

Controlador difuso-evolutivo para robot seguidor de luz

Carlos Vargas Arvizu, Angel Rodríguez-Liñán, Luis M. Torres-Treviño
Universidad Autónoma de Nuevo León, FIME, Posgrado en Ingeniería Eléctrica
ing.arvizu.imtc@gmail.com, angel.rodriguezln@uanl.edu.mx,
luis.torres.ciidit@gmail.com

RESUMEN

En este trabajo, se propone la implementación de un controlador por lógica difusa optimizando sus reglas mediante un algoritmo genético. Este controlador comanda a un sistema robótico que consiste en un mecanismo diseñado para realizar seguimiento de una fuente lumínica, aún si el entorno presenta perturbaciones lumínicas. El uso de los sistemas de lógica difusa nos permite tener buenas aproximaciones e interpretaciones del conocimiento que alcanza el algoritmo genético, además presenta robustez y buena precisión.

PALABRAS CLAVE

Robótica evolutiva, Lógica difusa, Algoritmo genético, Sistema difuso-evolutivo, Control de robots.

ABSTRACT

In this work, it is proposed an implementation of a fuzzy logic rule-based controller with rules optimized by a genetic algorithm. This controller leads a robotic system that consists in a linkage designed to perform light tracking, even if the environment changes due to the light disturbances. The fuzzy-evolutionary system allows finding good approximations for the control task due to the evolutionary behavior, also has robustness and relative precision.

KEYWORDS

Evolutionary robotics, Fuzzy logic, Genetic algorithm, Fuzzy-evolutionary system, Robot control.

INTRODUCCIÓN

La Robótica Evolutiva (RE) es un paradigma derivado de aplicar los principios de la Teoría de la Evolución biológica, en forma de técnicas computacionales para robots no pre-programados; y con el propósito de que los robots sean capaces de evolucionar, hasta alcanzar la mejor solución a ciertos problemas complejos.¹ Este paradigma se ha investigado recientemente por un creciente número de investigadores en los diferentes aspectos de la robótica² (como en diseño estructural, diseño de controladores, coevolución ‘cerebro-estructura’ del robot, percepción), así como el reto de la implementación física de dicho paradigma. Este trabajo se enfoca particularmente en la aplicación de controladores evolutivos para robots.

ANTECEDENTES

Algunas de las técnicas de control evolutivo, reportadas en la literatura, incluyen lógica difusa y optimización por medio de algoritmos evolutivos (AE) para mejorar las reglas de sistemas difusos. En el trabajo de (Cervantes-Castillo, 2013)³ se evoluciona a las funciones de membresía de un sistema de lógica difusa para optimizarlas. Otra estrategia de optimización evolutiva es modificar los valores de las variables de las reglas de un sistema de lógica difusa hasta llegar a la solución óptima.^{4,5} Un sistema difuso-evolutivo basado en reglas reemplaza fácilmente a un sistema difuso respetando las entradas y salidas requeridas para su correcto funcionamiento, también presenta la misma ventaja de un sistema puramente difuso de describir las entradas y salidas en forma lingüística basándose en aproximaciones cuantitativas, y adicionalmente, mejora la robustez y adaptabilidad de los métodos de control clásico.^{3,6,7} A menudo, en el sistema robótico evolutivo se implementa una calibración inicial para proporcionar una mayor precisión en las mediciones de los sensores.⁸

Contribución

En este trabajo, se desarrolla un prototipo de robot seguidor de luz con un controlador difuso-evolutivo basado en reglas en la plataforma *Arduino Due*[®], el cual no requiere el conocimiento previo de una persona experta acerca de la dinámica del sistema, además se plantean experimentos de seguimiento de luz en cuatro escenarios de prueba, incluyendo perturbaciones y variaciones lumínicas, lo que ilustra la adaptabilidad del sistema evolutivo.

En las siguientes secciones se explica el controlador propuesto, basado en lógica difusa y un algoritmo genético, que optimiza las reglas difusas; así como el hardware del prototipo robótico diseñado, y posteriormente las discusiones de los experimentos en los escenarios propuestos.

CONTROLADOR DIFUSO-EVOLUTIVO BASADO EN REGLAS

Sistemas de lógica difusa

La lógica difusa se presenta como un paradigma de sistemas inteligentes, que permite modelar un sistema o proceso a través de aproximaciones conocidas como términos lingüísticos que definen a conjuntos y reglas difusas. En un modelo de lógica difusa las entradas, salidas y valores involucrados son diferentes a los habitualmente utilizados en la lógica Booleana ‘clásica’, que consisten en ‘1’ o ‘0’ (VERDADERO o FALSO, respectivamente); los valores de lógica difusa consisten en grados de membresía o pertenencia al conjunto difuso. Una ventaja de la representación difusa es que permite trasladar de manera más natural el conocimiento de un individuo experto en valores numéricos aproximados de variables de un sistema, de tal forma que se obtiene un modelo relativamente funcional y sencillo a pesar del uso de información imprecisa. Un sistema de lógica difusa basado en reglas está conformado por:

1. *Reglas*: Las reglas son relaciones entre variables de entradas y salidas que modelan los aspectos cualitativos de un sistema sin requerir un análisis cuantitativo de precisión. Estas reglas son de la forma ‘SI-ENTONCES’ (IF-THEN en inglés), que abstraen los conocimientos de humanos expertos acerca

de la interpretación de datos numéricos del proceso que se desea modelar, mediante términos lingüísticos. Algunos ejemplos son:

1) “SI temperatura_A ES caliente y temperatura_B ES caliente ENTONCES velocidad_abanico ES rápida”,

2) “SI temperatura_A ES fría y temperatura_B ES caliente ENTONCES velocidad_abanico ES moderada”,

donde frío, caliente, moderada y rápida son los términos lingüísticos de las reglas. Cabe señalar que a cada valor cuantitativo de las variables del proceso le corresponden valores difusos de cada valor lingüístico (como caliente y frío) asignados por las llamadas ‘funciones de membresía’. Una función de membresía se define como una relación que determina el grado de pertenencia de un valor cuantitativo de las variables del sistema en un conjunto difuso, dado para cada término lingüístico, como se ilustra en la figura 1. En la literatura hay diversos trabajos sobre la optimización de los parámetros de funciones de membresía.^{7,9}

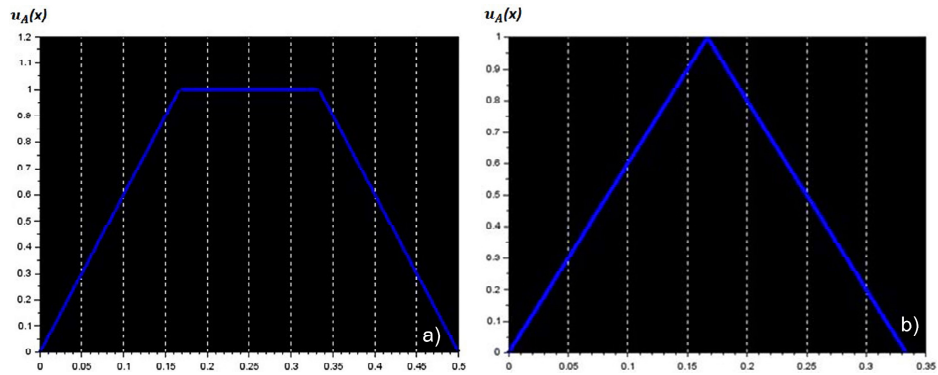


Fig. 1. Funciones de membresía.

2. *Fusificación*: Es el proceso que consiste en transformar los valores cuantitativos de las variables de entrada en valores difusos asociados a los conjuntos difusos, mediante las funciones de membresía; para poder ser interpretados en las reglas del sistema difuso. Éstas se distribuyen en un intervalo normalizado entre 0 y 1, como se muestra en la figura 2.

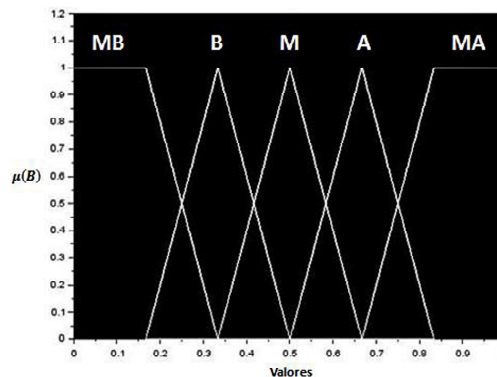


Fig. 2. Funciones de membresía uniformemente distribuidas para cada conjunto difuso.

3. *Mecanismo de Inferencia*: Es el proceso que determina una salida difusa para cada una de las reglas, es decir, valores difusos de un conjunto difuso son mapeados a otro conjunto difuso de salida. Además, este mecanismo determina la activación de las reglas y la forma en cómo éstas se combinan. Funge como el ‘cerebro’ del sistema difuso, haciendo las inferencias para determinar la salida difusa del sistema.

4. *Defusificación*: Es el proceso que consiste en tomar el valor difuso de la salida, generado en el mecanismo de inferencia, y convertirlo en un valor cuantitativo de la salida.

Algoritmo genético

Un algoritmo genético (**AG**) es una técnica de búsqueda iterativa basada en los principios biológicos de la selección natural. Con esta técnica, la evolución biológica de las especies se modela como un algoritmo que inicia con una ‘población’ de ‘individuos’ con ‘cromosomas’ representados por vectores, cuyas n componentes son llamadas ‘genes’ o ‘alelos’. La capacidad de ‘supervivencia’ de cada individuo de la población se evalúa mediante una función de aptitud; luego de ser evaluados, se seleccionan los individuos con mejor aptitud de la población y se reproducen para generar nuevos individuos con cromosomas similares. En las nuevas poblaciones generadas, se ejecutan los mismos procesos de evaluación y reproducción de los individuos, iterativamente. Después de N generaciones, los individuos de la población tendrán los cromosomas que mejor cumplan la función de aptitud dada. Por lo cual, estos algoritmos son utilizados para resolver problemas de optimización de valores y parámetros en procesos y sistemas, donde se busca maximizar o minimizar una función aptitud, identificando los individuos (valores de parámetros) óptimos.^{10,11}

En general, los algoritmos de búsqueda son propensos a estancarse en óptimos locales, a lo cual se denomina convergencia prematura; sin embargo, los **AG** tienen mecanismos de selección, cruce y mutación, que los hacen más eficaces para prevenir la convergencia prematura y mejorar la búsqueda, para encontrar el óptimo global. Otra ventaja de los **AG** es que pueden resolver problemas de un alto grado de complejidad y altamente dimensionales, a diferencia de los métodos de optimización tradicionales analíticos.

En la figura 3 se representa la analogía formulada a partir de los cromosomas biológicos, para generar un cromosoma artificial con genes en un algoritmo genético. Las etapas del algoritmo genético de genes enteros son:

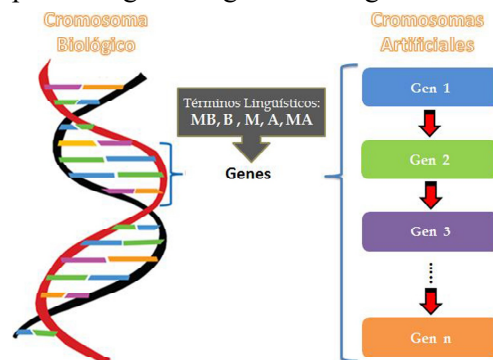


Fig. 3. Cromosomas en algoritmos genéticos.

1. *Inicialización*: Consiste en la generación de la población inicial de individuos y cromosomas que se proponen para el problema de optimización. En el caso de sistemas de lógica difusa, los términos lingüísticos se encuentran en un rango de valores, por lo tanto, para generar cada valor de los genes de los cromosomas iniciales, se asignan valores aleatorios dentro del rango de los términos lingüísticos. La población inicial está dada por

$$P_{inicial} = \begin{bmatrix} g_{11} & g_{12} & \cdots & g_{1n} \\ g_{21} & g_{22} & \cdots & g_{2n} \\ \vdots & \vdots & \vdots & \vdots \\ g_{m1} & g_{m2} & \cdots & g_{mn} \end{bmatrix}, \quad (3)$$

donde g_{ij} , con $j=1,2,\dots, m$, son los m genes producidos aleatoriamente, para el i -ésimo cromosoma del individuo, con $i=1,2,\dots, n$.

2. *Evaluación y selección*: Consiste en evaluar cada uno de los elementos de la población mediante la función de aptitud, y asignar una puntuación en un intervalo normalizado. Dependiendo del criterio de evaluación (maximizar o minimizar) la función de aptitud es:

$$f_{fit}(n) = \begin{cases} norm(f_{(e)}), & \text{maximización} \\ norm(1 - f_{(e)}), & \text{minimización} \end{cases} \quad (4)$$

para n número de cromosomas. Además, se considera la puntuación obtenida por cada individuo, tomando los de mejor valor de aptitud.

3. *Reproducción*: A partir de los individuos seleccionados con mejor aptitud, se genera una nueva población que contiene individuos con cruce de cromosomas de los antepasados más aptos, por lo tanto se espera sean una mejor población.

4. *Mutación*: La mutación es otro mecanismo de evolución de cromosomas, que ayuda a evitar la convergencia prematura, y crea un mejor espacio de soluciones para la búsqueda del óptimo global. La mutación se ve afectada por un parámetro de probabilidad de mutación que establece la probabilidad de producir un cambio en cada uno de los genes de cualquier cromosoma. El valor de la probabilidad de mutación se selecciona normalmente en el intervalo [0.01, 0.3]. La mutación implica la alteración de algunos genes en el cromosoma.

5. *Elitismo*: Para mejorar el rendimiento del algoritmo genético se introduce el elitismo, el cual es responsable de tomar al individuo más apto de la generación anterior e insertarlo nuevamente en la población para garantizar que el individuo con los mejores genes esté presente en la población de cada generación. Esto ayuda al algoritmo genético a converger a la solución óptima en menos tiempo.

En este trabajo, el algoritmo genético tiene como objetivo mejorar el sistema de lógica difusa. En la figura 4 se muestra su diagrama de flujo. Los parámetros que se han seleccionado para el algoritmo genético están en la tabla I.

Tabla I. Parámetros de algoritmo genético que optimiza al sistema de lógica difusa.

Número de Generaciones	Número de Poblaciones	Número de Genes del Cromosoma	P_c	M_c
10	10	25	0.8	0.1

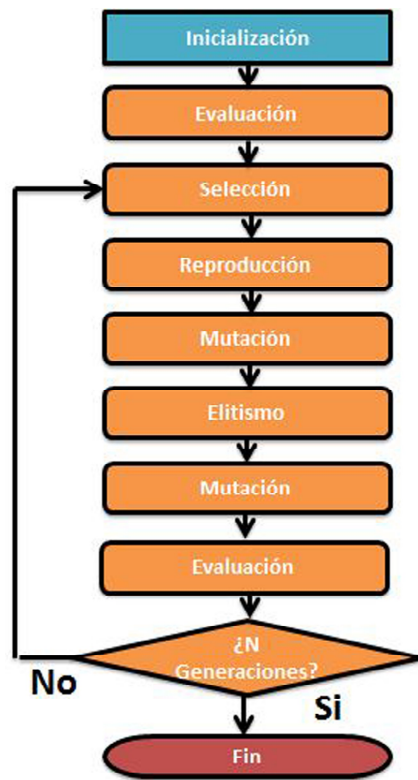


Fig. 4. Diagrama de flujo del algoritmo genético.

En la literatura se ha reportado el desarrollo de métodos innovadores para minimizar las desventajas de los algoritmos genéticos simples,¹² entre los cuales se han propuesto nuevos esquemas de algoritmos evolutivos, o también, manteniendo la estructura simple del algoritmo pero variando algunos mecanismos importantes, por ejemplo, la mutación.¹³

Esquema propuesto de controlador difuso-evolutivo basado en reglas

En este trabajo se propone un controlador difuso-evolutivo, que está compuesto por un sistema de lógica difusa de tipo I y un algoritmo genético con cromosomas de genes enteros. En el sistema de lógica difusa tipo I, se utiliza el método de defusificación por alturas, cuya expresión de cálculo es

$$y_h = \frac{\sum_{l=1}^M y^{-l} \mu_{B^k}(y^{-l})}{\sum_{l=1}^M \mu_{B^k}(y^{-l})}, \quad (1)$$

donde y^{-l} es el centro de gravedad del conjunto difuso B^k asociado con la activación de la regla R^l y con M reglas. Este defusificador es fácil de implementar debido a que, comúnmente, el centro de gravedad de las funciones de pertenencia se conoce previamente. Independientemente si se utiliza la inferencia del producto o mínimo, el centro de gravedad coincide siempre con el centro de la forma geométrica de la función de pertenencia simétrica. También se hace uso de las

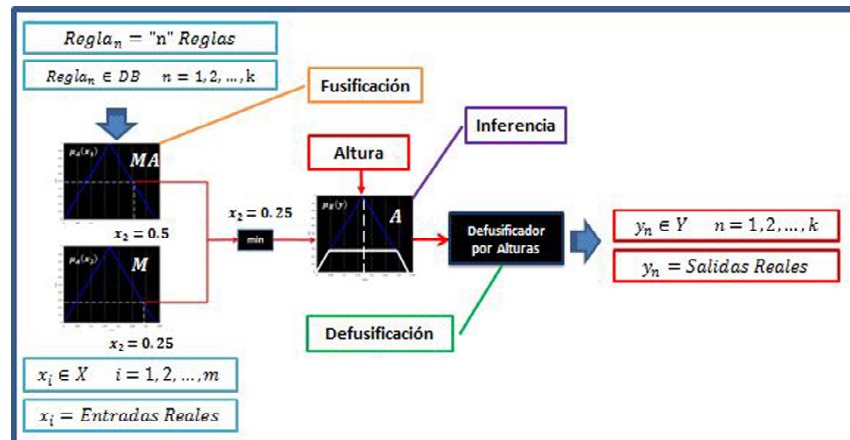


Fig. 5. Esquema del sistema difuso.

funciones de membresía trapezoidal y triangular, que se ilustran en la figura 1a y la figura 1b, respectivamente. En la figura 2 se ilustra la distribución de las funciones de pertenencia de los conjuntos difusos. La figura 5 representa un esquema del sistema difuso considerado en este trabajo.

Los términos lingüísticos usados para las aproximaciones son:

- MB = Término lingüístico de Muy Bajo que corresponde al valor cuantitativo de 1.
- B = Término lingüístico de Bajo que corresponde al valor cuantitativo de 2.
- M = Término lingüístico de Medio que corresponde al valor cuantitativo de 3.
- A = Término lingüístico de Alto que corresponde al valor cuantitativo de 4.
- MA = Término lingüístico de Muy Alto que corresponde al valor cuantitativo de 5.

El total de reglas para el sistema gobernado por lógica difusa se determina por la relación

$$R^l = (B^k)^{x_i}, \quad (2)$$

donde R^l es el número de reglas, B^k los conjuntos difusos asociados a los términos lingüísticos establecidos y x_i es el número de entradas.

Aunque los sistemas difusos son modelos suficientemente fáciles de interpretar, al utilizar un controlador difuso-evolutivo, donde un algoritmo evolutivo se encarga de la optimización, la interpretación del resultado obtenido puede no ser del todo comprensible, por ello, hay estudios que tratan de hacer más comprensible la extracción del conocimiento de los resultados obtenidos por estos algoritmos.⁵

El algoritmo genético implementado optimiza la siguiente función de aptitud del controlador difuso-evolutivo:

$$f(e) = |X_{an1} - X_{an2}|, \quad (5)$$

donde X_{ami} (para $i = \{1, 2\}$) son los valores de los voltajes analógicos medidos por el Arduino DUE®.

El algoritmo genético considera 10 cromosomas, donde cada cromosoma está compuesto por 25 genes que corresponden al número de reglas del sistema de lógica difusa, y estas reglas son los parámetros a optimizar, similar a como se propuso en el trabajo de Wei-Wang.¹⁴ Además, se selecciona y guarda el individuo con el mejor valor de aptitud (elitismo). Para la optimización de las reglas del sistema de lógica difusa se considera la minimización del error generado por la diferencia de los voltajes medidos en sensores fotoresistivos (LDR por sus siglas en inglés). Para la etapa de selección, se asigna el mecanismo de selección estocástica, el cual asigna un valor de probabilidad a cada uno de los elementos, de modo que la selección es aleatoria y emula de mejor manera los principios de la selección natural. La selección estocástica se formula de la siguiente manera:

$$S_E(n) = \sigma_{fit}^n(n), \tag{6}$$

para n número de cromosomas y donde σ es la probabilidad aleatoria asignada en cada evaluación con un valor en el intervalo $[0,1]$. Para la reproducción, se asigna el mecanismo de cruce uniforme, ilustrado en la figura 6, el cual implica la generación de una máscara binaria aleatoria que define la condición del intercambio de genes, y que establece que al haber '1's en la máscara, los genes del primer padre se intercambia con el primer hijo y los genes del segundo padre se intercambia con el segundo hijo, en cambio, cuando hay '0's, el intercambio de los genes se invierte, es decir, el primer hijo toma genes del segundo padre y el segundo hijo toma genes del primer padre, de tal modo que hay un intercambio de genes de mayor aleatoriedad, además, la selección de los padres en el cruce es aleatoria y el cruce se lleva a cabo con diferentes individuos (cromosomas), es decir, sin cruzarse consigo mismos. La probabilidad de cruce P_c establece la probabilidad de que ocurra la reproducción, y se escoge 0.8, del intervalo $[0,1]$. La probabilidad de mutación establece que los genes pueden mutar en valores que se encuentran en el intervalo $[1,5]$, es decir, los posibles valores de las reglas del sistema de lógica difusa; dicha probabilidad se elige como 0.1 del intervalo $[0,1]$.

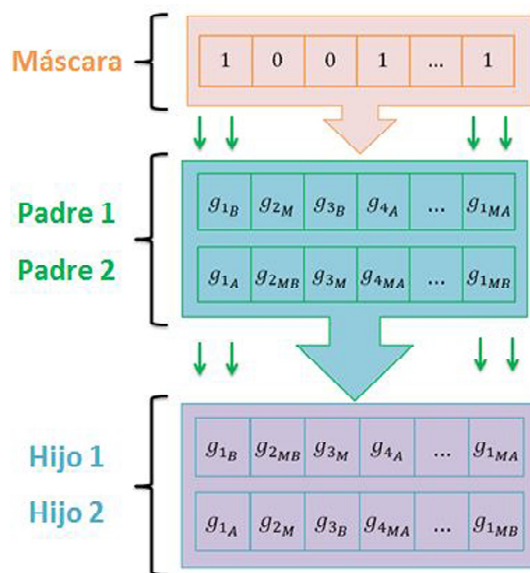


Fig. 6. Cruce uniforme para la reproducción de los cromosomas.

IMPLEMENTACIÓN DEL CONTROLADOR DIFUSO-EVOLUTIVO EN EL SISTEMA ROBÓTICO

El sistema robótico evolutivo completo consta del controlador difuso-evolutivo y de un prototipo robótico. Un diagrama de bloques del sistema robótico evolutivo se muestra en la figura 7.

El hardware del sistema robótico evolutivo se ilustra en la figura 8 y se describe a continuación:

- **Microcontrolador:** La tarjeta electrónica Arduino Due® contiene el código computacional del controlador difuso-evolutivo. Las características más importantes para la selección de esta tarjeta son principalmente, la capacidad de procesamiento y el espacio de memoria disponible para almacenamiento de código y datos.
- **Servomotores:** El robot tiene dos servomotores, un servomotor controlado por la salida del sistema de lógica difusa y un servomotor de la fuente lumínica para la posición de la misma en las etapas de calibración, evaluación y el modo de pruebas del sistema.
- **LED de Potencia:** Su objetivo es proporcionar la fuente lumínica que sigue el robot. Se puede ajustar la cantidad de luz proporcionada y el encendido/apagado.

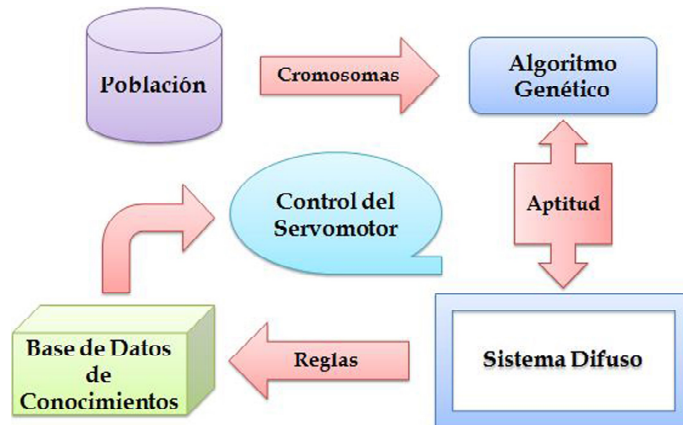


Fig. 7. Diagrama de bloques del sistema robótico evolutivo.

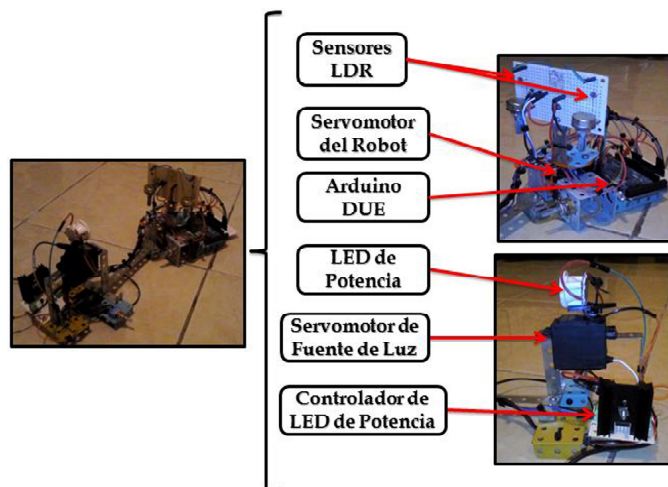


Fig. 8. Hardware del sistema robótico evolutivo.

- Sensores LDR: Se utilizan sensores fotosensibles en el robot para medir la cantidad de luz de su entorno. Los valores de los sensores son las entradas al sistema de lógica difusa. Estos componentes son sensores analógicos de bajo costo, y además, tiene potenciómetros para calibrar la sensibilidad con la que los sensores detectan las variaciones de la intensidad lumínica.

RESULTADOS

Las pruebas experimentales para el controlador difuso-evolutivo con el sistema robótico ilustran la capacidad de optimización en cuatro casos de estudio; en los que el sistema alcanza la mejor solución en 10 generaciones de evaluación. Se observa su evolución y adaptación en las distintas condiciones de luz del entorno.

Se establece que el primer y tercer caso de prueba, consideran que no hay perturbaciones de luz en el entorno, es decir, la diferencia entre los valores medidos por los dos sensores sea aproximadamente cero cuando no está presente la fuente de luz a seguir. En el segundo y cuarto caso, se consideran perturbaciones lumínicas significativas en el entorno, de tal modo que al obtener las mediciones de los sensores se ven afectadas debido a la cantidad de luz presente.

Tabla II. Reglas difusas de las pruebas I y II.

Reglas(R^i)	Entrada (x_1)	Entrada (x_2)
R^1	Muy Bajo	Muy Bajo
R^2	Muy Bajo	Bajo
R^3	Muy Bajo	Medio
R^4	Muy Bajo	Alto
R^5	Muy Bajo	Muy Alto
R^6	Bajo	Muy Bajo
R^7	Bajo	Bajo
R^8	Bajo	Medio
R^9	Bajo	Alto
R^{10}	Bajo	Muy Alto
R^{11}	Medio	Muy Bajo
R^{12}	Medio	Bajo
R^{13}	Medio	Medio
R^{14}	Medio	Alto
R^{15}	Medio	Muy Alto
R^{16}	Alto	Muy Bajo
R^{17}	Alto	Bajo
R^{18}	Alto	Medio
R^{19}	Alto	Alto
R^{20}	Alto	Muy Alto
R^{21}	Muy Alto	Muy Bajo
R^{22}	Muy Alto	Bajo
R^{23}	Muy Alto	Medio
R^{24}	Muy Alto	Alto
R^{25}	Muy Alto	Muy Alto

Prueba I

En esta prueba, no hay perturbaciones de luz en el entorno. Se considera una población inicial aleatoria que tiene diez cromosomas generados aleatoriamente. Las reglas optimizadas por el algoritmo genético para este caso están en la tabla II y los genes del individuo más apto encontrado se muestran en la tabla III. Los resultados obtenidos a partir de los valores de aptitud para cada uno de los individuos de cada generación se muestran en la figura 9. En la figura 10 se ilustra el error de los sensores con respecto al desplazamiento de la fuente de luz.

Tabla III. Genes para el mejor individuo encontrado en la prueba I.

g^1	g^2	g^3	g^4	g^5	g^6	g^7	g^8	g^9	g^{10}	g^{11}	g^{12}	g^{13}	g^{14}	g^{15}	g^{16}	g^{17}	g^{18}	g^{19}	g^{20}	g^{21}	g^{22}	g^{23}	g^{24}	g^{25}
3	1	2	5	4	5	2	5	4	2	4	1	1	2	3	4	2	5	2	2	3	2	1	5	3

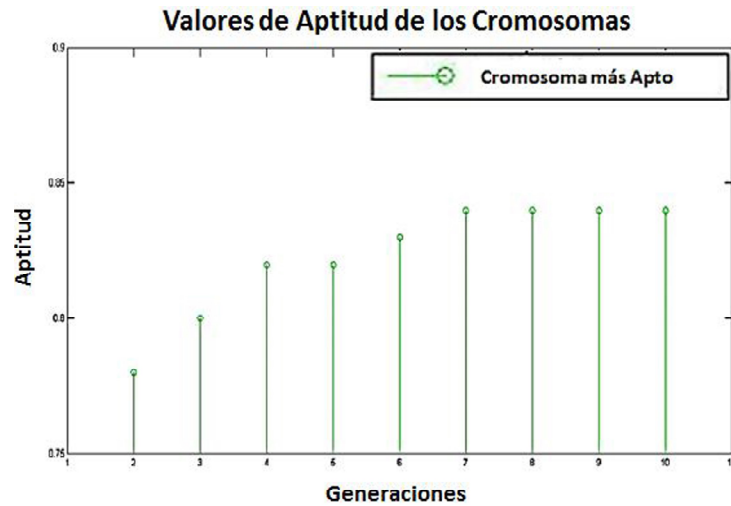


Fig. 9. Valor de aptitud de cada individuo más apto en cada generación evaluada de la prueba I.

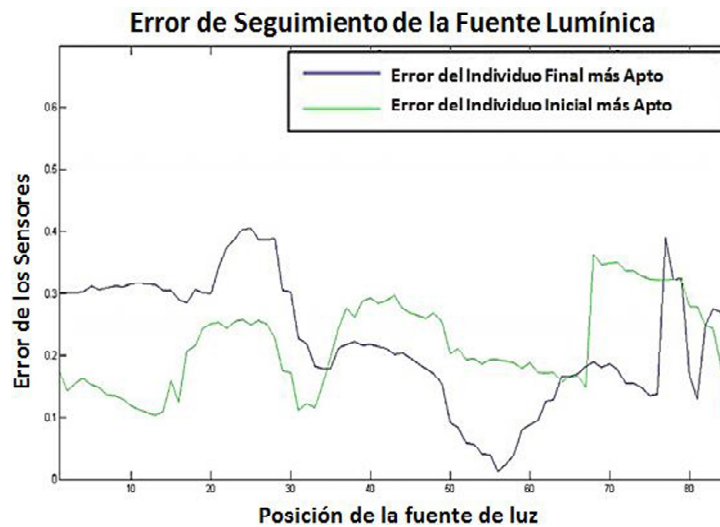


Fig. 10. Error promedio de diez muestras tomadas para cada grado de desplazamiento del servomotor desde 40° hasta 125° del individuo más apto encontrado en la prueba I.

Prueba II

Se considera que la cantidad de luz en el entorno es una perturbación. Se obtiene información para determinar cómo afecta la luz ambiental en comparación con el caso anterior, donde no existe tal perturbación. La población inicial es generada aleatoriamente y sus reglas de optimización son descritas en la tabla II y los genes del individuo más apto encontrado se muestran en la tabla IV. La figura 11 muestra el valor de aptitud del individuo más apto y en la figura 12 se muestra el error de los sensores con respecto al desplazamiento de la fuente de luz.

Tabla IV. Genes del mejor individuo encontrado en la prueba II.

g^1	g^2	g^3	g^4	g^5	g^6	g^7	g^8	g^9	g^{10}	g^{11}	g^{12}	g^{13}	g^{14}	g^{15}	g^{16}	g^{17}	g^{18}	g^{19}	g^{20}	g^{21}	g^{22}	g^{23}	g^{24}	g^{25}
3	1	2	3	2	2	2	4	1	5	2	3	2	4	2	4	5	4	5	5	2	3	5	1	4

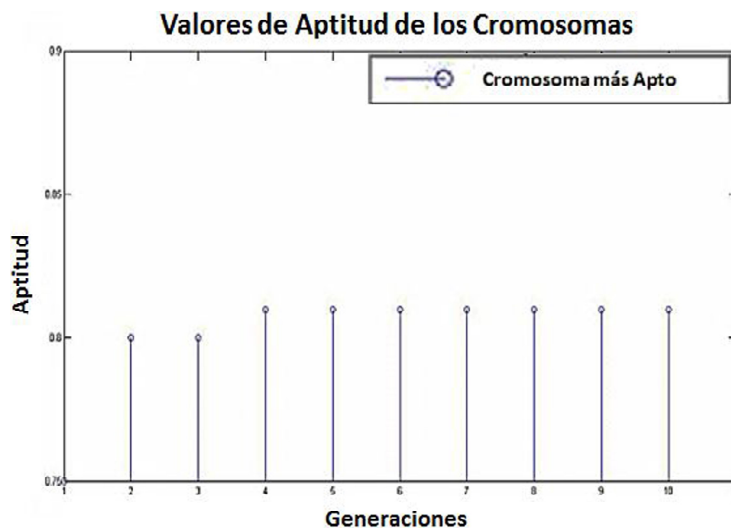


Fig. 11. Valor de aptitud de cada individuo más apto en cada generación evaluada de la prueba II.

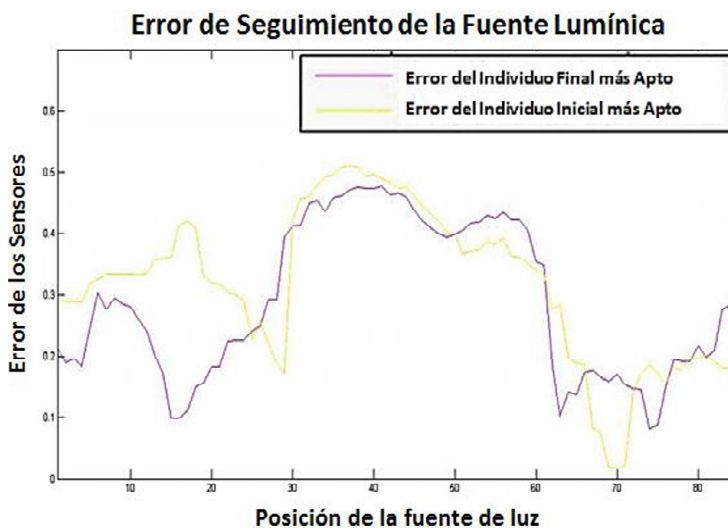


Fig. 12. Error promedio de diez muestras tomadas para cada grado de desplazamiento del servomotor desde 40° hasta los 125° para el individuo más apto encontrado en la prueba II.

Prueba III

Caso en donde se siembra una solución (conjunto de reglas), para determinar si el control difuso-evolutivo es capaz de encontrar una mejor solución a partir de una solución derivada del conocimiento de un experto. En esta prueba se establecen las mismas consideraciones que en la prueba I. Se toma una población aleatoria de nueve cromosomas en la población inicial y se siembra un cromosoma que proviene del conocimiento experto. Se espera que la convergencia a la solución óptima sea más rápida en comparación con las pruebas anteriores. En esta prueba, el sistema de lógica difusa basado en reglas es capaz de dar prioridad a una sola entrada, así que, cuando alguna de las reglas optimizadas sea igual a cero, la prioridad recae en la variable lingüística diferente de cero para cada una de las reglas que tienen valores nulos en algunas de sus entradas. Las reglas se muestran en la tabla V y los genes del individuo más apto encontrado se muestran en la tabla VI. La figura 13 muestra el valor de la aptitud del individuo más apto y en la figura 14 se muestra el error o diferencia de voltajes de los sensores con respecto al desplazamiento de la fuente de luz.

Tabla V. Reglas difusas de las pruebas III y IV.

Reglas(R^i)	Entrada (x_1)	Entrada (x_2)
R^1	Muy Bajo	0
R^2	Muy Bajo	Muy Alto
R^3	Bajo	Muy Alto
R^4	Bajo	Alto
R^5	Muy Bajo	Alto
R^6	Alto	Muy Alto
R^7	Medio	Muy Alto
R^8	Bajo	Medio
R^9	Muy Bajo	Medio
R^{10}	Medio	Medio
R^{11}	Bajo	Bajo
R^{12}	Alto	Alto
R^{13}	Muy Alto	Muy Alto
R^{14}	Muy Bajo	Muy Bajo
R^{15}	Muy Alto	Alto
R^{16}	Muy Alto	Medio
R^{17}	Muy Alto	Bajo
R^{18}	Alto	Medio
R^{19}	Alto	Bajo
R^{20}	Muy Alto	Muy Alto
R^{21}	Alto	Muy Bajo
R^{22}	Medio	Bajo
R^{23}	0	Muy Bajo
R^{24}	Muy Alto	Bajo
R^{25}	Muy Alto	Muy Bajo

Tabla VI. Genes del individuo más apto encontrado en la prueba III.

g^1	g^2	g^3	g^4	g^5	g^6	g^7	g^8	g^9	g^{10}	g^{11}	g^{12}	g^{13}	g^{14}	g^{15}	g^{16}	g^{17}	g^{18}	g^{19}	g^{20}	g^{21}	g^{22}	g^{23}	g^{24}	g^{25}	
1	1	1	2	2	2	2	2	2	3	3	2	3	3	4	4	4	4	4	4	5	4	2	5	5	5

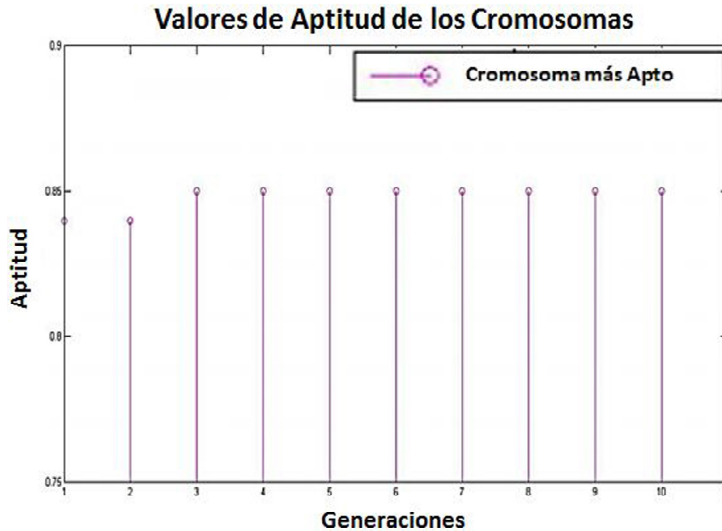


Fig. 13. Valor de aptitud de cada individuo más apto en cada generación evaluada de la prueba III.

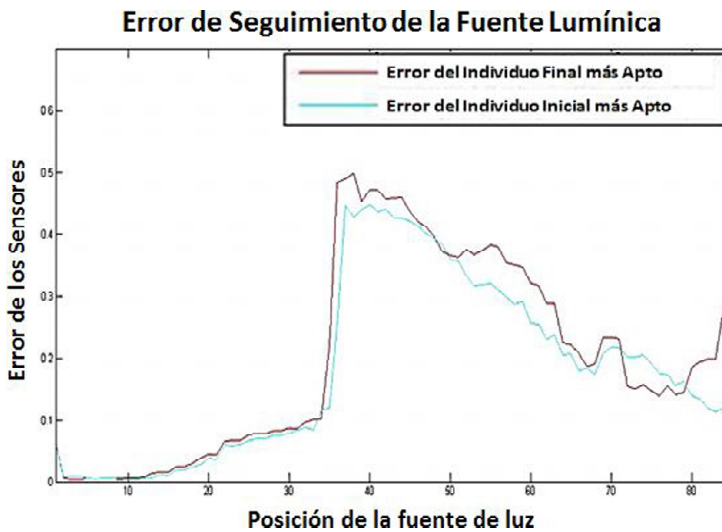


Fig. 14. Error promedio de diez muestras tomadas para cada grado de desplazamiento del servomotor desde 40° hasta 125° para el individuo más apto de la prueba III.

Tabla VII. Los mejores genes encontrados para el mejor individuo de la prueba IV.

g^1	g^2	g^3	g^4	g^5	g^6	g^7	g^8	g^9	g^{10}	g^{11}	g^{12}	g^{13}	g^{14}	g^{15}	g^{16}	g^{17}	g^{18}	g^{19}	g^{20}	g^{21}	g^{22}	g^{23}	g^{24}	g^{25}
1	2	1	5	2	2	4	4	2	4	1	3	1	3	1	5	5	4	5	3	5	4	5	5	5

Prueba IV

En este caso se toman las condiciones de luz ambiental del entorno de la prueba II, se realiza el sembrado de un cromosoma, como en la prueba III y se observa cómo se ve afectada la optimización evolutiva, posteriormente, se

analiza la información del desempeño del sistema robótico en la búsqueda del mejor comportamiento, que se explica en la sección de discusiones. Las reglas se muestran en la tabla V y los genes del individuo más apto encontrado se muestran en la tabla VII. La figura 15 muestra el valor de la aptitud del individuo más apto encontrado y en la figura 16 se muestra el error o la diferencia de voltajes de los sensores fotosensibles que se produce debido al desplazamiento de la fuente de luz.

Discusión de resultados

En todas las pruebas se ilustraron los valores de aptitud que fueron alcanzados en 10 generaciones, y además, el error promedio de diez muestras en cada grado del servomotor que comprende un desplazamiento desde los 40° hasta los 125°,

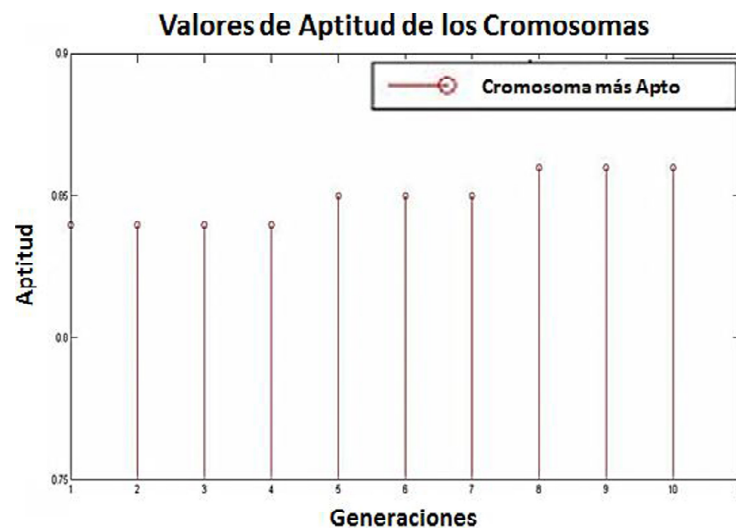


Fig. 15. Valor de aptitud de cada individuo más apto en cada generación evaluada de la prueba IV.

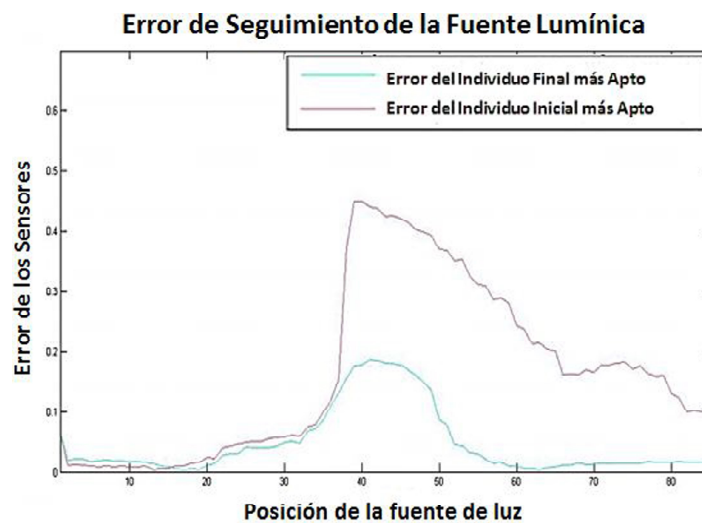


Fig. 16. Error promedio de diez muestras tomadas para cada grado de desplazamiento del servomotor desde 40° hasta los 125° que se obtuvieron a partir del individuo más apto de la prueba IV.

es decir, se tomaron 85 muestras, que son el número de posiciones donde se detecta la luz, así como también, corresponde a la cantidad de evaluaciones del error. En la figura 9 se ilustran los valores de aptitud en donde se toma en cuenta que las condiciones de luz del entorno de prueba no suponen una perturbación significativa para el sistema robótico evolutivo y el error es mínimo, como se ilustra en la figura 10, para cada posición del mejor individuo inicial y el mejor individuo final, donde también se observa cómo se ve afectado el primer individuo más apto después de la optimización del sistema hecha por el algoritmo genético. Se logra mejorar el desempeño del sistema, ya que el error del individuo final fue menor que el error generado por el individuo inicial.

En la prueba II se ilustran los individuos más aptos de cada generación en la figura 11, se introdujeron perturbaciones en la luz ambiental del entorno, sin embargo, no resulta ser una limitante, ya que en la última generación se logra el objetivo de optimización, y se observa el desempeño de los individuos inicial y final en la figura 12. Dado que la diferencia del error entre el individuo inicial y el individuo final es muy pequeña, se considera que la luz ambiental es un factor que impidió tener una optimización.

Tabla VIII. Los valores de la aptitud de los individuos más aptos de cada prueba.

Individuos más aptos	Valores de aptitud
Prueba I	0.84
Prueba II	0.81
Prueba III	0.85
Prueba IV	0.86

Para las pruebas III y IV se considera inicialmente el sembrado de una solución óptima, que corresponde al mejor individuo inicial que se obtuvo en un entorno de prueba en el que no se considera la luz ambiental como una perturbación, en la población. En la figura 13 se ilustran los mejores individuos de cada generación de la prueba III. La optimización no fue muy efectiva según la figura 14, donde el comportamiento del sistema con la solución final está por debajo de la solución sembrada inicialmente, aunque hay algunas posiciones donde la solución final tiene un mejor rendimiento, la optimización no da buenos resultados como se esperaba. La optimización de la solución final podría mejorar significativamente si se modifican parámetros del algoritmo. En la figura 15 se muestra el valor de aptitud de cada uno de los individuos en cada generación de la prueba IV, estableciendo las mismas condiciones de la prueba II. Hubo una optimización significativa en el valor de aptitud del individuo final, a comparación de la prueba III, esto se ilustra en la figura 16, por lo tanto, el proceso de optimización logra que el robot evolutivo se adapte y minimice el efecto de la perturbación presente en el entorno de prueba, logrando que la optimización sea satisfactoria. De los cuatro experimentos, los valores de aptitud alcanzados por los individuos finales fueron similares como se ilustra en la tabla 8, sin embargo, mejores a pesar de las perturbaciones, por lo tanto, el sistema es capaz de adaptarse a las condiciones de su entorno, aún si se le brinda información inicial o no al sistema, lo cual es una característica de la optimización evolutiva. El valor de aptitud más alto se logró en la prueba IV, en donde la luz ambiental es una perturbación, pero gracias a la optimización evolutiva, se tiene un sistema robusto que logra un buen desempeño.

Otro de los aspectos a resaltar es que el valor de aptitud que se alcanza podría parecer bajo o regular, pero es un resultado que ofrece un buen comportamiento para el robot. En estas pruebas se emplean 10 generaciones, aunque si se aumenta este número podría mejorar la optimización.

CONCLUSIONES

En la implementación de un sistema robótico con controlador difuso-evolutivo se han obtenido buenos resultados para la tarea encomendada. Además, a pesar de que el sistema esté inicializado con una población de individuos generados completamente al azar, el conjunto de soluciones obtenido al final de la optimización logra buenas soluciones y el sistema robótico se comporta como se espera para la tarea en cuestión. También se concluye que es necesario hacer ajustes en el software, como calibración inicial, de modo que las mediciones obtenidas sean precisas y el algoritmo genético pueda trabajar para obtener mejores resultados. Se observó que en el caso de haber perturbaciones en el entorno de prueba, la optimización evolutiva permite que el sistema se adapte a las condiciones presentes. A pesar de ser pocas generaciones, la evaluación de individuos más aptos del sistema por medio de la función objetivo propuesta es un proceso lento, y es deseable mejorar esta etapa. Finalmente, para el prototipo robótico y la información extraída de los experimentos, se logra una implementación satisfactoria de un controlador difuso-evolutivo, que brinda cierta robustez, adaptabilidad y facilidad de interpretación.

AGRADECIMIENTOS

El autor Carlos Vargas Arvizu agradece a CONACyT por su beca de estudio No. 610956.

REFERENCIAS

1. Duarte, M., Sancho, M.O., and Lyhne Christensen, A. Evolution of hybrid robotic controllers for complex tasks. *Journal of Intelligent & Robotic Systems*, (2015) 78(3):463–484.
2. Gupta, S., and Singla, E. Evolutionary robotics in two decades: A review. *Indian Academy of Sciences*, (2015) 40(4):1169–1184.
3. Cervantes, L., and Castillo, O. Genetic optimization of modular type-1 fuzzy controllers for complex control problems. *Genetic Optimization of Modular Type-1 Fuzzy Controllers for Complex Control Problems: Soft Computing Applications in Optimization, Control, and Recognition*, (2013) 294:125-154.
4. Ishibashi, R., and Nascimento, C.L. Knowledge extraction using a genetic fuzzy rule-based system with increased interpretability. *IEEE 10th International Symposium on Applied Machine Intelligence and Informatics*. (2012) 247-252.
5. Ishibashi, R., and Nascimento, C.L., GFRBS-PHM: A genetic fuzzy rule-based system for phm with improved interpretability. *IEEE International Conference on Prognostics and Health Management*. (2013).

6. Chiou, J.-S., y Liu, M.-T. Using fuzzy logic controller and evolutionary genetic algorithm for automotive active suspension system. *International Journal of Automotive Technology*. (2009) 10(6):703–710.
7. Gao S., Zhang H., Zhang Y., Zhang B. Optimization to the inverted pendulum system by genetic fuzzy strategies. 24th Chinese Control and Decision Conference. (2012) 1562-1565.
8. Chávez, F., Fernández, F., Alcalá, R., Alcalá-Fdez, J., and Herrera, F. Evolutionary learning of a laser pointer detection fuzzy system for an environment control system. *IEEE International Conference on Fuzzy Systems*. (2011) 256-263.
9. Maldonado, Y., Castillo, O., and Melin, P. Optimization of membership functions for an incremental fuzzy pd control based on genetic algorithms. *Soft Computing for Intelligent Control and Mobile Robotics*, (2011) 318:195-211.
10. Yassami, M., and Ashtari, P. Using fuzzy genetic algorithm for the weight optimization of steel frames with semi-rigid connections. *International Journal of Steel Structures*, (2015) 15(1):63–73.
11. Sencan-Sahin, A., Kiliç, B., and Kiliç, U. Optimization of heat pump using fuzzy logic and genetic algorithm. *Heat Mass Transfer*, (2011) 47:1553–1560.
12. Ling Liu, Y., and Gomide, F. Participatory genetic learning in fuzzy system modeling. *IEEE International Workshop on Genetic and Evolutionary Fuzzy Systems*. (2013) 1-7.
13. Feng, L., and Qi-wen, Y. Improved genetic operator for genetic algorithm. *Journal of Zhejiang University SCIENCE*, (2002) 3(4):431–434.
14. Wei, J., and Wang, P. Optimization of fuzzy rule based on adaptive genetic algorithm and ant colony algorithm. *International Conference on Computational and Information Sciences*. (2010) 359-36.

