PP | PM |
| Z | NG | NM | NP | Z | PP | PM | PG |
| PP | NM | NP | Z | PP | PM | PG | PGG |
| PM | NP | Z | PP | PM | PG | PGG | PGG |
| PG | Z | PP | PM | PG | PGG | PGG | PGG |

Figura 7. Matriz de control

Tabla 1. Parámetros de entrada/salidas variables lingüísticas

|  |  |  |
| --- | --- | --- |
| NG = error negativo grande  NM = error negativo mediano  NP = error negativo pequeño  Z = no hay error  PP = error positivo pequeño  PM = error positivo mediano  PG = error positivo grande | NG = error negativo grande  NM = error negativo mediano  NP = error negativo pequeño  Z = no hay error  PP = error positivo pequeño  PM = error positivo mediano  PG = error positivo grande | NGG = error negativo muy grande  NG = error negativo grande  NM = error negativo mediano  NP = error negativo pequeño  Z = no hay error  PP = error positivo pequeño  PM = error positivo mediano  PG = error positivo grande  PGG = error positivo muy grande |

Luego como se muestra en la Figura 9, se determinar los parámetros del controlador.