

Control de posición del robot LEGO® MINDSTORMS® EV3 usando una red neuronal adaptativa con sistema de inferencia difuso

RESUMEN: En este artículo, se presenta el control de posición de un robot móvil LEGO MINDSTORMS EV3. Para ello, se utilizan técnicas de control inteligente como son las redes ANFIS. La programación del robot se hace a través del software de MATLAB llamado SIMULINK. Se explica a groso modo en que consiste el algoritmo de las redes ANFIS. Se genera y se hace el entrenamiento de la red ANFIS por medio de una caja de herramientas de lógica difusa incluida en MATLAB. Una vez creada la red ANFIS, se empuja en un bloque que puede ser usado en SIMULINK. Por medio de la técnica de control por aprendizaje inverso se crea en SIMULINK el sistema que vía Wi-Fi controla el robot EV3. Para el análisis de resultados, por medio de SIMULINK se crea un controlador PID, mismo que se autoajusta optimizando sus parámetros. Se comparan los resultados dados por el controlador PID, contra los resultados obtenidos por el controlador ANFIS propuesto en esta aportación. En lo referente a mayor rapidez en la respuesta y mayor precisión y exactitud para llegar a un punto de referencia dado, resultó mejor el controlador ANFIS propuesto que el controlador PID.

Palabras clave: red ANFIS, sistemas de inferencia difusos, robot LEGO, aprendizaje inverso, SIMULINK, controlador.



Colaboración

Julio Ramírez; Daniel Tellechea; Gildardo Jiménez, Instituto Tecnológico de Nogales

ABSTRACT: This paper presents a robot LEGO MINDSTORMS EV3 position control. To achieve this, intelligent control techniques are used such as the ANFIS networks. The robot programming is done through software called SIMULINK from MATLAB. It is explained in roughly how ANFIS algorithm works. It builds and carries out the ANFIS network training using a fuzzy logic toolbox that belongs to MATLAB. Once created the ANFIS network, it is embedded in a block that can be used in SIMULINK. The system is created in Simulink for robot control via Wi-Fi. In order to achieve this a inverse learning technique is used. For results analysis a PID controller is created in SIMULINK. This controller auto-adjusts itself to optimize its parameters. The results given by the PID controller, with the results obtained by the proposed ANFIS controller are compared. With regard to faster response and greater precision and accuracy to reach a given reference point, the proposed ANFIS controller performed better than PID controller.

Keywords: ANFIS network, fuzzy inference systems, LEGO robot, inverse learning, SIMULINK, controller.

INTRODUCCIÓN

Uno de los enfoques más nuevos en la teoría de control en lazo cerrado es el control Neurodifuso. En el área de la Inteligencia Computacional [14] las redes Neurodifusas ANFIS son un paradigma híbrido que resulta de la combinación de redes neuronales con lógica difusa.

Lo que se desea controlar con las redes ANFIS (Adaptive Network-based in Fuzzy Inference Systems) es la posición del robot LEGO® MINDSTORMS® EV3. Este es un robot de relativamente bajo costo que está teniendo muy buena aceptación en la enseñanza de la robótica [4,16,3]. Otra herramienta que se usa en este trabajo es el software de programación gráfica llamado SIMULINK®. Este es un software de la compañía MATLAB®, muy usado en la comunidad académica de las Ingenierías. SIMULINK en su versión 2014 incluye librerías que facilitan el uso del robot EV3.

Uno de los problemas que se tiene para incentivar la ciencia y la ingeniería a nivel bachillerato y licenciatura, es lo abstracto que resulta la enseñanza de estas disciplinas a nivel teórico [13]. En ese sentido, el objetivo de este trabajo es presentar de una manera muy atrayente para el alumno, el uso de técnicas innovadoras del control inteligente. La ventaja de este enfoque es que no representa una gran dificultad para el alumno que puede hacer uso de los comandos de MATLAB, que cuenta con una caja de herramientas de lógica difusa que incluye las ANFIS.

Existen trabajos relacionados del uso de las ANFIS pero son con robots de laboratorio más costosos [12]. También existen muchas contribuciones que incluyen robots LEGO en sus aportaciones [13,6], pero no usan SIMULINK. Existen contribuciones que incluyen el uso de robots LEGO manipulados a través de SIMULINK [1], pero no hacen uso del controlador tipo ANFIS. En ese sentido, el actual trabajo es de los primeros en controlar la posición de un robot LEGO MINDSTORMS EV3 usando herramientas de control inteligente como las ANFIS que se pueden incluir en SIMULINK.

El control de posición en los robots móviles de este tipo es importante. El caso de un robot usado para desactivar explosivos [11] o para la exploración de volcanes [2] debe tener una exactitud de avance y retroceso milimétrica. Además de las aplicaciones anteriores en el control de posición de robots móviles, existen muchas más que se mencionan en la literatura [15].

MATERIAL Y MÉTODOS

La robotización, es decir, los sistemas automatizados, se encuentran en casi todos los sectores de la sociedad. Los podemos encontrar en la industria, en la medicina y en el sector militar. Por tal motivo, la robótica móvil y los sistemas autónomos son tópicos de creciente interés en los programas educativos de Ingeniería y Ciencias Computacionales.

LEGO MINDSTORMS EV3

Bloque de LEGO MINDSTORMS EV3, es el robot de tercera generación en la línea robótica LEGO MINDSTORMS. Es el robot sucesor de segunda generación LEGO MINDSTORMS NXT 2.0. La designación "EV" se refiere a la "evolución" de la serie NXT. "3" se refiere al hecho de que es la tercera generación de bloques de LEGO - primero fue RCX y el segundo es NXT.

En Estados Unidos LEGO ha liderado en lo referente al programa gubernamental STEM [8], inspirando a los usuarios a participar en la diversión y el aprendizaje práctico. El corazón de LEGO MINDSTORMS es el bloque EV3. Es el bloque inteligente programable que controla los motores y sensores. También proporciona comunicación inalámbrica. Existe un software propiedad de LEGO de naturaleza gráfica para programar a los robots. Los autores prefirieron usar SIMULINK por ser más común entre los estudiantes de ingeniería. En la figura 1, se muestra el robot EV3 con algunas de sus herramientas.

El actuador principal del EV3 es el llamado motor grande [9]. Es un motor inteligente que tiene incorporado un sensor con resolución de un grado para un control preciso en la marcha. La salida principal del sistema es un sensor ultrasónico. El sensor ultrasónico es un sensor digital que puede medir la distancia a un objeto delante de él. Lo hace mediante el envío de ondas sonoras de alta frecuencia que miden el tiempo que tarda el sonido que se refleja de regreso al sensor.

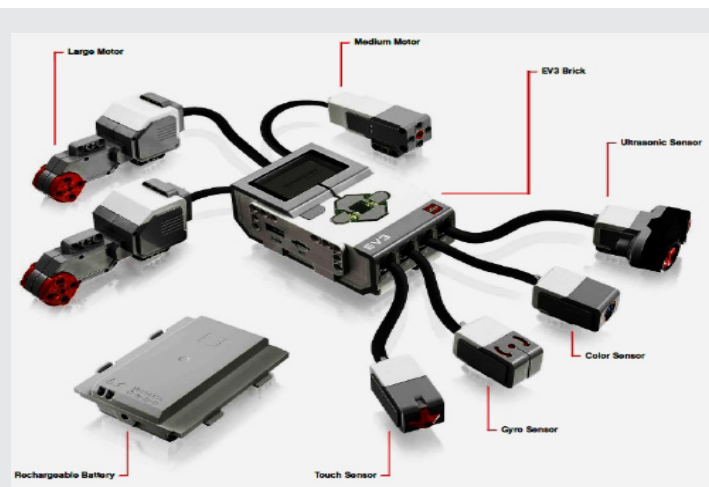


Figura 1. Herramientas del LEGO MINDSTORMS EV3.

El rango de detección de este sensor está entre 3 y 250 centímetros (con una precisión de +/- 1 cm).

SIMULINK Y LEGO MINDSTORMS EV3

SIMULINK es un entorno de programación visual, que funciona sobre el entorno de programación MATLAB. Es posible utilizar SIMULINK para programar y ejecutar algoritmos sobre el robot LEGO MINDSTORMS EV3. El EV3 incluye un microcontrolador basado en ARM9, varios sensores, además de los descritos anteriormente (por ejemplo, de rayos infrarrojos, color, tacto, y otros), y tres servomotores. Con SIMULINK se puede programar el robot para que realice tareas: como seguir una línea o detectar y evadir objetos.

Con SIMULINK también se puede desarrollar algoritmos para sistemas de control y aplicaciones de robótica [5], y luego simular para verificar que los algoritmos funcionan según lo previsto. Con el clic de un botón, es posible descargar y ejecutar los algoritmos directamente sobre LEGO MINDSTORMS EV3 (vía Wi-Fi o una conexión Ethernet). El soporte integrado de SIMULINK para la plataforma LEGO MINDSTORMS EV3 incluye:

- 1.- Instalación y configuración automatizada.
- 2.- Biblioteca de bloques de SIMULINK que conectan con las entradas y salidas de LEGO MINDSTORMS EV3, como botones y altavoces; numerosos sensores de entrada incluyendo el color, el tacto, giroscopio, ultrasonidos, infrarrojos, y el codificador; y sensores de salida tales como servo motor, luz de estado, y la pantalla.

3.- Parámetro de ajuste interactivo de los productos y la monitorización de señales de entradas de sensor para las aplicaciones que se ejecutan en el robot EV3.

SISTEMAS DE INFERENCIA DIFUSOS

Los sistemas de inferencia difusos se basan en la lógica difusa. Uno de los objetivos de la lógica difusa es proporcionar las bases del razonamiento aproximado que utiliza premisas imprecisas como instrumento para formular el conocimiento. Los sistemas de inferencia difusos, también son llamados por su acrónimo en inglés FIS (*Fuzzy Inference Systems*). Los FIS se basan en la configuración de tres componentes conceptuales: una base de reglas, la cual contiene una selección de las reglas difusas; una base de datos, la cual define las funciones de pertenencia usadas en las reglas difusas y un mecanismo de razonamiento o motor de inferencia, el cual ejecuta el procedimiento de inferencia.

Existen principalmente tres tipos fundamentales de sistemas de inferencia difuso. El tipo 1 o modelo de *Mamdani*; el tipo 2 o modelo de *Tsukamoto* y el tipo 3 o modelo de *Takagi-Sugeno*. En la figura 2 se esquematiza los tres tipos de mecanismos de inferencia difusos. Una regla de inferencia difusa tiene la forma siguiente:

Si x es A_1 Y y es B_1 Entonces $f_1 = p_1x + q_1y + r_1$

Donde a la premisa del SI condicional, se le llama "antecedente" y a la premisa del resultado Entonces, se le llama "consecuente".

ANFIS

El modelo ANFIS fue desarrollado por J.R. Jang en 1993 [10]. Se basa en las redes neuronales adaptativas y en los sistemas de inferencia difusos. Las capacidades adaptativas de las redes ANFIS las hacen directamente aplicables a una gran cantidad de áreas como control adaptativo, procesamiento y filtrado de señales y series de tiempo, clasificación de datos y extracción de características a partir de ejemplo.

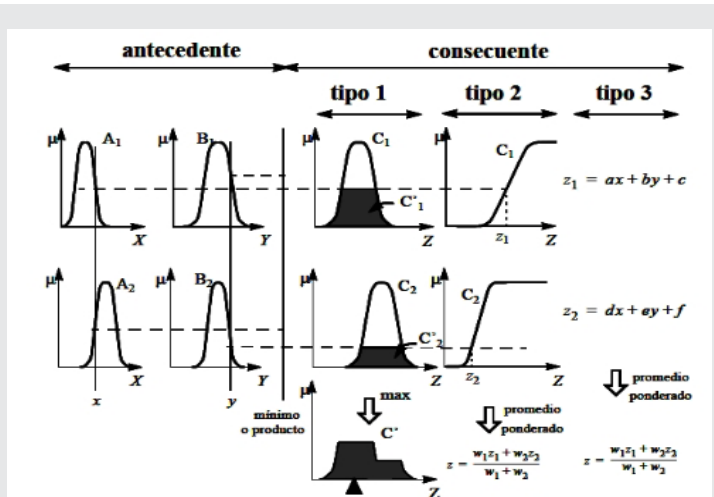


Figura 2. Mecanismo de razonamiento difuso.

La red ANFIS, como se muestra en la figura 3, es una estructura de varias capas con nodos y enlaces direccionales a través de los cuales dichos nodos son conectados. Esta red adaptativa está fundamentada en principios de aprendizaje híbrido, es decir, que usa diferentes métodos de aprendizaje básico para el entrenamiento de la red.

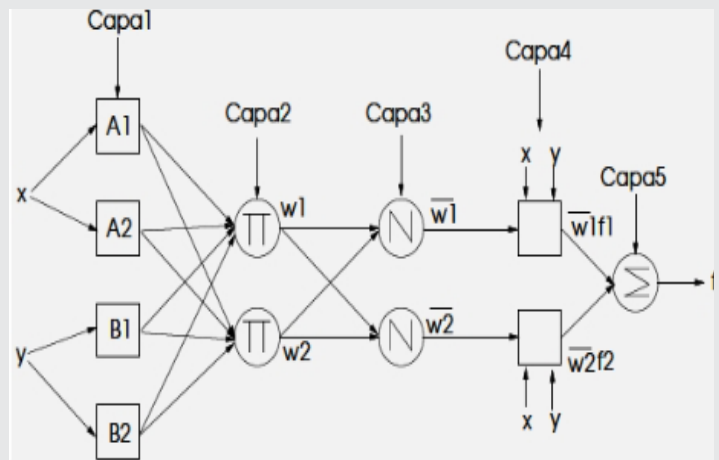


Figura 3. Arquitectura típica para ANFIS con reglas tipo Takagi-Sugeno.

Se propone un sistema de inferencia difuso con dos entradas "x" y "y" junto con una salida "z", además, se establecen dos reglas difusas de tipo TS (*Takagi-Sugeno*) como se muestra a continuación.

Regla 1: Si x es A_1 Y y es B_1
Entonces $f_1 = p_1x + q_1y + r_1$

Regla 2: Si x es A_2 Y y es B_2
Entonces $f_2 = p_2x + q_2y + r_2$

Entonces conforme a la estructura de la red la información se propaga capa por capa.

Capa 1: En ésta capa se difusifica cada entrada (x, y), donde usualmente es utilizada la función de membresía campana generalizada Ec. (1).

$$O_i^1 = \mu_{A_i}(x) = \frac{1}{1 + \left| \frac{x - c_i}{a_i} \right|^{2b}} \quad \text{Ec. (1)}$$

Capa 2: Para esta segunda capa, cada nodo calcula la fuerza de ponderación de cada regla Ec. (2).

$$O_i^2 = w_i = \mu_{A_i}(x) \cdot \mu_{B_i}(y) \quad \text{Ec. (2)}$$

Capa 3: Los nodos en esta capa indican la normalización N de los niveles de ponderación de cada regla Ec. (3).

$$O_i^3 = \bar{w}_i = \frac{w_i}{\sum w_i} \quad \text{Ec. (3)}$$

Capa 4: Para la cuarta capa se calcula la salida Takagi-Sugeno de cada regla Ec. (4).

$$O_i^4 = \bar{w}_i f_i = \bar{w}_i (p_i x + q_i y + r_i) \quad \text{Ec. (4)}$$

Capa 5: Con éste último nodo se calcula la salida total del sistema Ec. (5).

$$O_i^5 = f(x) = \sum \bar{w}_i f_i = \frac{\sum w_i f_i}{\sum w_i} \quad \text{Ec. (5)}$$

Las redes ANFIS son usadas para el aprendizaje de funciones no lineales de control en modo supervisado. Esta propiedad de aprender este tipo de funciones, es lo que las hace atractivas para su uso en el control de sistemas.

MÉTODO POR APRENDIZAJE INVERSO

El método por aprendizaje inverso se basa en ecuaciones en diferencias, por lo que la técnica de control usada aquí es la del control digital. El desarrollo del aprendizaje inverso, envuelve dos fases en el diseño de controladores Neurodifusos [7]. Una es la fase de aprendizaje y la otra la fase de aplicación. En el enfoque actual, el aprendizaje de la dinámica inversa del modelo de planta se hace fuera de línea.

Asumiendo que el orden de la planta (i.e., el número de variables de estado) se conoce y todas las variables de estado se pueden medir, se tiene la Ec. (6).

$$x(k + 1) = f(x(k), u(k)) \quad \text{Ec. (6)}$$

A partir de la Ec. (6) podemos generalizar y llegar a la Ec. (7).

$$x(k + n) = F(x(k), U(k)) \quad \text{Ec. (7)}$$

En la Ec. (7), n es el orden de la planta, F es una función compuesta múltiple de f, y U es la acción de control desde k hasta k+n-1, lo cual es igual a $[u(k), u(k+1), \dots, u(k+n-1)]^T$. La ecuación anterior apunta al hecho de que dada una entrada de control u desde el paso de tiempo k hasta k+n-1, el estado de la planta se moverá desde x(k) hasta x(k+n) exactamente en n pasos de tiempo.

La red ANFIS debe de aprender el comportamiento de la planta, o sea, el comportamiento del robot EV3. Para esto, se entrena la red con una función que pueda mapear el espacio de movimiento del robot. Por ejemplo, si se desea que el robot a partir de una posición cero obedezca comandos de avanzar y luego retroceder para de nuevo avanzar, entonces para el entrenamiento se debe usar una función senoidal.

Dado que el aprendizaje es fuera de línea es necesario contar con el modelo matemático de la planta, para poder hacer el entrenamiento en un simulador. Asumiendo que se cuenta con el modelo matemático del robot, entonces en SIMULINK (simulador a usar) se aplica una señal de entrada. En el caso particular de estudio, esta señal puede ser una función senoidal. Esta señal será la entrada de control $u(k)$, que produce las salidas $x(k)$ y $x(k+1)$ según se muestra en la figura 4.

Figura 4. Diagrama a bloques para el método de aprendizaje inverso: (a) bloque de planta; (b) fase de entrenamiento; (c) fase de aplicación.

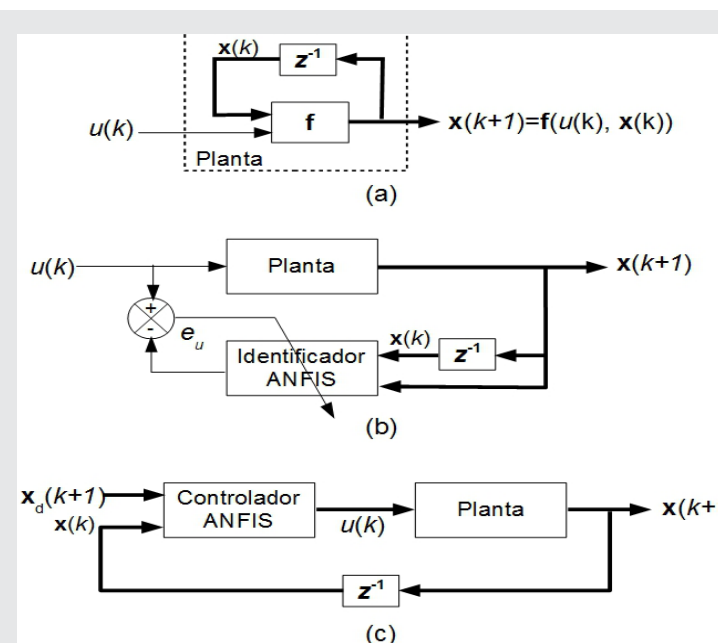


Figura 4. Diagrama a bloques para el método de aprendizaje inverso: (a) bloque de planta; (b) fase de entrenamiento; (c) fase de aplicación.

A partir de las señales se forman los vectores de entrenamiento que se organizan en la matriz mostrada en la Ec. (8).

$$[x(k)^T, x(k + 1)^T; U^T] \quad \text{Ec. (8)}$$

En la fase de entrenamiento (ver figura 4(b)) se aplica a la entrada el vector columna UT (función senoidal discreta). Después de pasar por la planta (modelo matemático del robot) se producen las señales discretas que conforman los vectores columna x(k) y x(k+1) de la Ec. (8). El bloque z-1 es usado para representar el operador correspondiente al retardo de una unidad de tiempo. A partir de la señal de salida x(k+1) y del bloque z-1 se obtiene la señal x(k). Estas dos señales al pasar por el identificador ANFIS regeneran la señal de entrada u(k). Cuando el error eu tiende a cero, entonces el identificador ANFIS aprendió el comportamiento dinámico de la planta; a esto se le llama aprendizaje inverso.

Al decir que el identificador ANFIS aprendió el comportamiento dinámico de la planta, se quiere dar a entender que la red ANFIS se debe comportar en la misma forma que se comportaría la planta ante cualquier señal de entrada. En la fase de aplicación Figura 4 (c), a la señal de control se la define como xd(k+1). A esta señal se la llama señal deseada. Al hecho de llamarle de esta forma, da a entender el comportamiento predictivo del sistema de control.

Si nos fijamos, xd(k+1) se refiere a una señal con una unidad de tiempo en adelante o señal futura (virtual). El identificador ANFIS que en esta fase funge como controlador ANFIS, que fue diseñado para recibir una señal en adelante xd(k+1) y una señal actual x(k), ajustará su salida u(k) tal que x(k+1) siga a xd(k+1). De esta manera es que se logra el diseño del controlador ANFIS.

MODELO MATEMÁTICO DE LA PLANTA

Con el fin de poder simular el entrenamiento de la red ANFIS es necesario contar con el modelo matemático del comportamiento dinámico del robot EV3. Para ello se usó un método empírico de ingeniería inversa. El procedimiento para obtener el modelo matemático primero implica: armar el robot EV3 con los motores y el sensor ultrasónico de proximidad; como se muestra en la figura 5.

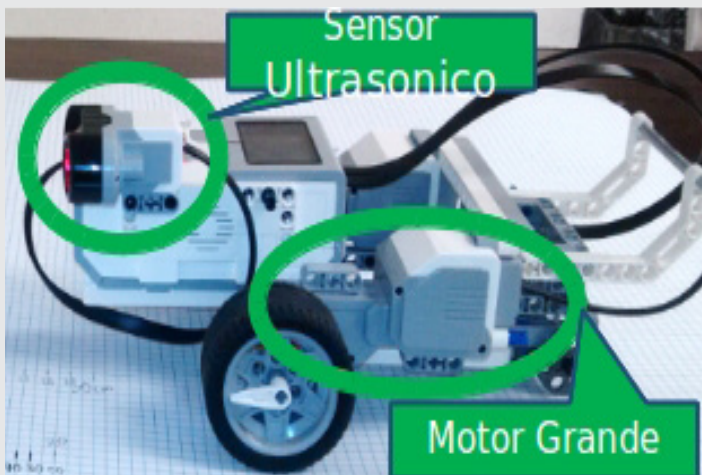


Figura 5. Robot EV3 con motores y sensor.

Después de ello, es aplicada una señal oscilante tipo senoidal con una amplitud de 88 cm a diferentes frecuencias. Las lecturas son registradas en la Tabla 1.

Tabla 1. Lecturas del comportamiento dinámico del robot EV3.

Recorrido	Distancia(m)
Troqueladora 07 – Torno 13	9.08
Troqueladora 07 – Torno 24	14.73
Troqueladora 07 – Torno 25	5.50
Torno 13 – Troqueladora 12	9.70
Torno 24 – Troqueladora 12	17.70
Torno 25 – Troqueladora 12	6.40
TOTAL	63.11

En la Tabla 1, la velocidad es una estimación de velocidad en la escala de 0 a 100 % que por defecto incluye el software de programación del EV3. La frecuencia angular ω es la medición de esa velocidad en rad/seg. La ganancia $|G(j\omega)|$ es la razón entre la distancia medida y la distancia programada, expresada en decibeles (dB). Esto es, la distancia máxima recorrida D_m que se midió con el sensor, entre la distancia máxima a recorrer programada D_p . La ganancia $|G(j\omega)|$ se obtiene con la Ec. (9).

$$|G(j\omega)|_{dB} = 20 \log_{10} \left| \frac{D_m}{D_p} \right| \quad \text{Ec. (9)}$$

De la Tabla 1 podemos observar que las atenuaciones en la ganancia se dieron en $\omega = 0.1587$ rad/seg y en $\omega = 0.1960$ rad/seg. En el dominio de la frecuencia donde $s = j\omega$, estas frecuencias se pueden expresar como las raíces o polos dominantes del sistema. Por esta razón es que un modelo matemático muy aproximado para el robot EV3 se puede representar con la Ec. (10).

$$G_p(s) = \frac{1}{(s + 0.1587)(s + 0.1960)} \quad \text{Ec. (10)}$$

En la Ec. (10) $G_p(s)$ es el modelo matemático de planta en función de la transformada de Laplace (s).

ENTRENAMIENTO DE LA RED ANFIS

Una vez conseguido el modelo de planta se procede a entrenar la ANFIS a través de SIMULINK. Se diseña en SIMULINK el sistema con el modelo de planta $G_p(s)$ que en la figura 6, se expresa en forma polinómica (Transfer Fcn).

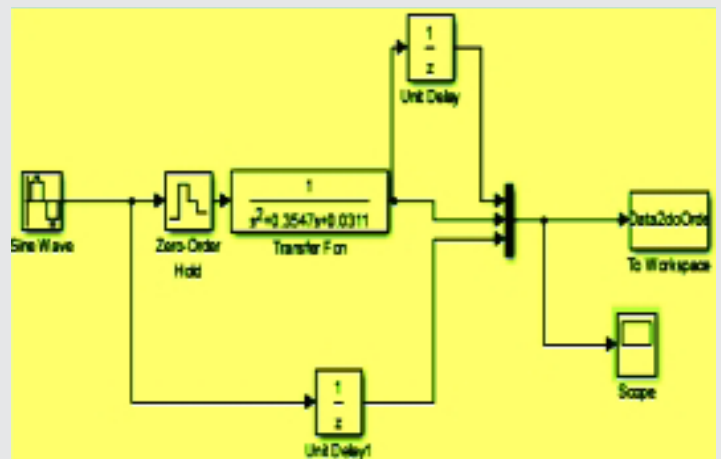


Figura 6. Sistema de simulación que genera los patrones de entrenamiento de la Ec. (8).

La señal a usar como se puede ver en el bloque de entrada de la figura 6, es una función senoidal. La barra obscura vertical que envía las señales colectadas al espacio de trabajo (Workspace) es llamado multiplexor y colecta de arriba abajo los vectores $[x(k)^T, x(k+1)^T; U^T]$. Estos patrones de entrenamiento se almacenan en una variable (Data2doOrden) en el espacio de trabajo. Después de esto, usando la caja de herramientas de MATLAB se diseña la red ANFIS de dos entradas, una para $x(k)$, y otra para $x(k+1)$ como se ve en la figura 7.

La figura 7 muestra la etapa inicial en el diseño de la ANFIS donde se define el número de entradas y se elige el sistema de inferencia difuso Takagi-Sugeno. A continuación se abre otra ventana (Figura 8) donde se estructura la red ANFIS y se lleva a cabo el entrenamiento. Con el fin de hacer breve la explicación del diseño de la red ANFIS, no se mencionan las etiquetas lingüísticas de cada una de las entradas de control. Estas etiquetas son generadas automáticamente con

un número por defecto en la cantidad de las mismas, pero que el usuario puede ajustar tanto en cantidad como en tipo, en la búsqueda de la optimización del controlador.

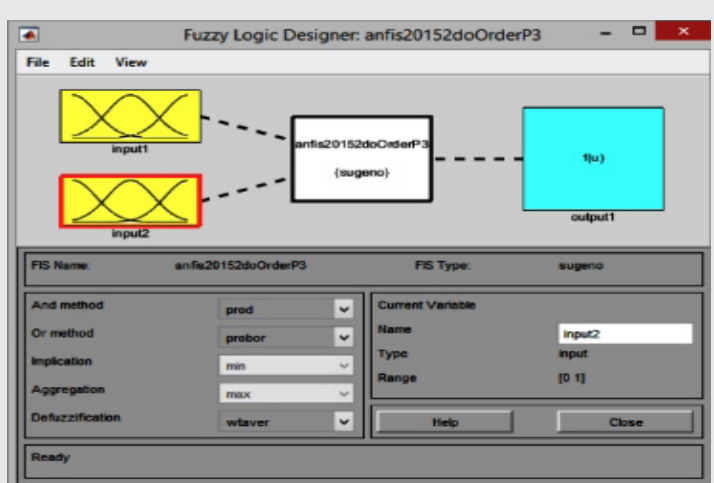


Figura 7. Diseño de la red ANFIS con la caja de herramientas de lógica difusa.

En el recuadro inferior derecho de la Figura 8, se cargan los datos de la variable que contiene los patrones de entrenamiento (Data2doOrden) provenientes del espacio de trabajo. En los recuadros subsiguientes se genera la FIS, se entrena la red y por último se prueba. En el recuadro superior se muestra como la salida de la red ANFIS sigue casi sin error a una función senoidal que se usa para probar la red una vez que esta ha sido entrenada. Cuando se logra que la salida de la ANFIS siga a la perfección a la señal de prueba, entonces decimos que la red ANFIS ha logrado aprender el comportamiento dinámico de nuestro robot.

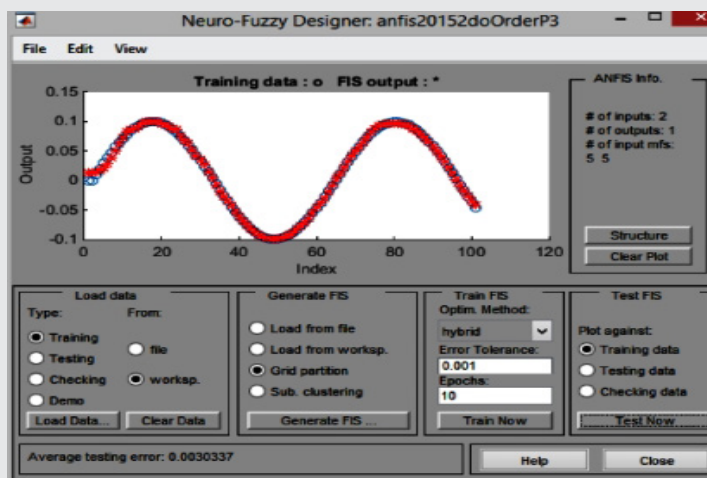


Figura 8. Diseñador de la red ANFIS.

En el recuadro inferior derecho de la figura 8, se encuentra la opción para prueba de la FIS. Si se quiere pueden usarse datos de prueba y de revisión. En este caso particular solo se usaron datos de entrenamiento que resultaron suficientes para la prueba de la red. Una vez diseñado el controlador inverso ANFIS pasamos a la implementación del sistema de control en SIMULINK.

IMPLEMENTACIÓN DEL SISTEMA DE CONTROL ANFIS

El sistema consta de varias etapas (ver Figura 9) que se explican a continuación.

Lectura de posición. Esta parte consiste del encoder del robot EV3, el cual ayuda a monitorizar la posición de los motores grandes del robot EV3 y además lee el valor del sensor ultrasónico en el Display, que es salida del sistema.

Comparación. Este detector de error ayuda a tener una comparación entre el valor leído en el encoder y el valor de la posición a la que se quiere llegar.

Controlador ANFIS. Esta parte se encarga del control del sistema. Controla la posición de los motores y durante el análisis de resultados, fue aquí donde se hicieron los cambios pertinentes de controladores.

Sistema de movimiento de los motores. Esta parte es un sistema de control hacia adelante y hacia atrás el cual hace que el motor camine hacia adelante y hacia atrás con la función Sign (detector de cambio de dirección) el cual da un valor de salida de 1 cuando la entrada es mayor de 0 y un valor de -1 cuando el valor de entrada es menor a 0. Después esta la función Sign1 la cual ayuda a tener la misma polaridad de los motores (necesaria para que el EV3 se desplace en línea recta). Luego se encuentran 2 ganancias para mantener el valor de entrada de velocidad de los motores en 30 (esto evita la zona muerta de los motores). Por último están los bloques de librerías (dlls) de los motores, en los puertos B y C.

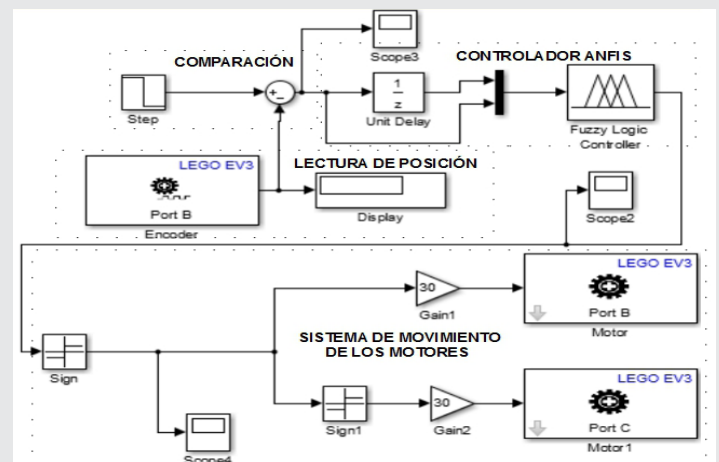


Figura 9. Diseño del sistema de controlador inverso ANFIS del robot EV3 en SIMULINK.

RESULTADOS Y CONCLUSIONES

Es usual en ingeniería de control automático comparar los resultados de una nueva metodología con los paradigmas tradicionales del control automático. En ese sentido, es común usar el controlador Proporcional, Integral, Derivativo (PID). Este paradigma de con-

trol a pesar de su veteranía, es uno de los más usados actualmente en la industria. Esto se debe a su atributo principal, que es su gran robustez. No obstante su robustez, también es capaz de otorgar gran exactitud en su salida de control, minimizando el error de control con mucha eficiencia. La salida del controlador PID genera una señal de control $m(t)$. En la Ec. (11) se describe los términos matemáticos que componen esta acción de control.

$$m(t) = K_p e(t) + K_i \int e(t) dt + K_d \frac{de(t)}{dt} \quad \text{Ec. (11)}$$

Los términos K_p , K_i y K_d , son los parámetros de los modos de control Proporcional, Integral y Derivativo, que se corresponden respectivamente. La señal $e(t)$ es la señal de error que corresponde a la salida del restador de la etapa de comparación en la Figura 9. Diseñar un controlador PID en el SIMULINK es tarea fácil. Solamente se agrega un bloque llamado PID que se sustituye por el controlador ANFIS. El ajuste de parámetros K_p , K_i y K_d en el bloque PID se puede desarrollar de manera automática, ya que tiene una función de auto ajuste. Para el sistema de control estudiado el auto ajuste brindó los siguientes valores: $K_p = 0.001$, $K_i = 0.0001$ y $K_d = 0$.

En la figura 10 se muestra el scope3 que es la señal que sale del restador donde se compara el valor deseado con el valor medido por el encoder. Esta señal es el error de posición final. El encoder va montado en las llantas del EV3 y muestra la posición final del robot. Las lecturas de este sensor son iguales a las del sensor ultrasónico usado para entrenar el controlador ANFIS.

Lo que podemos apreciar en la figura 10, es que una vez que el robot EV3 llegó a una posición de 50 cm, se le ordenó volver al origen. En esta ocasión el robot tardó 11 segundos a una velocidad constante observando un error de aproximadamente +1 cm más allá del origen. También se puede observar una oscilación como de 0.5 cm alrededor del punto de paro.

Después de observar el comportamiento del robot EV3 con un controlador PID, se cambia el controlador de nuevo por el controlador ANFIS. Ahora con el sistema configurado como el de la figura 9 se ejecutan los mismos comandos sobre el robot y esta vez se obtiene un comportamiento como el que se muestra en la figura 11.

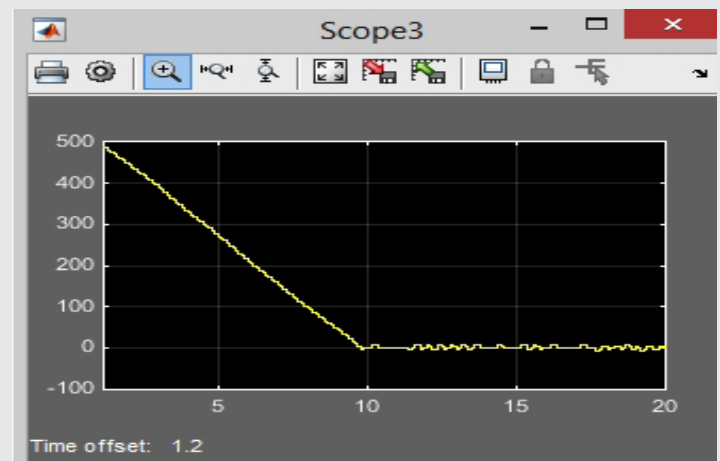


Figura 11. Señal de error del controlador ANFIS usado en el control de posición del EV3.

Lo que se observa en la figura 11, es que para volver al origen con una velocidad constante, igual a la usada para el controlador PID, solo tardó 9 segundos. Dos segundos menos que con el controlador PID. También podemos observar que el error en estado estable es cero, pues no tuvo sobrepaso y permanece en cero. Además podemos observar que las oscilaciones en estado estacionario son de menor frecuencia.

CONCLUSIONES

Según los resultados observados en el apartado anterior, el controlador ANFIS tiene mejores características que el controlador PID. El controlador ANFIS tardó 2 segundos menos en llegar a la posición ordenada, el error fue mucho menor (cero) y las oscilaciones aunque de amplitud aproximada mucho menos frecuentes. Por lo anterior, es ampliamente recomendable el uso de un controlador inverso tipo ANFIS, que en el enfoque actual, resultó mucho mejor que el controlador PID. Además, el controlador ANFIS por sus características adaptativas, puede ser usado en controladores o aplicaciones que busquen la propiedad de poder predecir el comportamiento futuro del sistema, como sucedió en este caso.

Otro de los logros de esta contribución, es el poder ofrecer un reporte sobre el comportamiento de un robot LEGO MINDSTORMS EV3, controlado por la implementación de un control inverso a través de una red ANFIS desarrollada totalmente en SIMULINK. De esta forma, aportamos un enfoque didáctico más atrayente para los estudiantes de pregrado y posgrado.

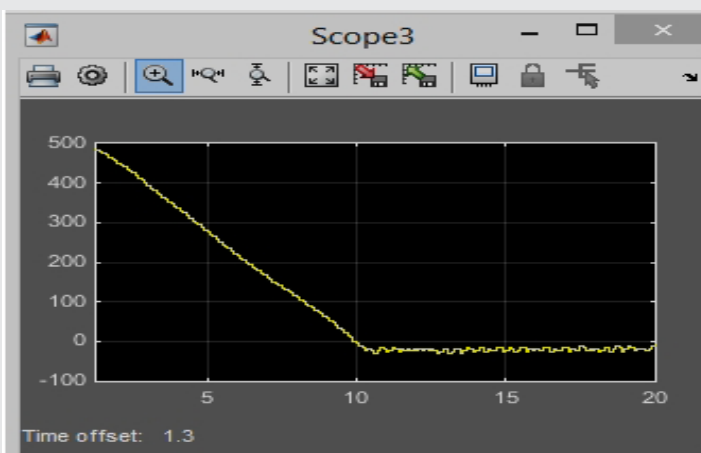


Figura 10. Señal de error del controlador PID usado en el control de posición del EV3.

BIBLIOGRAFÍA

- [1] Bonilla, M. I., Novelo, F. Q., Enríquez, I. G., & Cortés, J. R. (2009). Desplazamiento de un robot con localización y evasión de obstáculos por visión y ultrasonido. CONAGOLFO 2009, Instituto Tecnológico de Orizaba.
- [2] Caltabiano, D., Ciancitto, D., & Muscato, G. (2004, April). Experimental results on a traction control algorithm for mobile robots in volcano environment. In *Robotics and Automation, 2004. Proceedings. ICRA'04. 2004 IEEE International Conference on* (Vol. 5, pp. 4375-4380). IEEE.
- [3] Danahy, E., Eric, W., Brockman, J., Carberry, A., Shapiro, B., & Rogers, C. B. (2014). Lego-based robotics in higher education: 15 years of student creativity. *International Journal of Advanced Robotic Systems*, 11.
- [4] Gallegos Gallegos, E. G., & Gallegos Mayorga, E. M. (2015). Evaluación de la plataforma didáctica Lego Mindstorms EV3 para el concurso ecuatoriano de robótica en la categoría creatividad lego (Doctoral dissertation).
- [5] Hardware Support. LEGO MINDSTORMS EV3 Support from Simulink. Consultado el 7 de noviembre de 2015. www.mathworks.com/hardware-support/lego-mindstorms-ev3-simulink.html.
- [6] Huan Dinh and Tamer Inanc (2009). Low Cost Mobile Robotics Experiment with Camera and Sonar Sensors. American Control Conference, Hyatt Regency Riverfront.
- [7] Jang, J. S. R. (1993). ANFIS: adaptive-network-based fuzzy inference system. *Systems, Man and Cybernetics, IEEE Transactions on*, 23(3), 665-685.
- [8] Kuenzi, J. J. (2008). Science, technology, engineering, and mathematics (stem) education: Background, federal policy, and legislative action, USA.
- [9] LEGO MINDSTORMS EV3 (2013). User Guide. Consultado el 7 de noviembre de 2015. www.nr.edu/csc200/labs-ev3/ev3-user-guide-EN.pdf
- [10] Mendoza Y., Mazo A. (2009). Análisis del modelo ANFIS en el pronóstico de un título de renta variable. Tesis de grado. Universidad Nacional de Colombia.
- [11] Murphy, R. R., Kravitz, J., Stover, S. L., & Shoureshi, R. (2009). Mobile robots in mine rescue and recovery. *Robotics & Automation Magazine, IEEE*, 16(2), 91-103.
- [12] Paolini, C., Huber, G., Collier, Q., & Lee, G. K. (2011). A Web-Based Mobile Robotic System for Control and Sensor Fusion Studies. In *Proc. of the ISCA Int'l Conference on Computers in Industry and Engineering*, Honolulu.
- [13] Pereira, G., Ierache, J. S., & Mazza, N. (2014). Robótica en el contexto de las tecnologías de la educación. En *IX Congreso sobre Tecnología en Educación & Educación en Tecnología* (La Rioja, 2014).
- [14] Samanta, B. (2011). *Computational intelligence: a Tool for Multidisciplinary Education and Research. Proceedings of the 2011 ASEE Northeast Section Annual Conference*, University of Hartford.
- [15] Siegwart, R., Nourbakhsh, I. R., & caramuzza, D. (2011). *Introduction to autonomous mobile robots*. MIT press.
- [16] Tello-Leal, E., Guerrero-Melendez, T. Y., & Saldívar-Alonso, V. P. (2013). Revisión de la plataforma robótica LEGO Mindstorms para aplicaciones educativas y de investigación. *Sistemas & Telemática*, 11(26).



Ciencias de la Computación

Ingeniantes

Instituto Tecnológico Superior de Misantla