



Ciencias
de la
Computación

Ingeniantes

Control de visión inteligente por lógica difusa y RNA para el seguimiento de un objeto aplicado a una estructura robótica de 6 GDL



Colaboración

Mariana N. Ibarra-Bonilla; Iván Reyes-Castillo; Fernando Sánchez-Texis, Instituto Tecnológico Superior de Atlixco; Fernando J. Quiñones-Novelo, INTESC: Electronics & Embedded

RESUMEN: Este artículo presenta el desarrollo de un control de visión por computadora, el cual incorpora técnicas de inteligencia artificial: lógica difusa y redes neuronales artificiales (RNA) para ejecutar el seguimiento autónomo de la posición de un objeto de color específico. La información visual de la posición y color del objeto se obtienen por medio de una cámara web montada en un brazo robótico. El movimiento de cada articulación del brazo robot es controlado por lógica difusa y una RNA perceptrón-multicapa. La RNA es entrenada para clasificar y reconocer el color del objeto. Estos algoritmos son desarrollados utilizando las cajas de herramientas de lógica difusa, redes neuronales y procesamiento de imágenes de Matlab® v8.5. Este trabajo responde a las aplicaciones de inteligencia artificial realizadas para un brazo robótico servo-controlado de 6 grados de libertad (GDL) diseñado en CATIA v5.0 y manufacturado dentro de las instalaciones del Instituto Tecnológico Superior de Atlixco.

PALABRAS CLAVE: control difuso, red neuronal artificial, visión, brazo robótico.

ABSTRACT: This paper presents the development of a vision computer control, which incorporates artificial intelligence techniques: fuzzy logic and artificial neural networks (ANN) to performance the autonomous tracking of the position of a specific color object. The visual information of the position and color of the object are obtained through a web-cam mounted on a robotic arm. The movement of each articulation of the robotic arm is controlled using fuzzy logic and a perceptron-multi-layer ANN. The ANN is trained to classify and recognize the color object. These algorithms are developed using the fuzzy logic, neural networks and image processing toolboxes of Matlab® v8.5. This work responds to artificial intelligence applications made for a 6 degree of freedom (DOF) servo-controlled robotic arm designed using CATIA v5.0 and manufactured using the facilities within the Instituto Tecnológico Superior de Atlixco.

KEYWORDS: fuzzy control, artificial neural network, vision, arm-robot.

INTRODUCCIÓN

Las aplicaciones de inteligencia artificial, como la lógica difusa y las redes neuronales artificiales, están creciendo significativamente en las áreas de automatización, robótica y control. Uno de los principales beneficios de la lógica difusa, es su capacidad de incorporar el conocimiento experto de un ser humano en lo referente a cómo controlar un sistema, lo que permite diseñar controladores libres de un modelo matemático. Así que el control difuso puede emplearse para controlar plantas no lineales o cualquier sistema cuyo modelo matemático pudiera ser complejo [1].

Por otra parte, las redes neuronales artificiales (RNA) incorporan la capacidad de aprender, y al igual que los humanos, pueden realizar tareas de comparación de patrones [2]. Por ello, las redes neuronales y los sistemas de control difuso son excelentes para el desarrollo de sistemas artificiales, porque pueden realizar el mismo tipo de tratamiento de la información como lo hace el cerebro humano.

Además de las técnicas de inteligencia artificial, actualmente, múltiples aplicaciones robóticas han incorporado información visual a sus sistemas de posicionamiento [3, 4, 5]; por ejemplo en la operación de robots industriales, un sistema de visión artificial puede hacer que un robot manipulador sea mucho más versátil, permitiéndole tratar de forma adecuada las variaciones en la posición y orientación de las piezas que deba manipular. Los sistemas de visión por computadora utilizan imágenes y video para detectar, clasificar y seguir eventos, y de esta manera, interpretar escenas del mundo real. Las capacidades actuales de cómputo, ya permiten el uso frecuente de técnicas de visión, como elemento de medición para cerrar el lazo de control.

En robótica, los sistemas de inferencia difusos han sido implementados en diversas aplicaciones, tales como control de posición de servomotores [6, 7] y control de brazos robot [8, 9]. 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 conocimiento. Las variables del sistema son representadas en forma de conjuntos difusos, definidos en base a funciones de membresía. Las aplicaciones del control difuso que incorporan información visual, se encuentran principalmente en la navegación de vehículos o robots autónomos. Por ejemplo, Taryudi y Wang en [10] presentan un experimento que incorpora un control neuro-difuso para comandar los movimientos de un brazo robot de 6GDL (grados de libertad). En el experimento, un objeto de color es colocado dentro del área de alcance o manipulación del brazo. El propósito del experimento es tomar dicho objeto con el brazo y trasladarlo hacia una nueva posición. La información de la posición del objeto, es obtenida mediante dos cámaras de video idénticas. Las cámaras fueron colocadas en una configuración bifocal, con el objetivo de capturar la escena completa, la del brazo y el objeto de color. Con esta información la red neuronal, calcula la posición del objeto con respecto de la posición del brazo y controla los movimientos de este último. A diferencia del anterior, en el presente trabajo se propone el uso de una sola cámara, para realizar el seguimiento de un objeto de color en tiempo real.

Ligutan, et al. en [11] describen un sistema difuso para el control de movimiento de un brazo robot de 6GDL. El objetivo de este trabajo, es similar al anterior en lo que respecta a la tarea que debe realizar el brazo. Sin embargo, en dicho trabajo, se emplea una cámara Kinect® y un sensor inercial MP6050 para obtener las imágenes, coordenadas y datos de entrada. La presente propuesta se diferencia

de dicho trabajo, por incorporar una técnica adicional de inteligencia artificial: RNA.

Zapata en [12] describe el uso de la lógica difusa para el seguimiento de un objeto de color, que se desplaza dentro de su campo visual, usando un brazo robot de 2GDL y una sola cámara. El seguimiento se realiza mediante la identificación de únicamente dos colores. Para el cálculo de la posición del objeto, el autor emplea técnicas ya conocidas de procesamiento de imágenes. La cámara montada sobre el brazo robot, es desplazada en un espacio bidimensional o plano. El control difuso calcula las posiciones que deben de adoptar las articulaciones del brazo, para mantener al objeto en el centro del campo visual de la cámara. Dicho trabajo tiene similitud con la presente propuesta, no obstante, en este artículo se presenta un control difuso aplicado a un brazo robot de 6GDL y un clasificador de 7 colores, lo que aumenta el grado de complejidad del sistema.

En general, las redes neuronales, como los sistemas de control o de toma de decisiones basados en lógica difusa, son excelentes para desarrollar sistemas que pueden ejecutar el mismo tipo de procesamiento de información que el cerebro humano.

El objetivo del presente trabajo es incorporar técnicas de inteligencia artificial: control difuso y redes neuronales, al control de movimiento de un brazo robot de 6 grados de libertad (GDL). De manera que al visualizar un objeto de color, el robot realice el seguimiento autónomo de dicho objeto, manteniéndolo en el centro del campo visual de la cámara mientras se desplaza. La información visual de la posición y color del objeto, se obtienen por medio de una cámara web, montada sobre el brazo robot.

Adicionalmente los algoritmos desarrollados en este proyecto servirán como una herramienta didáctica para los cursos de Robótica y Control Digital que se imparten actualmente en el plan de estudios de Ingeniería Mecatrónica del Instituto Tecnológico Superior de Atlixco. De esta manera, los estudiantes tendrán una visión inmediata de los conceptos teóricos-prácticos involucrados en estos campos de la ingeniería.

MATERIALES Y MÉTODOS

El diseño mecánico del brazo robótico incorpora 2 GDL para el hombro y 1GDL para cada articulación restante que conforma el brazo, es decir, el codo, antebrazo, muñeca y soporte de la cámara, completando un total de 6 GDL. Cada GDL es ejecutado por la acción de un servomotor. El diseño CAD del brazo robot, fue realizado en el software CATIA v5 y se presenta en la Figura 1. Dicha figura muestra la distribución de los servomotores y el montaje de la cámara web. Los servomotores del número uno al cinco, son de la marca HITEC-RCD® de 15 Kg-cm de torque, mientras que el servomotor número seis es de la marca TowerPro® de 1.5 Kg-cm de torque. En la construcción de las piezas que conforman la estructura del brazo robot,

se utilizó como materia prima un panel compuesto por dos caras de aluminio y cuerpo interno de PVC espumado de alta densidad. Cada pieza fue manufacturada en una máquina CNC, de la marca Charly-4U, que se encuentra en el Laboratorio de Manufactura, dentro de las instalaciones del Instituto Tecnológico Superior de Atlixco.

La cámara, montada sobre el brazo robot, es un sensor CIF-CMOS de 640 x 480 pixeles de resolución, a 30 cuadros por segundo, con conexión USB. Para el control de movimiento de los seis servomotores, se utilizó la tarjeta electrónica Pololu-Maestro de seis canales con salida PWM. La tarjeta se comunica a un ordenador mediante un puerto CDC-USB a través de comandos AT.

El sistema de control de movimiento y el clasificador de color, han sido desarrollados utilizando las cajas de herramientas de procesamiento de imágenes, redes neuronales y lógica difusa de la plataforma MATLAB. Actualmente, MATLAB está disponible en cualquier universidad o industria, y es usado, entre otras aplicaciones, en el diseño de sistemas de control.



Figura 1. Modelo CAD del brazo robot de 6 GDL.

Metodología para el reconocimiento del color: entrenamiento de la RNA.

Descripción de la RNA.

Una RNA es un modelo computacional, inspirado en la función de las redes neuronales biológicas, el cual está compuesto por elementos de procesamiento (neuronas) y múltiples conexiones entre ellos.

Para esta investigación, se utilizó una RNA tipo perceptrón-multicapa backpropagation para el reconocimiento de siete colores diferentes: rojo, verde, azul, amarillo, cian, magenta y naranja.

El perceptrón es un tipo de red de aprendizaje supervisado, esto significa que la red necesita conocer los valores esperados, t , para cada una de las entradas presentadas, p . De esta manera, el comportamiento de la red está definido por pares entrada-salida de la forma

[p , t]. La arquitectura de la RNA se presenta en la Figura 2 y consiste de dos capas. La capa oculta tiene 10 neuronas y una función de activación sigmoide. La capa de salida tiene 7 neuronas y una función de activación lineal. Las 7 neuronas en la capa de salida representan las siete condiciones de clasificación. De esta manera, la salida está determinada por la siguiente expresión:

$$y = \text{pureline} \{ \mathbf{W}^2 \text{sigmoide} (\mathbf{W}^1 \mathbf{p} + \mathbf{b}^1) + \mathbf{b}^2 \} \quad \text{Ec. (1)}$$

donde, y es el valor de salida de la red, W^1 y W^2 son las matrices de pesos asociadas a las capas 1 y 2, p es el vector de entrada y b^1 y b^2 son los vectores de umbrales asociados a cada capa.

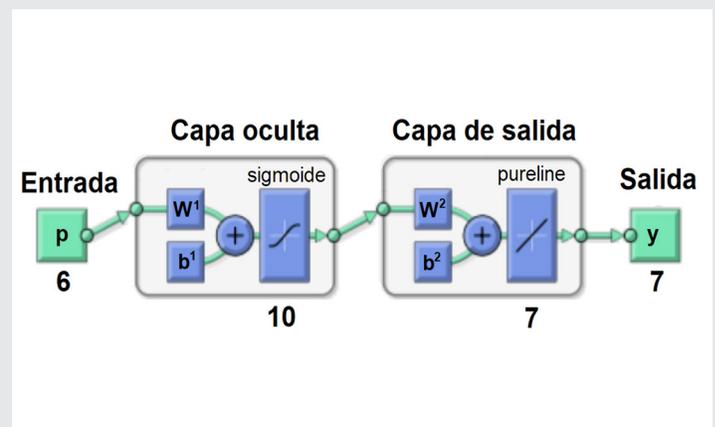


Figura 2. Estructura de la red neuronal para la clasificación de color.

Extracción de características.

La información visual, obtenida de la cámara web (sensor de visión), es importada al entorno de trabajo de MATLAB, usando las funciones de adquisición de imágenes, para formar el vector de entrada a la RNA, p . El vector p está formado por seis características: las tres componentes RGB (del inglés Red, Green, Blue) y las tres componentes HSV (del inglés Hue, Saturation, Value), para cada color. En el modelo RGB, cada color aparece en sus componentes espectrales primarias: rojo, verde y azul, mientras que en HSV cada color aparece en sus componentes: matiz, saturación y brillo.

Para obtener estas características se lleva a cabo una serie de operaciones de pre-procesamiento sobre la señal de video de entrada: la imagen capturada, de 640x480 pixeles, es segmentada a 352x288 pixeles y binarizada. Posteriormente, se eliminan los objetos menores a 30 pixeles y se detecta el contorno exterior de cada objeto detectado. Los pixeles contenidos en cada contorno, representan a cada objeto presente en la imagen, en consecuencia, su posición ya es conocida. La imagen capturada originalmente se descompone, respectivamente, en las componentes RGB y HSV, y se realiza un promedio de los valores contenidos den-

tro de las coordenadas correspondientes a cada contorno. Así el vector de entrada está conformado por:

$$p = [R \ G \ B \ H \ S \ V]^T \quad \text{Ec. (2)}$$

El aprendizaje de la RNA es supervisado, por lo que es necesario establecer previamente los valores deseados (vectores objetivo) para cada conjunto de datos presentes a la entrada. El vector objetivo tiene el mismo número de elementos que neuronas en la capa de salida. Un ejemplo del vector objetivo para el color rojo es:

$$\text{rojo} = [1 \ 0 \ 0 \ 0 \ 0 \ 0]^T \quad \text{Ec. (3)}$$

Entrenamiento de la RNA.

El propósito general del entrenamiento de una RNA es obtener un conjunto de vectores o matriz de pesos, W, y un conjunto de niveles de umbral de activación b, de tal forma que se minimice el error entre la salida de la red neuronal con el valor deseado del sistema a modelar por la RNA [2]. Los algoritmos de entrenamiento son los encargados de modificar los pesos de cada neurona para identificar a lo largo de una serie de iteraciones, cuál es el patrón que mejor se ajusta a los objetivos. En la caja de herramientas de redes neurales de MATLAB se incluyen diferentes algoritmos de entrenamiento que usan métodos basados en gradiente o Jacobiano [13]. De acuerdo con Hagan, Demuth y Beale en [2] y [13] los algoritmos más rápidos son Levenberg-Marquardt (trainlm) y BFGS Quasi Newton (trainbfg), y para aplicaciones con reconocimiento de patrones, los algoritmos de Gradiente Conjugado Escalado (trainscg) y Retropropagación resistente (trainrp) son las elecciones apropiadas. En esta investigación se propuso utilizar como algoritmos de entrenamiento para la red neuronal backpropagation, los siguientes: trainlm, trainbfg y trainscg.

Para formar los pares entrada/salida para entrenamiento y validación de la RNA, se obtuvieron 5 muestras de 4 tonalidades diferentes por color, así se generaron 20 vectores de entrada, p, por cada color. Estos datos se dividieron en dos sub-conjuntos, el primero para entrenamiento y el segundo para validación. La matriz de entrenamiento está conformada por las componentes R-G-B-H-S-V obtenidas de las tres primeras tonalidades de cada color, y la matriz de validación está compuesta por las componentes de la cuarta tonalidad. De acuerdo con lo anterior, el entrenamiento de la red utiliza la técnica conocida por su nombre en inglés leave-one-subject-out-cross-validation. Este proceso es repetido hasta que los datos de cada tonalidad han sido usados para el conjunto de validación.

Metodología para el seguimiento autónomo del objeto: control difuso.

El principal propósito de la lógica difusa es permitir el uso de conceptos vagos para caracterizar las variables de un sistema usando términos lingüísticos humanos. Esto es posible debido a que, en la teoría de conjuntos difusos, un

elemento no está restringido para pertenecer o no pertenecer a un conjunto. En cambio, un elemento puede tener un grado de pertenencia o membresía intermedio en todo el rango (0, 1). Entonces, un sistema basado en lógica difusa proporciona una metodología formal para representar, manipular e implementar el conocimiento heurístico de un ser humano sobre cómo controlar un sistema [14]. La estructura básica de un sistema difuso, con cinco bloques funcionales, se muestra en la Figura 3. La función de cada bloque es la siguiente:

- La base de reglas, contiene las reglas difusas de tipo IF-THEN.
- La base de datos, define las funciones de membresía de los conjuntos difusos usados en las reglas difusas.
- La unidad de toma de decisiones, ejecuta las operaciones de inferencia sobre las reglas.
- La interface de fusificación, transforma las entradas a grados de membresía dentro de los valores lingüísticos.
- La interface de defusificación, transforma los resultados de la inferencia a los valores de salida.

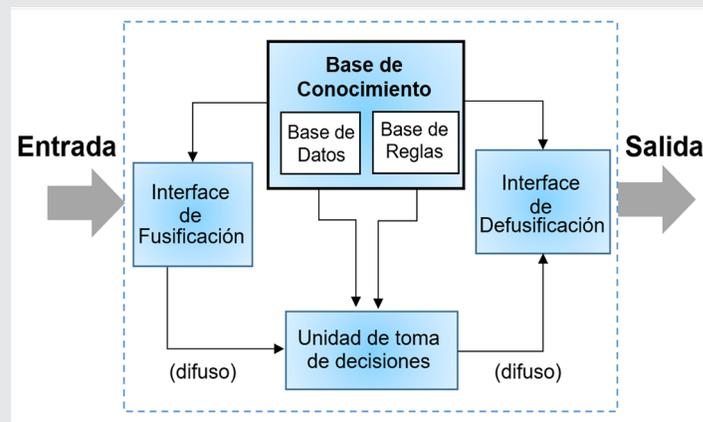


Figura 3. Sistema basado en lógica difusa.

Para este trabajo, el objetivo del controlador difuso es entregar la posición angular de cada servomotor instalado en las articulaciones del brazo robot, tomando como entrada el error en la posición, en coordenadas X-Y, del objeto detectado por la cámara con respecto al centro de la imagen capturada. El esquema del diagrama a bloques del controlador se presenta en la Figura 4, donde el bloque del controlador difuso está compuesto por los elementos presentes en la Figura 3.

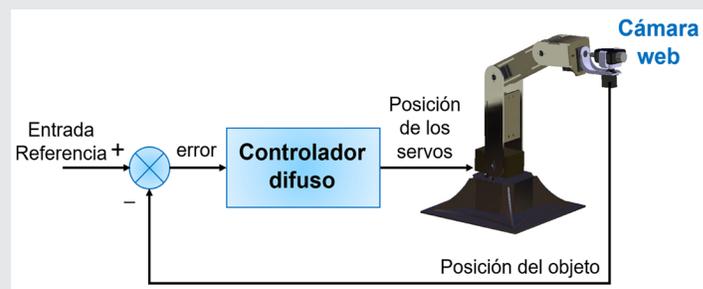


Figura 4. Diagrama a bloques del control de posición difuso.

Definición de entradas y salidas del controlador difuso.

El control difuso requerido para este trabajo es un MIMO, del inglés Multiple-Input-Multiple-Output. Se definen dos variables lingüísticas de entrada: error-X y error-Y, cada una con particiones en 7 funciones de membresía gaussianas etiquetadas como grande-negativo (G-Neg), negativo (Neg), poco-negativo (P-Neg), cero (C), poco-positivo (P-Pos), positivo (Pos) y grande-positivo (G-Pos). La primera entrada error-X corresponde a la diferencia entre el centro de la imagen y el centro del objeto a seguir, medida sobre el eje X de la imagen. La segunda entrada error-Y es similar, pero el error se mide sobre el eje Y. Las funciones de membresía de las variables lingüísticas de la entrada se presentan en la Figura 5.

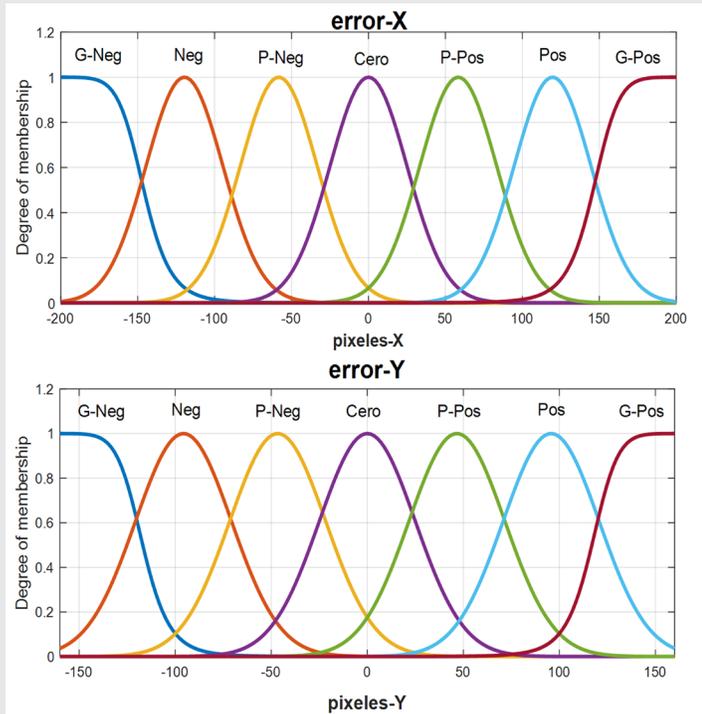


Figura 5. Funciones de membresía de las variables de entrada.

Se definen seis variables lingüísticas para la salida del controlador: servo1-servo6. Cada variable de salida representa a cada uno de los grados de libertad del robot. Para definir los parámetros de las variables de salida fue necesario realizar pruebas para caracterizar los movimientos de cada servomotor. Estos parámetros se definen de acuerdo a la duración del ancho del pulso de la señal de control, PWM (del inglés, Pulse Width Modulation), para el posicionamiento de un servomotor [15], pues la duración del ancho del pulso indica el ángulo de posición del servomotor. En este trabajo, el rango considerado para el ancho de pulso es de 500us - 2500us, lo que permitirá al servomotor moverse entre una posición angular mínima y máxima (0° - 180°). Por ejemplo, el valor 1500us indica la posición central o neutra (90°), cualquier otro valor dentro del rango corresponde a las posiciones intermedias. Un esquema de esta señal se presenta en la Figura 6.

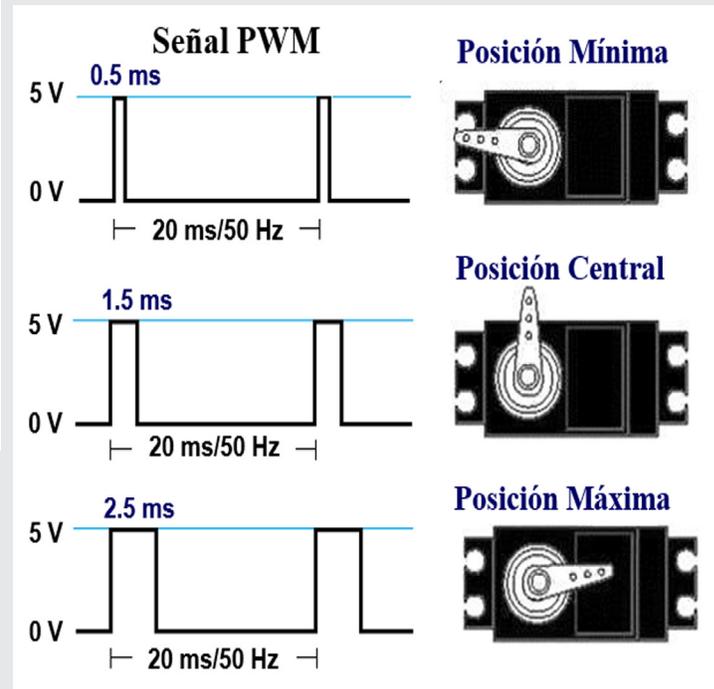


Figura 6. Esquema de la señal PWM para servo control.

Figura 6. Esquema de la señal PWM para servo control. La posición inicial de los servomotores se presenta en la Tabla 1 y es expresada en duración del ancho del pulso en micro segundos us. A partir de estos, se definen los parámetros de las variables de salida como los valores, en us, que se deben sumar o restar al ancho del pulso. De igual manera se definen 7 funciones de membresía para cada variable de salida. La Figura 7 presenta las funciones de membresía definidas para la variable lingüística servo-1.

Tabla 1. Posición Inicial de los Servomotores.

	servos					
	1	2	3	4	5	6
Duración del pulso en us	1500	1550	2200	1500	1440	1500

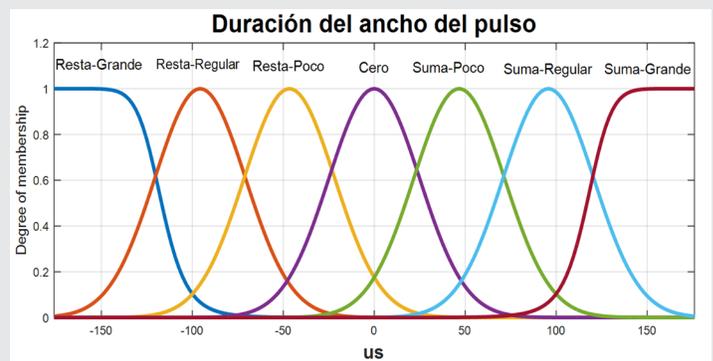


Figura 7. Funciones de membresía de la variable de salida servo-1.

Base de Conocimiento: definición de reglas difusas.

Las reglas de inferencia que conforman la base del conocimiento están definidas como reglas IF (antecedente) THEN (consecuente). Las reglas son propuestas, de tal manera que los giros de los servomotores generen un movimiento en la cámara hacia la misma dirección del desplazamiento del objeto. Como se expuso anteriormente, los movimientos del robot fueron caracterizados y la Tabla 2 presenta un resumen de esta caracterización en cuatro movimientos generales: "subir", "bajar", "izquierda" y "derecha". Para generar cada movimiento de la Tabla 2, se toman como referencia los valores del ancho de pulso de la Tabla 1, y se indica la operación que se debe aplicar sobre dichos valores: sumar y restar; y así posicionar correctamente a la cámara. De acuerdo con esto, un ejemplo de la regla de inferencia para generar un movimiento a la izquierda es: *IF error-X es Neg AND error-Y es Cero THEN servo-1 es Suma-Regular AND servo-6 es Resta-Regular.*

Tabla 2. Caracterización del movimiento de los Servomotores.

	servos					
	1	2	3	4	5	6
Subir		suma	resta	resta	suma	
Bajar		resta	suma	suma	resta	
Izquierda	suma					resta
Derecha	resta					suma

En total se definieron 49 reglas que completan la base de conocimiento del controlador. Los operadores difusos min, max-min y el método por centro de gravedad son usados como los operadores de intersección, regla composicional y método de defusificación, respectivamente.

RESULTADOS

Reconocimiento de color por RNA.

En cada entrenamiento y validación de la RNA se calcularon los valores de rendimiento MSE (Mean Squared Error) para cada algoritmo de entrenamiento, los cuales indican el error mínimo alcanzado. Los resultados promedios del MSE se presentan en la Tabla 3, en donde se indica que el algoritmo Levenberg-Marquardt proporcionó el mejor rendimiento en cuanto a entrenamiento y validación de la red neuronal. Por lo anterior, Levenberg-Marquardt es el algoritmo con el que se entrena la red implementada en este trabajo.

Tabla 3. Comparación del rendimiento de los diferentes algoritmos de entrenamiento.

Algoritmo	Entrenamiento (MSE)	Validación (MSE)
Levenberg-Marquardt	0.0403	0.00063
BFGS Quasi Newton	0.114	0.052
Gradiente Conjugado Escalado	0.0561	0.0398

Para validar el desempeño del reconocimiento de colores, se tomaron 70 imágenes que incluían los diferentes colores, donde cada color en sus diferentes tonali-

dades, se repite en 10 imágenes. La Figura 8a presenta un ejemplo de una de las imágenes capturadas. En la Figura 8b se presenta el resultado del pre-procesamiento, la 8c es el resultado de la clasificación sobre la imagen binarizada y la 8d presenta una máscara de la imagen original con la imagen obtenida para reconocer un color amarillo.

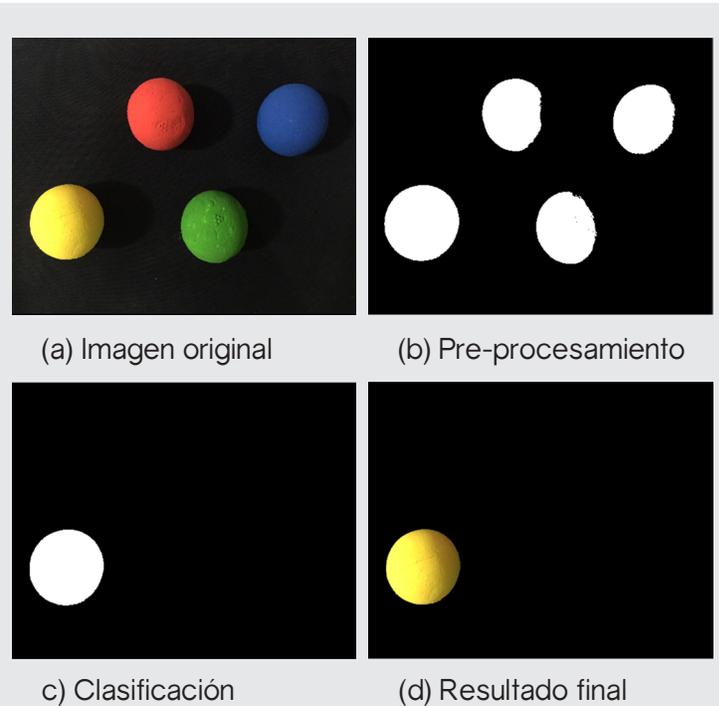


Figura 8. Resultados de la clasificación de la RNA para reconocer un color amarillo.

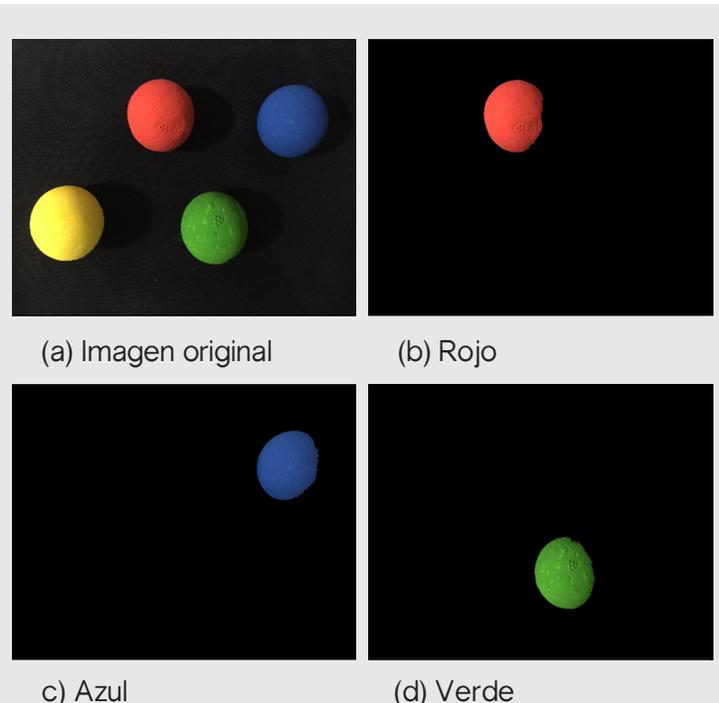


Figura 9. Resultados de la clasificación de los colores rojo, azul y verde.

La Figura 9 presenta los resultados finales del reconocimiento de los demás colores contenidos en la Figura 8a. La Tabla 4 presenta la matriz de confusión que muestra el porcentaje de precisión obtenido durante la validación. El porcentaje general de precisión es de 94.28%.

Tabla 3. Comparación del rendimiento de los diferentes algoritmos de entrenamiento.

	rojo	verde	azul	amarillo	cyan	magenta	naranja
rojo	100	0	0	0	0	0	0
verde	0	90	0	10	0	0	0
azul	0	0	100	0	0	0	0
amarillo	0	0	0	100	0	0	0
cyan	0	0	10	0	90	0	0
magenta	0	0	0	0	0	100	0
naranja	20	0	0	0	0	0	80
94.28%							

Control difuso de seguimiento autónomo.

La Figura 10 presenta la imagen real del brazo robótico y la cámara web en su posición inicial. Es importante mencionar que cuando se trabaja con video, la iluminación es un factor importante a considerar, pues si no se cuenta con buenas condiciones de iluminación los resultados de la clasificación se verán afectados. Para disminuir los efectos de las sombras se decidió realizar las pruebas sobre un fondo de color negro, tal como se muestra en la Figura 10. Para la comunicación entre la PC y el brazo robot y para el control de servos se utilizó el Maestro Servo Controller USB del fabricante Pololu [16].

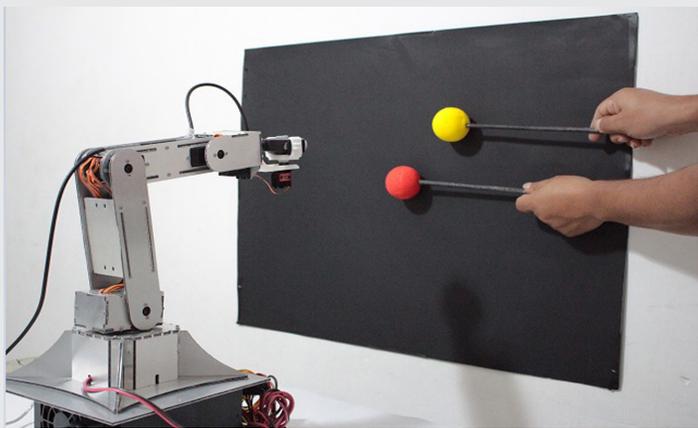
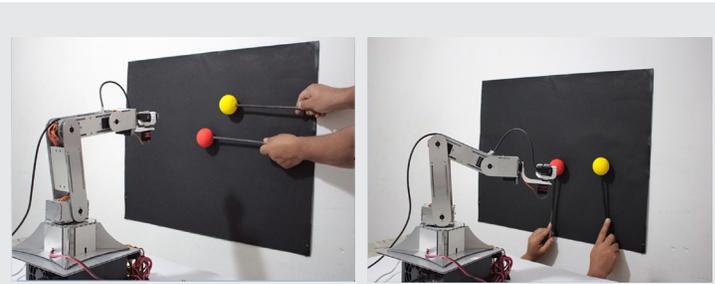


Figura 10. Posición inicial del brazo robot.

El controlador difuso entregó correctamente la posición angular de los servomotores, de tal forma que se cumple el amarre de la posición angular de la cámara con la información visual de la escena detectada. La Figura 11 presenta imágenes del sistema completo funcionando. Se considera el punto inicial el centro de la imagen y el seguimiento es para el objeto de color amarillo. Se observa que la cámara web se mantiene centrada apuntando al objetivo, en este caso al objeto de color amarillo. Esto es debido a que el movimiento de los servos del brazo compensa el desplazamiento del objetivo a través del control difuso implementado.



(a) A la derecha

(b) A la derecha y arriba



(c) A la derecha

(d) A la derecha y abajo

Figura 11. Demostración de los movimientos del brazo robot.

CONCLUSIONES

Se presentó un control de visión por computadora, el cual incorpora técnicas de inteligencia artificial: lógica difusa y redes neuronales artificiales, para ejecutar el seguimiento autónomo de la posición de un objeto de color específico. El sistema está desarrollado en Matlab® versión 8.5 y fue diseñado como una aplicación didáctica de un brazo robot de 6 GDL.

El brazo robot fue diseñado en CATIA v5 y manufacturado en las instalaciones del Instituto Tecnológico Superior de Atlixco. Los resultados obtenidos en la clasificación de color comprueban el éxito obtenido en la etapa de validación de los resultados de nuestra red neuronal, obteniendo un promedio general de 94.28%. Como se mencionó anteriormente, este porcentaje se puede mejorar cambiando las condiciones de iluminación, aunque el propósito general se logró con éxito. El controlador difuso entregó correctamente la posición angular de los servos, con ello se demostró que fue posible ejecutar el control de posición sin un modelo matemático del proceso, incorporando una base de conocimiento que emula el razonamiento humano.

Como trabajo futuro, se pretende incorporar este control en un sistema embebido con una cámara de alta velocidad y de esta forma eliminar el uso de una PC.

BIBLIOGRAFÍA

- [1]. Pasino, K. M. Pasino, Yurkovich S. (1998) *Fuzzy Control*. USA: Addison Wesley.
- [2]. Hagan M.T., Demuth H.T., Beale M.H. (1996). *Neural Networks Design*, Boston, MA: PWS Publishing.

- [3]. Lopez, M. B., Pérez, M. A. A., & Lizarralde, F. (2014). Experimental Results for a LMI-Based Fuzzy Visual Servoing Controller Applied in Robots Manipulators in 2D. *Epsilon*, (22), 151-168.
- [4]. Juang, J. G., Tsai, Y. J., & Fan, Y. W. (2015). Visual recognition and its application to robot arm control. *Applied Sciences*, 5(4), 851-880.
- [5]. Kim, H. J., Tanaka, Y., Kawamura, A., Kawamura, S., & Nishioka, Y. (2015, July). Improvement of position accuracy for inflatable robotic arm using visual feedback control method. In *Advanced Intelligent Mechatronics (AIM), 2015 IEEE International Conference on* (pp. 767-772). IEEE.
- [6]. Azar, A. T., Ammar, H. H., & Mliki, H. (2018, February). Fuzzy Logic Controller with Color Vision System Tracking for Mobile Manipulator Robot. In *International Conference on Advanced Machine Learning Technologies and Applications* (pp. 138-146). Springer, Cham.
- [7]. Meena, P. K., & Bhushan, B. (2017). Simulation for Position Control of DC Motor using Fuzzy Logic. *International Journal of Electronics, Electrical and Computational System*, (6), 188-191.
- [8]. Mirzadeh, M., Haghghi, M., Khezri, S., Mahmoodi, J., & Karbasi, H. (2014). Design Adaptive Fuzzy Inference Controller for Robot Arm. *IJ Information Technology and Computer Science*, 9, 66-73.
- [9]. Rivai, M., & Purwanto, D. (2015, May). Implementation of fuzzy logic control in robot arm for searching location of gas leak. In *Intelligent Technology and Its Applications (ISITIA), 2015 International Seminar on IEEE* (pp. 69-74).
- [10]. Wang, M. S. (2018). Eye to hand calibration using ANFIS for stereo vision-based object manipulation system. *Microsystem Technologies*, 24(1), 305-317.
- [11]. Ligutan, D. D., Cruz, L. J. S., Del Rosario, M. C. D., Kudhal, J. N. S., Abad, A. C., & Dadios, E. P. (2017). Design and implementation of a fuzzy logic-based joint controller on a 6-DOF robot arm with machine vision feedback. In *Computing Conference IEEE 2017*, (pp. 249-257).
- [12]. Zapata S. (2016). Object Tracking with a 2 DOF Robot Arm Using Fuzzy Logic. Consultado el 10 de abril de 2018, Mathworks: <https://la.mathworks.com/matlabcentral/fileexchange/58205-object-tracking-with-a-2-dof-robot-arm-using-fuzzy-logic--matlabhw2k16>.
- [13]. Demuth, H., & Beale, M. (1993). *Neural Network Toolbox For Use with Matlab--User'S Guide Version 3.0*.
- [14]. Sivanandam S.N., Sumathi S., Deepa S.N. (2007). *Introduction to fuzzy logic using MATLAB*. Springer-Verlag.
- [15]. Pinckney, N. (2006). Pulse-width modulation for microcontroller servo control. *IEEE potentials*, 25(1), 27-29.
- [16]. Pololu Corporation: *Maestro Servo Controller. User's Guide* (n.d.) Obtenida de la página electrónica: <https://www.pololu.com/docs/pdf/0J40/maestro.pdf>

Agradecimientos

Los autores agradecen el apoyo financiero del Tecnológico Nacional de México para la construcción del brazo robótico 6 GDL, obtenido del proyecto "Módulo didáctico de brazos robots para el equipamiento del laboratorio de robótica" bajo el programa de Apoyo a la Investigación Científica y Desarrollo Tecnológico de los programas educativos de los Institutos Tecnológicos Descentralizados.