

Motor de CD controlado por medio de un sistema difuso

M. Orozco García, *Estudiante de Ingeniería Biónica, UPIITA*

Resumen—Los sistemas difusos tienen gran importancia en sistemas cuyas ecuaciones sean muy complejas o cuyo cómputo consume muchos recursos. Dado que los problemas de ingeniería biónica en su mayoría tienen estas características es importante su conocimiento. Por ello el presente trabajo contempla un ejercicio en el cual se pone en práctica los conceptos básicos del control difuso. El dispositivo en general consiste en un motor de DC, un encoder, un Arduino Mega 5906 y un puente H; estos componentes corresponden al actuador, sensor, controlador y etapa de potencia respectivamente. El sistema de inferencia difuso toma como entrada el error de posición y la velocidad deseada, lo cual genera un PWM que se modifica constantemente para obtener los parámetros de velocidad y posición deseados hasta alcanzarlos.

Índice de términos—Arduino, control difuso, defusificación encoder, fusificación, motor CD, reglas de inferencia.

I. INTRODUCCIÓN

“EN los últimos años han surgido varias técnicas inteligentes, tales como la Lógica Borrosa, Redes Neuronales, Algoritmos Genéticos y Razonamiento Probabilístico, las cuales se han agrupado bajo el nombre de *Computación Suave* o *Soft Computing*. Este nombre es usado para diferenciarla de la computación *dura*, *hard* o *tradicional*.” [1]

Las ventajas que ofrece el control difuso sobre el tradicional es que sin tener necesariamente que analizar todos los casos el sistema puede ofrecer respuestas siguiendo las reglas de inferencia creadas solo de algunos casos. Esto es útil particularmente en sistemas muy complejos: donde existan muchas variables de entrada y salida, donde el modelo sea de difícil obtención o bien donde el modelo consuma muchos recursos computacionalmente hablando.

Los motores son un campo ampliamente estudiado en cuanto al control de los mismos por lo cual existen varios métodos para controlarlos dependiendo del tipo de motor y de sus aplicaciones. En el presente trabajo utilizaremos un control de CD para controlar tanto posición como velocidad. Esto se logrará con ayuda de un Arduino Mega 5906, como controlador y un encoder y potenciómetro como sensores.

II. SOBRE LOS SENSORES

El sistema utiliza dos sensores uno para la posición y otro para la velocidad. Estos son un encoder incremental y un potenciómetro.

A. Encoder incremental.

Está incorporado a la parte posterior del motor, consta de dos optocopladores. Estos leen la presencia o ausencia de una hélice de 12 divisiones sujeta al eje del motor. Esta hélice posee una relación de 1:65 con respecto al movimiento frontal. En este sistema solo se utilizará un optocoplador ya que la direccionalidad del motor podrá ser obtenida gracias al potenciómetro y al programa.

La señal de salida de este sensor está conectada a un pin de interrupción del Arduino cuya subrutina suma una unidad a un contador cada vez que sensa. Este contador será posteriormente leído así como el tiempo transcurrido, dichas cantidades restadas por las leídas en el ciclo anterior y divididas entre sí, dan una medida de velocidad.

Con esto se obtiene la velocidad en divisiones por microsegundo que se convierten a revoluciones por minuto posteriormente para la medición de velocidad.

B. Potenciómetro

Está sujeto a la parte frontal del motor este puede medir alrededor de 220 grados. Sin embargo se delimitó en el programa su uso a 180 grados (de -90 a 90). Y se dio un margen de seguridad de 10 grados pasando los cuales el programa apaga el motor.

La señal que se obtiene gracias al potenciómetro se conecta a un pin de lectura analógica que contiene un ADC de 10 bits y obtiene la señal una lectura de 0 a 1023.

Se realizó un ajuste polinomial de segundo orden basado en varias muestras tomadas a diferentes ángulos para convertir el voltaje leído a grados de -90 a 90.

III. SOBRE EL CONTROL DIFUSO

El controlador difuso se realizó con dos variables de entrada y una de salida. Las variables de entrada son errores obtenidos restando el valor deseado menos el medido tanto de posición como de velocidad. La variable de salida será un cambio en la proporción del PWM.

El sistema de manera general quedará como se muestra en la Fig 1.

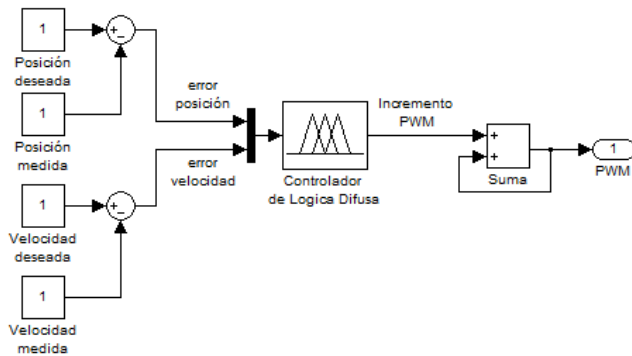


Fig. 1. Diagrama de bloques de control.

A continuación se mostrarán las etapas del control difuso de nuestro sistema.

A. Fusificación de variables de entrada.

La variable de entrada correspondiente al error de posición puede tener valores de 0 a 180 grados. Cabe mencionar que solo trabajamos con valores absolutos de error ya que el signo de esta variable solo lo utilizaremos para la dirección del motor, sin embargo el control es el mismo sin importar si es un giro horario o anti-horario. Sus conjuntos de pertenencia se muestran en la Fig. 2.

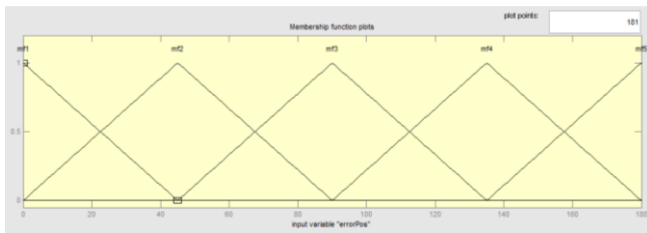


Fig. 2. Conjuntos de pertenencia del error de posición.

Para el error de velocidad se sigue una lógica similar, sin embargo el signo si es requerido para saber si hay que frenar o acelerar el motor. El motor se controla de 90 a 170 RPM; esto se debe a que el motor empieza a moverse con un PWM que genera 84 RPM y se deja una banda de confort hasta 90 RPM y, 170 RPM es el máximo medido con la fuente de 5V utilizada (las especificaciones del motor sugieren usar una fuente de 12V para generar 400 RPM). Por lo tanto el error de posición será de -80 RPM hasta 170 RPM.

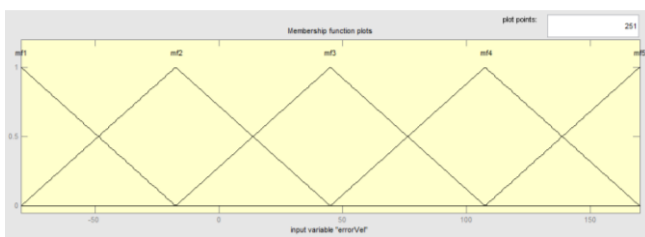


Fig. 3. Conjuntos de pertenencia del error de velocidad.

B. Reglas de inferencia.

Siguen un sistema tipo Mamdani que utiliza reglas de inferencia "and". Su superficie de control se muestra en la Fig. 4.

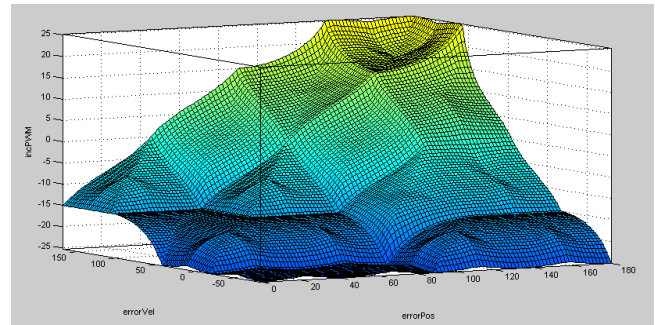


Fig. 4. Superficie de control.

C. Variables de salida.

La única variable de salida será un incremento en el PWM. Este sumará de -30 a 30 el valor previo de PWM para controlar tanto posición como velocidad. Para obtener este valor se utilizó el método del centroide. Si el valor excede el rango de PWM aceptado por Arduino [0,255] se ajustará para que el valor se mantenga dentro de los límites.

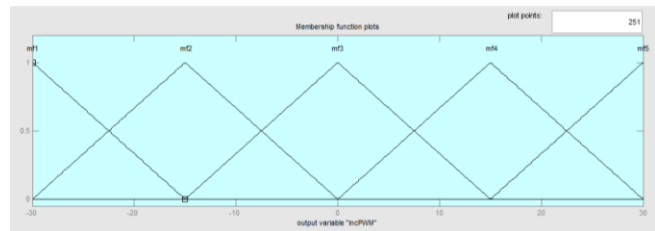


Fig. 5. Conjuntos de pertenencia del incremento de PWM.

IV. SOBRE LA ESTRUCTURA DEL PROGRAMA

El programa tiene una condicional principal que se controla a través de un botón físico. Si el botón es presionado el programa detiene el motor y espera a leer los valores deseados de posición y velocidad por el puerto serial. Una vez hecho esto la otra rama de la condicional se activa hasta que se vuelve a presionar el botón.

La segunda rama contiene el programa dedicado al control. En esta se sigue el proceso mostrado en la Fig. 5.



Fig. 6. Flujo de trabajo de la segunda rama del programa.

V. RESULTADOS

El sistema completo quedó como se muestra a continuación y muestra valores aceptables considerando que los acoples y el control no es tan preciso.

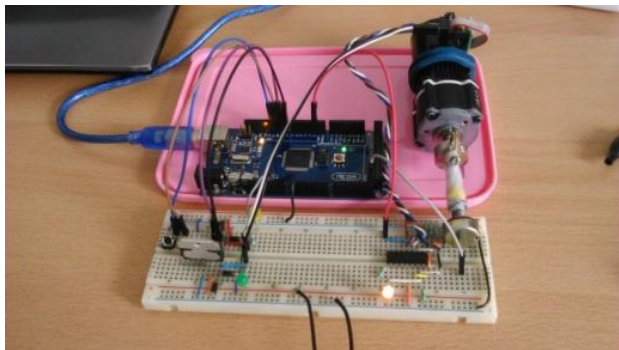


Fig. 7. Conjuntos de pertenencia del error de velocidad.

Para fines futuros se podría anexar un mejor acople y un mejor marcador o flecha para los ángulos (incluso se podría pensar en una escala graduada), así como un tacómetro que mida mejor la velocidad.

VI. REFERENCIAS

- [1] R. A. Maguiña Pérez, "Sistemas de inferencia basados en Lógica Borrosa: Fundamentos y caso de estudio." Vol: 7, No: 1, Revista de Ingeniería de Sistemas e Informática, Enero - Julio 2010. Pp. 91-104.
- [2] H. E. Espitia Cuchango y J. J. Soriano Méndez, "Sistema de inferencia difusa basado en relaciones Booleanas." Vol. 15, No. 2, Ingeniería, 2010 Pp. 52-66.