

# Seguimiento Autónomo de la Posición de un Objeto por Visión y Control Neuro-difuso en MATLAB

Irma Josefina García Enríquez<sup>1</sup>, Juan Manuel Ramírez Cortés<sup>1</sup>, Mariana Natalia Ibarra Bonilla<sup>1</sup>,  
Ma. del Pilar Gómez Gil<sup>2</sup>

1. Coordinación de Electrónica 2. Coordinación de Ciencias Computacionales  
Instituto Nacional de Astrofísica, Óptica y Electrónica; Tonantzintla, Puebla, México

**Resumen** — Se presenta un sistema de visión por computadora y control difuso que realiza el seguimiento automático de un objeto detectado visualmente a través de una cámara web, montada en una estructura controlada por un motor de pasos. La implementación de este sistema es la primera etapa de un proyecto orientado a la navegación de un robot autónomo controlado por visión que lleve a cabo diversas tareas. El sistema ha sido desarrollado en MATLAB 7.0 usando las cajas de herramienta de lógica difusa y de adquisición y tratamiento de imágenes. Se describe el sistema en cuatro etapas: Adquisición y procesamiento de las imágenes, control difuso, control sobre el motor a pasos e integración a un sistema Neuro-Difuso. Se presentan resultados obtenidos.

**Palabras clave** – difuso, neuronal, control, visión.

## I. INTRODUCCION

La lógica difusa y las redes neuronales son herramientas computacionales adecuadas para usarse en sistemas de automatización y control, y especialmente en mecanismos adaptivos, en función de su capacidad para emular el comportamiento humano basado en razonamiento aproximado [1]. En la actualidad existe un buen número de aplicaciones industriales que requieren de información visual para su correcto funcionamiento: Sistemas de control de calidad por inspección de imágenes, ensamble y acabado de piezas, control y movimiento de inventarios, sistemas de seguridad, o robots para ser operados con diversos objetivos en lugares inaccesibles o peligrosos para el ser humano [2,3,4]. El control difuso en sistemas de visión ha sido utilizado en diversas aplicaciones, una de las cuales es el control para vehículos o robots de navegación autónoma [5,6,7]. Debido al desarrollo de las capacidades de cómputo actuales, las técnicas de procesamiento de imágenes son cada vez más utilizadas en aplicaciones de tiempo real, una de las cuales es el uso de técnicas de visión como elemento de sensado para cerrar el lazo de control. [8,9,10]. Los sistemas de control difuso se basan en la toma de decisiones en forma de razonamiento aproximado de acuerdo con la información contenida en una base de conocimientos. Las variables son representadas en forma de conjuntos difusos definidos en base a funciones de membresía. Los sistemas neuro-difusos incorporan las características de las redes neuronales en cuanto a su capacidad de aprender y generalizar, con el objeto de obtener o de optimizar la partición en conjuntos difusos de las funciones involucradas [1]. En este trabajo se presenta un sistema de seguimiento automático de un objeto

por medio de una cámara web montada en una estructura controlada por un motor de pasos. El objetivo es que el sistema amarre la posición angular de la cámara con la información visual proveniente de la escena detectada. Esta función cubre la primera etapa de un proyecto orientado a proporcionar capacidades de navegación autónoma a un robot controlado por visión. A continuación se presenta el desarrollo de este trabajo en base al software MATLAB.

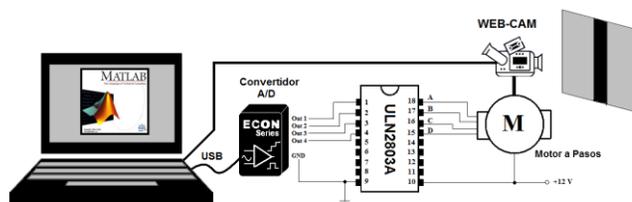


Fig. 1 Diagrama del sistema de seguimiento autónomo guiado por visión.

## II. DESARROLLO

### A. Primera Etapa: Adquisición y Procesamiento de las imágenes

La información visual requerida por el sistema de control difuso es captada a través de una web-cam e importada hacia el entorno de trabajo de MATLAB por medio de sus funciones de adquisición de imágenes. Con el objeto de obtener información numérica acerca de la posición de la línea detectada por la cámara, se lleva a cabo una serie de operaciones de pre-procesamiento sobre la señal de video de entrada. La cámara entrega originalmente una secuencia de imágenes a color con una resolución de 640x480. La imagen de entrada es binarizada con un umbral dado por el nivel de gris 50/256, determinado experimentalmente. La imagen es finalmente sub-muestreada, con lo cual se obtiene una imagen de 214x160 lista para ser analizada. La figura 2 muestra las imágenes antes y después del pre-procesamiento.

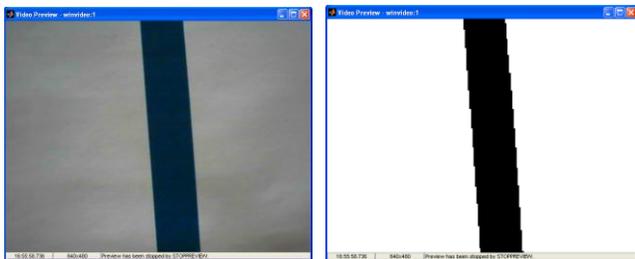


Fig. 2 (a) Imagen captada por la web-cam en tiempo real. (b) Imagen de la línea binarizada.

Con el objeto de determinar numéricamente la posición relativa de la línea, se realiza un escaneo en una fila de la matriz de píxeles. Este escaneo se realiza en la fila número 80 de la matriz (exactamente a la mitad) y se localiza la coordenada en donde ocurre la transición del valor del píxel de blanco a negro. Una vez detectada dicha coordenada, se le resta un valor offset de 107 para ajustar al rango de -107 a 107 y de esta manera ubicar el centro de la imagen en la posición 0.

*B. Segunda Etapa: Control Difuso*

El control difuso requerido para el sistema es a una sola entrada y una sola salida. Éste consiste en tomar como dato de entrada, la información numérica de la posición de la línea detectada a través de la cámara web, para posteriormente entregar como salida un nuevo dato numérico correspondiente al desplazamiento en grados en la trayectoria de la cámara. Las funciones de membresía introducidas a la entrada del sistema son: dos de tipo “trapezoidal” y tres de tipo “triangular”; dichas funciones se describen en la tabla 1 y su representación gráfica se muestra en la figura 3.

**Tabla 1. Funciones de membresía para la entrada**

Variable	Tipo	Partición
Izquierda (Left)	Trapezoidal	[-107 -107 -90 0]
Ligeramente a la Izquierda (LigLeft)	Triangular	[-90 -50 0]
Centro (Center)	Triangular	[-50 0 50]
Ligeramente a la Derecha (LigRight)	Triangular	[0 50 90]
Derecha (Right)	Trapezoidal	[0 90 107 107]

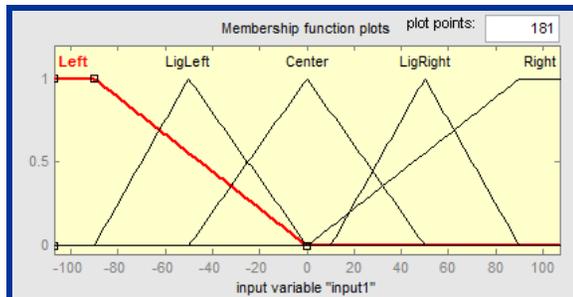


Fig. 3. Funciones de membresía de la entrada.

En el caso de la variable de salida, las funciones de membresía utilizadas en el control de posición del motor se describen en la tabla 2 y se representan gráficamente en la figura 4.

**Tabla 2. Funciones de membresía para la salida**

Variable	Tipo	Partición
Izquierda (Left)	Trapezoidal	[-30 -30 -20 -10]
Ligeramente a la Izquierda (LigLeft)	Triangular	[-20 -10 0]
Centro (Center)	Triangular	[-10 0 10]
Ligeramente a la Derecha (LigRight)	Triangular	[0 10 20]
Derecha (Right)	Trapezoidal	[0 20 30 30]

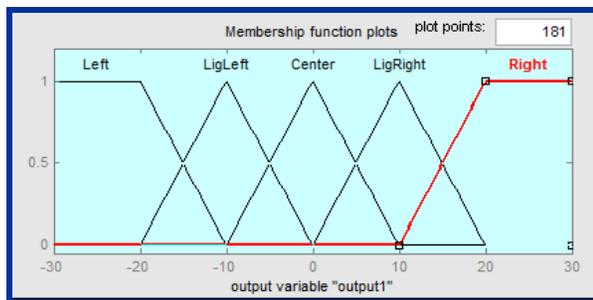


Fig. 4. Funciones de membresía de la salida.

Las reglas de inferencia para el control son basadas en las reglas heurísticas IF (antecedente) THEN (consecuente). En un sistema de control difuso, en el cual las variables de entrada tienen el mismo número de funciones de membresía, el número de reglas está dado por:

$$NR = (NM)^n \tag{1}$$

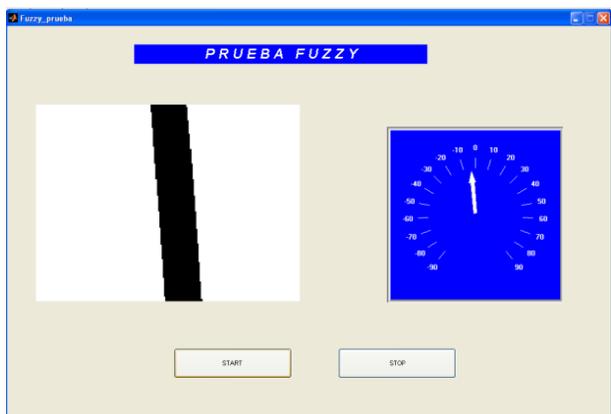
donde:  
 NR=Número de reglas  
 NM=Número de conjuntos difusos en la partición de las variables de entrada  
 n= Número de entradas

De acuerdo con la ecuación 1, en este trabajo se deben implementar cinco reglas. Dado que se requiere que la web-cam siga la trayectoria de la línea, las reglas son propuestas de tal manera que el desplazamiento del motor ocurra en la misma dirección al desplazamiento del objetivo; es decir, que el motor gire en la misma dirección a la línea. De acuerdo con lo anterior, las cinco reglas de inferencia se describen en la tabla 3.

**Tabla 3. Reglas de Inferencia**

Rule	Entrada IF	Salida THEN
1	Izquierda (Left)	Izquierda (Left)
2	Ligeramente a la Izq. (LigLeft)	Ligeramente a la Izq. (LigLeft)
3	Centro(Center)	Centro(Center)
4	Ligeramente a la Der. (LigRight)	Ligeramente a la Der. (LigRight)
5	Derecha(Right)	Derecha(Right)

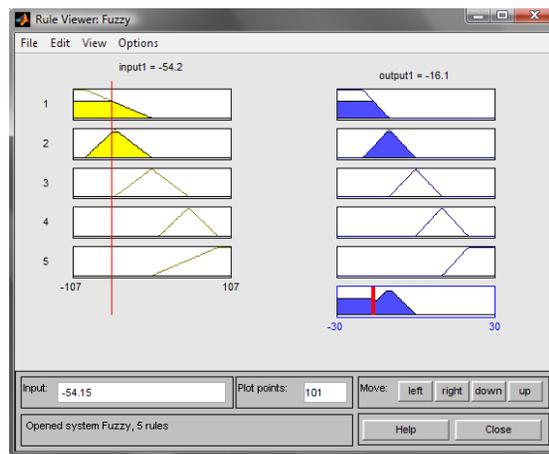
Con la finalidad de comprobar el buen funcionamiento del control difuso, se realiza una simulación bajo el ambiente GUI en Matlab, en la cual se introduce el uso de un activex (aguja) para imitar el movimiento del motor. Como puede apreciarse en la figura 5 la imagen captada por la cámara se encuentra centrada ligeramente a la izquierda. El desplazamiento es detectado por el sistema de control y representado por la posición de la aguja.



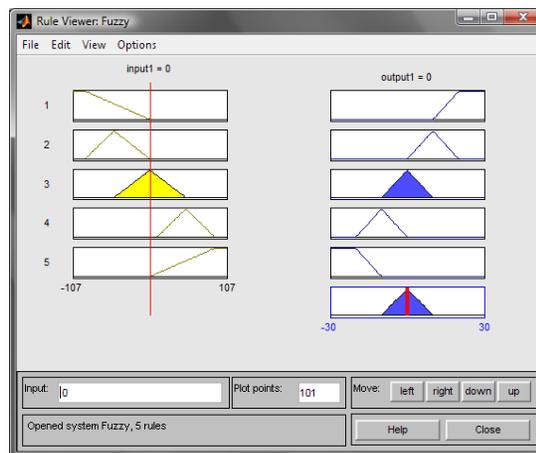
*Fig. 5 .Simulación en GUI.*

El código en Matlab del control difuso inicia con un ciclo while para dejar al sistema operando en tiempo real. Se evalúa la variable de entrada correspondiente al valor numérico del primer pixel negro que se detecta. El valor de entrada es monitoreado para verificar que se encuentra entre el rango de -8 y +8 pixeles. Lo anterior se debe a que el grueso de la línea negra abarca aproximadamente 16 pixeles, 8 en cada lado. Finalmente, se cierra el lazo de control retroalimentando la salida. Haciendo uso de la herramienta gráfica del Sistema de Inferencia Difuso de Matlab se obtienen los gráficos de la figura 6, en los cuales se observa

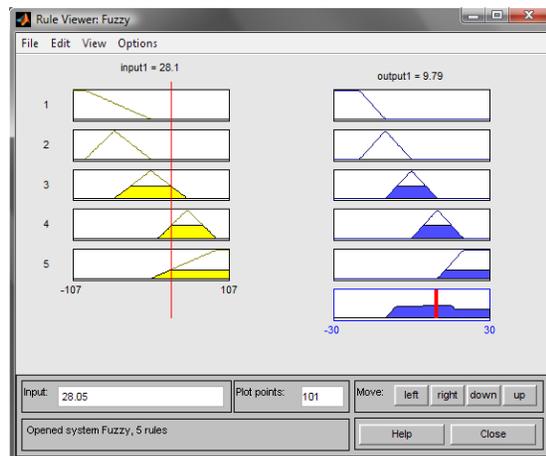
el comportamiento de las funciones de membresía del sistema.



(a)



(b)



(c)

*Fig. 6. Control Difuso en el editor difuso de Matlab. a) Entrada a la Izquierda, Salida a la Izquierda. b)Entrada en el centro, Salida en el centro. c) Entrada ligeramente a la derecha, Salida ligeramente a la derecha.*

*C. Tercera Etapa: Control sobre el Motor a Pasos*

La tercera etapa consiste en el circuito de control del motor a pasos de tipo unipolar, el cual incluye una etapa de potencia y una tarjeta de adquisición/entrega de datos. Estos dispositivos tienen el objetivo de servir como interfaz entre la computadora y el motor. La tarjeta utilizada es la Data Translation USB, modelo DT9812-10V de ocho canales analógicos y ocho salidas digitales (Ver Fig. 7). Para la etapa de potencia se utilizó el C.I. ULN2803A, tal como aparece en el diagrama de la figura 8. Para mover el motor se debe elegir la secuencia del paso. En este sistema se utiliza la secuencia de “medios pasos” descrita en la Tabla 4, la cual entrega un movimiento mas suave en comparación a la secuencia de pasos enteros (wave drive).

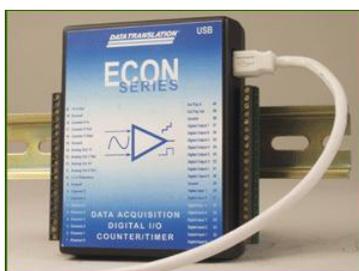


Fig. 7. Data Translation USB, modelo DT9812-10V

**Tabla 4. Secuencia de medios pasos para el motor unipolar**

Paso	A	B	C	D	Decimal	
1	ON	OFF	OFF	OFF	8	
2	ON	ON	OFF	OFF	12	
3	OFF	ON	OFF	OFF	4	
4	OFF	ON	ON	OFF	6	
5	OFF	OFF	ON	OFF	2	
6	OFF	OFF	ON	ON	3	
7	OFF	OFF	OFF	ON	1	
8	ON	OFF	OFF	ON	9	

Como se aprecia en la tabla anterior la secuencia es:  
 Secuencia a la Derecha: [8 12 4 6 2 3 1 9]  
 Secuencia a la Izquierda: [9 1 3 2 6 4 12 8]

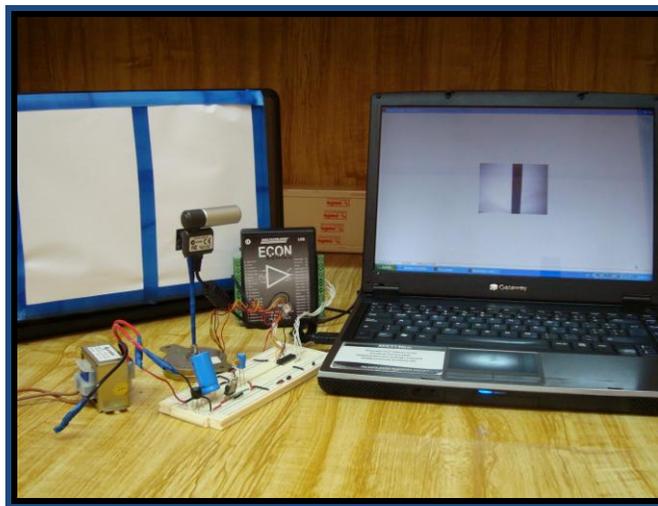


Fig. 8. Sistema de control de seguimiento autónomo guiado por visión.

El control de las secuencia para el motor, se representa en el diagrama de flujo de la figura 9.

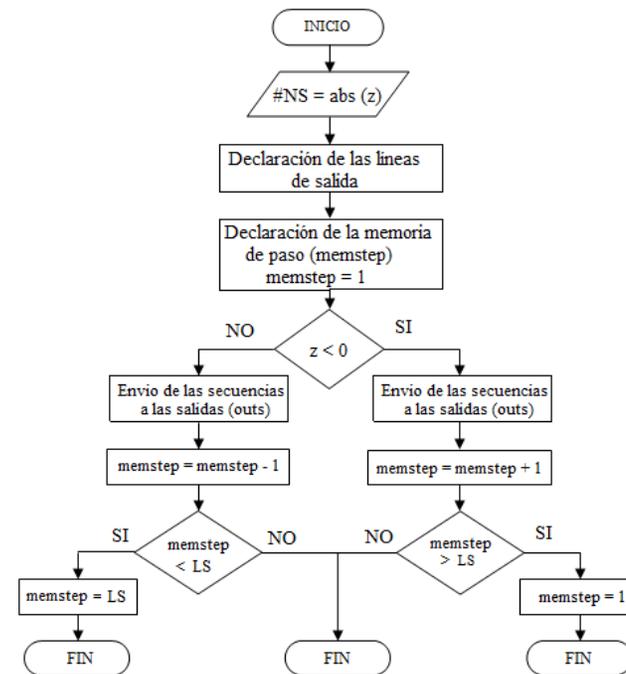


Fig.9. Diagrama de flujo de la secuencia del motor a pasos.

donde:  
 #NS = Número de veces que se envía la secuencia para el motor.  
 z = Valor numérico de salida del control difuso.  
 memstep = Variable de memoria de paso.

LS = Longitud de la matriz de secuencia.

El proceso descrito en el diagrama de flujo se encuentra dentro de un ciclo “for” en el código de Matlab, el cual se repite hasta llegar al número de veces en que se envía la secuencia. Se agrega la variable de la memoria de paso, ya que el sistema exige guardar el dato numérico de la secuencia y de esta manera determinar si el siguiente paso corresponde a un giro a la derecha o a la izquierda.

*D. Cuarta Etapa: Incorporación de datos del Sistema Neuro-Difuso.*

En esta última etapa se integra todo el sistema difuso anterior a un sistema Neuro-Difuso. Haciendo uso de las ventajas que proporciona el trabajar con redes neuronales, se desea que éstas últimas proporcionen las particiones óptimas para las funciones de membresía del sistema difuso. Sin embargo, toda red neuronal necesita un grupo de datos de entrenamiento y validación para inicializarse. De acuerdo a los rangos de operación del sistema se obtiene la función de la recta de control. Sus valores están determinados por el número de pixeles y su correspondiente valor en grados a la salida. La función de la recta de control, obtenida empíricamente de acuerdo con los rangos de operación del sistema es:

$$f(x)=0.2803738315 x. \quad (2)$$

Una vez que la red neuronal obtiene los datos que resultan de la evaluación de la función anterior, está preparada para encontrar las particiones óptimas de las funciones de membresía de la entrada y salida del controlador difuso. En esta etapa se utilizaron para los conjuntos difusos, tanto funciones de membresía triangular y trapezoidal, como funciones tipo campana. Se observó un desplazamiento mas suave en el último caso. Para el modelado de esta etapa se utiliza la herramienta ANFIS (adaptive neuro-fuzzy inference system) de Matlab. El modelo ANFIS incluido en Matlab consta de cuatro etapas fundamentales: Generación de los conjuntos de datos para el aprendizaje. Generación de un SIB de tipo Sugeno como estructura inicial. Algoritmo de aprendizaje. Validación del resultado obtenido. Utilizando la ecuación 2 se genera un conjunto de 100 puntos equidistantes y se guardan en una matriz. Posteriormente, de esta matriz se extraen 50 puntos para utilizarlos como datos de entrenamiento y los otros 50 como datos de prueba. Estos valores se muestran en la figura 10. Los datos de las matrices anteriores son utilizados para definir la estructura del ANFIS y establecer los valores iniciales de sus parámetros. En seguida, se especifica un ANFIS de primer orden con funciones de membresía de tipo campana. La partición del espacio de entrada para formar las funciones de membresía iniciales se visualizan en la figura 11.

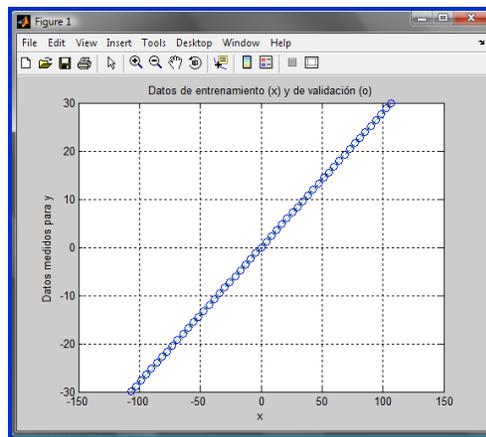


Fig. 10. Datos de entrenamiento y datos de prueba.

Con los datos obtenidos se lleva a cabo el proceso de aprendizaje híbrido con validación para la optimización paramétrica durante 1000 iteraciones. Las nuevas funciones de membresía ya optimizadas se pueden apreciar en la figura 12.

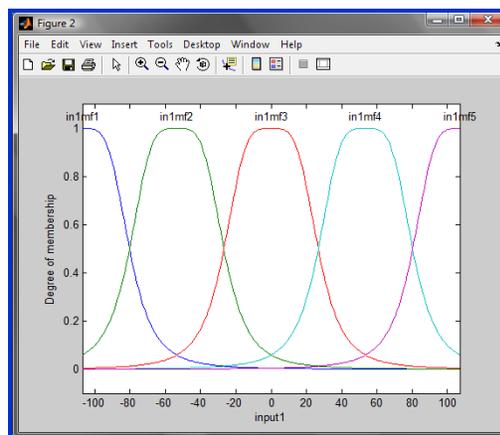


Fig. 11. Visualización de las funciones de membresía iniciales de tipo campana.

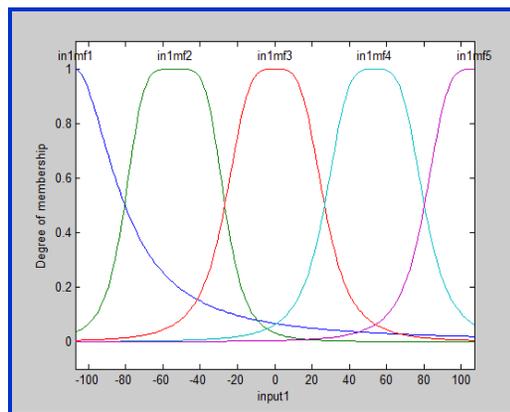


Fig. 12. Visualización de las funciones de membresía entregadas por la red neuronal.

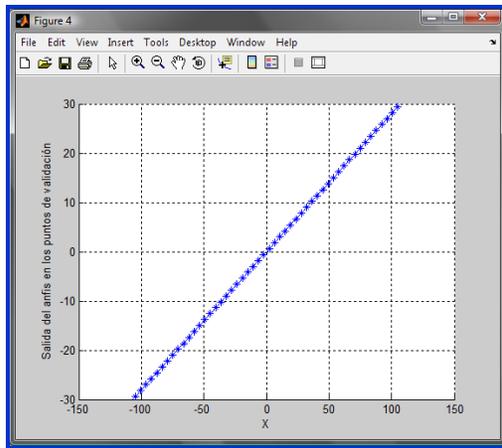


Fig. 13. Evaluación de la Respuesta del ANFIS. Salida del Anfis y función original.

A continuación, se evalúa la respuesta del ANFIS comparando sus resultados con los datos de validación. La representación gráfica de dicha comparación se representa en la figura 13. Finalmente, las particiones obtenidas se incorporan al programa que contiene el control del sistema.

### III. RESULTADOS

El sistema desarrollado realiza adecuadamente el amarre de la posición angular de la cámara con la información visual proveniente de la escena detectada. En la figura 14 se muestra una imagen del sistema funcionando, en donde a línea capturada por la web-cam se mantiene siempre centrada, debido a que el movimiento del motor compensa los desplazamientos del objetivo a través del control neurodifuso implementado.



Fig. 14. Imagen del sistema funcionando; la web-cam apunta hacia donde se encuentra la línea.

### IV. CONCLUSIONES

Se presentó un sistema de seguimiento autónomo de la posición de un objeto por control neurodifuso, implementado completamente en MATLAB, con ayuda de las cajas de herramientas de control neurodifuso y de adquisición y procesado de imágenes. El sistema desarrollado hasta ahora es un controlador difuso, cuyas funciones de membresía fueron optimizadas mediante el uso de redes neuronales a través del modelo ANFIS incluido en MATLAB. Como puede observarse, las particiones óptimas son muy parecidas a las iniciales debido a que la función de control propuesta es una recta. Este trabajo consistió en la primera etapa de un proyecto en el cual se pretende proporcionarle capacidades de navegación autónoma a un robot controlado por visión a través de técnicas de control difuso.

### V. AGRADECIMIENTOS

Los autores agradecen al CONACYT por el apoyo otorgado a través de las becas para estudios de Maestría con números 212442 y 212443.

### REFERENCIAS

- [1] R. Young-Jae, L. Young-Cheol, "Neuro Fuzzy Control System for Vision-based autonomous vehicle" in *Fuzzy System Conference IEEE International*, Vol. 3, 1999, pp. 1643-1648.
- [2] Flan N.N., Moore K.L., "A small mobile robot for security and inspection", *Control Engineering Practice*, vol. 10, issue 11, 2004, pp. 1265-1270. ISSN: 0967-0661.
- [3] Retreya Ramírez. "Redes Neuronales Difusas para modelado vía agrupamiento en línea". Instituto Politécnico Nacional. México, 2005.
- [4] D. Stipanicev, M. Cecic. "Eye-Hand coordination based on fuzzy vision" in *Fuzzy System Conference IEEE International*, 1992, pp. 29-35.
- [5] M. Todd, et al. "MANIAC: A next Generation Neurally Based Autonomous Road Follower". *The Robotics Institute, Carnegie Mellon University, Pittsburgh PA 15213*, 1999.
- [6] J.E. Naranjo, et al., "Using Fuzzy Logic in automated Vehicle Control", *IEEE Intelligent Systems*, 2007, pp. 36-45.
- [7] E. Tunstel, A. Howard and H. Seraji, "Rule-Based reasoning and neural network perception for safe off-road robot mobility", *Expert Systems*, Vol. 19, No. 4, pp. 191-200, September 2002.
- [8] Yu Wua Wong, et al, "Vision System for a robot Guide System", in *Fourth International Conference on Computational Intelligence, Robotics and Autonomous Systems*, 2007, pp. 337-341.
- [9] N. Popescu, J. Zhang, "Image Feature Extraction for a Robust Vision-Based Localization on a Mobile Robot", Politehnica University of Bucharest and University of Hamburg, 2003.
- [10] J.M. Ramírez, P. Gómez-Gil, F. López Larios, "A robot-vision System for Autonomous Vehicle Navigation with Fuzzy-Logic Control using Lab View", en el Cuarto Congreso de Electrónica, Robótica y Mecánica Automotriz, CERMA 2007, pp. 295-300.