

Modelo Neuro-difuso para el control de temperatura ambiental y humedad del suelo para cultivar el insecto NIIJ (Llaveia Axin).

Neurofuzzy model for the control of ambient temperature and soil moisture to cultivate the insect NIIJ (Llaveia Axin).

José Alberto Morales Mancilla (1).
Instituto Tecnológico de Tuxtla Gutiérrez.
amancilla58@hotmail.com.

Héctor Guerra Crespo (2), I. T. de Tuxtla Gutiérrez, hgcresto@hotmail.com.

Aída Guillermina Cossío Martínez (3), I. T. de Tuxtla Gutiérrez, acossio_m@yahoo.com.

Francisco de Jesús Suárez Ruiz (4). I.T. de Tuxtla Gutiérrez, franciscosuares@prodigy.net.mx.

Joel Gómez Pérez (5). I.T. de Tuxtla Gutiérrez, jgp10_10@hotmail.com.

Jorge Antonio Orozco Torres (6). I.T. de Tuxtla Gutiérrez, orozcotja@hotmail.com.

Artículo recibido en octubre 14, 2015; aceptado en noviembre 28, 2015.

Resumen.

Este artículo presenta el modelo de un sistema neuro-difuso para el cultivo y producción del insecto Niij (Llaveia Axin), controlando la humedad del suelo y temperatura ambiental dentro de un vivario. Este modelo utiliza una red neuronal tipo feedforward con algoritmo backpropagation, el cual sirve para discretizar los datos de entrada, que se utilizan en un sistema difuso tipo Takagi-Sugeno como una solución para clasificar el conjunto de datos en subconjuntos caracterizados por un comportamiento no lineal, a partir del cual se generan las reglas con las cuales se tomarán las acciones de control de humedad y temperatura. Los algoritmos del sistema neuro-difuso fueron desarrollados y simulados en MatLab e implementados en una placa Arduino.

Palabras clave: Modelo neurodifuso, cochinita de la laca, redes neuronales, lógica difusa.

Abstract.

This paper presents a model of a system for growing neurofuzzy for the cultivation and production of insect Nijj (Llaveia Axin) controlling soil moisture and ambient temperature within a vivarium. This model uses a feedforward network type with backpropagation algorithm, which is used to discretize the input data, that they are used in fuzzy system Takagi-Sugeno type. As a solution to classify the data set into subsets characterized by a nonlinear behavior, from the rules which control measures were taken humidity and temperatura generated. Neurofuzzy system algorithms were developed and simulated in MatLab and implemente on an Arduino board.

Keywords: Neurofuzzy model, cochineal lacquer, neural networks, fuzzy logic.

1. Introducción.

El trabajo artesanal siempre ha sido un factor reconocible en la sociedad chiapaneca, muchas familias de diversas

regiones del estado se dedican exclusivamente a realizar esta importante labor social e histórica como medio de sustento económico que además permite preservar la cultura y tradiciones de Chiapas.

Probablemente una de las artesanías más reconocidas a nivel nacional producidas en el estado es la que se realiza por medio de la utilización de un producto natural conocido como “laca”. Este material orgánico se obtiene exclusivamente de los ácidos grasos de las hembras del insecto cochinilla (*Llaveia Axin*), aproximadamente se producen 2.2 kg de laca de 17000 hembras de dicho insecto (Suazo-Ortuño, Del Val-De Gortari y Benítez-Malvido, 2013).

Desde la época prehispánica se ha utilizado la cochinilla para la producción de la laca, siendo así una de las formas de arte más antiguas en el país y que actualmente se ha visto afectada debido diversos factores ambientales debido a la actividad humana.

El principal producto artesanal de Chiapa de Corzo es la laca. Chiapas, Guerrero y Michoacán son los únicos estados que siguen produciendo en México objetos laqueados. En las antiguas culturas chiapanecas se usaron las cortezas de ciertos frutos como recipientes. Algunos de estos son la calabaza, el tomatillo, la jícara, el huacal y los pimientos. Había recipientes lisos y pintados con una depurada técnica de decoración denominada laca o maque. Con la llegada de los españoles, se dieron cambios en la técnica del laqueado como puede observarse en las lacas de Michoacán y Guerrero, estados que tienen mayor influencia española y oriental, llegada esta última con la Nao de China.

Durante el siglo XIX, continuó el proceso de transformación de la laca original para consolidarse como una técnica mestiza, cuyos principales rasgos son los que hasta ahora siguen identificando a la laca de Chiapa de Corzo. Para elaborar la laca se usan diversos materiales de origen mineral, vegetal y animal. La grasa ajé (también conocido como axe), se prepara de la hembra del insecto *Llaveia Axin*. A esa grasa se agregan el aceite de chía o de chicalote y el mineral llamado dolomita llamado en Chiapa de Corzo, tizate. Estos productos se usan para la base del laqueado, o fondeado, sobre el que se decora con pinceles hechos de pelo de gato, con los motivos florales típicos de la sensibilidad de Chiapa de Corzo (Colegiomexsur.edu.mx, 2015).

Cabe destacar que la laca no es de uso exclusivo para las artes, en diversas industrias se utiliza esta materia para diversos usos; por ejemplo, en la industria cosmética es bien sabido que es un material utilizado en maquillajes y artículos de belleza facial, también se ha utilizado como cicatrizante para heridas en la comúnmente llamada medicina tradicional.

Actualmente la industria de la laca se ha venido deteriorando con el paso del tiempo debido a la disminución del insecto responsable de su producción y a la destrucción de su hábitat, a la sobreexplotación de la población de la cochinilla, que afecta de manera directa a las familias que dependen económicamente de la artesanía de la laca, las artes y la cultura de la nación.

2. Métodos.

Obtención de la laca.

Para poder llevar a cabo el proyecto se tuvo que hacer un estudio del proceso de fabricación de la laca, y hacer algunas preguntas a los artesanos del proceso para la obtención de la laca con el fin de profundizar en la problemática. La grasa o “axe se obtiene exclusivamente de la hembra del insecto Cochinilla *Llaveia axin* y es la materia prima para la realización de las artesanías de la laca. Las artesanías de laca, se realiza aún en los estados de Chiapas (municipio de Chiapa de Corzo), Guerrero (Olinalá) y Michoacán (Uruapan y Pátzcuaro), (Barrera y Prado, 2003), (Martínez, 2006). Esta grasa presenta alto contenido de ácidos grasos libres y triglicéridos, y por este motivo es empleada como cosmético y ungüento medicinal para tratar quemaduras e infecciones de la piel (Jenkins, 1970). Para obtener 2 Kg. de esta materia prima (grasa animal) se requiere coleccionar un promedio de 17,000 insectos hembras adultos.

La sobreexplotación del insecto de la laca, aunado a la destrucción de su habitar natural por la práctica de la ganadería intensiva y la agricultura, ha ocasionado la disminución de las poblaciones de este importante recurso biótico, lo que pone en riesgo esta importante actividad, que es base para la economía, tradición y cultura de muchas familias de artesanos en México.

Cuando la temporada de las lluvias ha concluido, las hembras adultas, caen al suelo y ahí se cubre de una capa algodonosa que protege la cutícula del insecto y proporciona temperatura corporal al insecto. En esta etapa la hembra de la cochinilla logra ovipositar entre 300 a 500 huevecillos. La hojarasca y el suelo cubren a las hembras y sus huevecillos, y de esta manera quedan en etapa de latencia hasta que se presente condiciones de humedad y temperatura adecuada para que se inicie un nuevo ciclo de vida.

Ciclo de vida de la cochinilla.

La eclosión de los huevecillos coincide con el inicio de la temporada de lluvias, el suelo se humedece y hay variaciones de temperatura y humedad, así como difusión de oxígeno, bajo esta condición micro ambiental permite la emergencia o eclosión de las “ninfas” que se desplazan rápidamente y ascienden por el tronco de los arboles hospederos en busca de ramas y corteza húmedas y ricas en nutrientes. Las ninfas se fijan a la corteza de las ramas jóvenes y succionan la savia rica en sales, azúcares y nutrientes. Alrededor de 35 a 40 días tarda en alcanzar la etapa adulta. Una hembra adulta llega a medir hasta aproximadamente 1.5 cm de longitud. En las ramas los insectos de la laca protegen de una cubierta algodonosa y al terminar las lluvias caen al suelo e inicia en la etapa de incubación e inicio del ciclo.

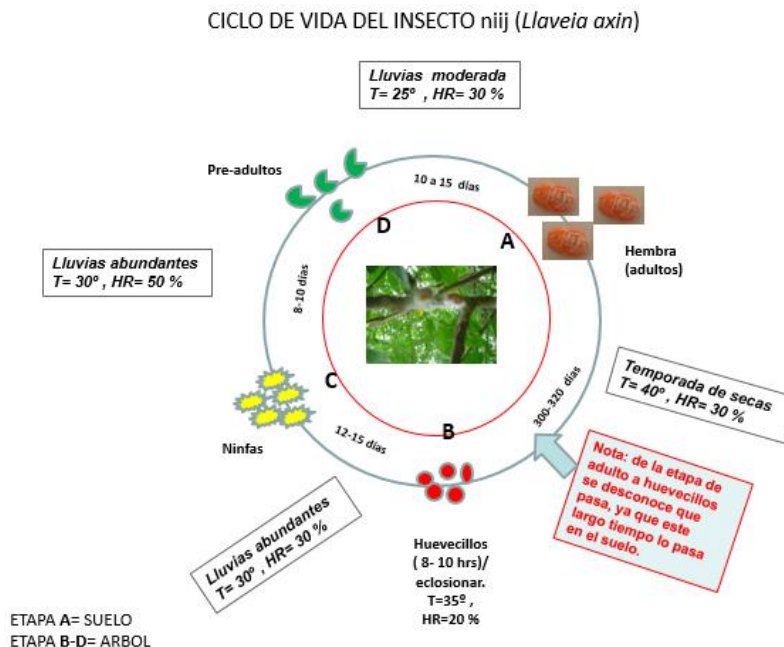


Figura 1. Ciclo de vida de la cochinilla *Llaveia Axin*.

Sistemas Neuro-Difusos.

Para el desarrollo de este proyecto se utilizó un sistema de control Neuro-Difuso el cual utiliza la combinación de dos técnicas de la inteligencia artificial, como son las redes neuronales artificiales (RNA) y la lógica difusa (LD). Las dos técnicas importantes de la inteligencia artificial, los Sistemas Difusos (SD) y las Redes Neuronales Artificiales (RNA) actualmente se aplican en muchos campos, como en los sistemas de control, producción, diagnóstico y supervisión. Han mejorado y evolucionado a través de los años para irlo adaptando a diferentes

necesidades que vayan surgiendo y a los avances tecnológicos. Como sistemas de LD y RNA han manejado juntos el concepto de una fusión entre ellas, lo que ha comenzado a tomar forma. Los Sistemas Neuro-Difusos (SND) utilizan las ventajas de ambas técnicas. Esta combinación de características hace que este tipo de sistemas sea muy útil para la solución de problemas complejos, también para las tareas de diagnóstico y mediciones técnicas.

Red neuronal artificial.

Una neurona es una función no lineal acotada y parametrizada. Por conveniencia una función parametrizada lineal a menudo se denomina una neurona lineal. Las variables de las neuronas son a menudo llamadas variables de entrada y variables de salida. Una neurona se puede representar como se muestra en la figura 2. Esta representación se debe a la inspiración biológica que impulsó el especial interés de formalizar la neurona entre los años 1940 y 1970. (Dreyfus, 2004).

La función f puede ser parametrizada de cualquier forma. Dos tipos de parametrización actualmente están en uso. Los parámetros están asignados a las entradas de las neuronas; la salida de la neurona es una combinación no lineal de $\{X_i\}$, ponderados por los parámetros $\{W_i\}$ los cuales son a menudo llamados pesos, para ser una reminiscencia de la inspiración de una red neuronal biológica con pesos sinápticos. El potencial v es frecuentemente utilizado para representar una suma ponderada de las entradas, con una constante adicional llamada "bias".

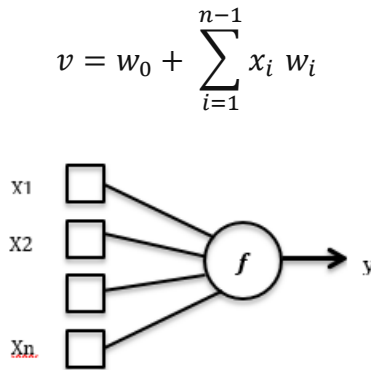


Figura 2. Una neurona es una función no lineal limitada $y = f(x_1, x_2, \dots, x_n; w_1, w_2, \dots, w_p)$ donde $\{x\}$ son las variables y $\{w\}$ son los pesos.

La figura 3 se muestra un modelo que representa la neurona artificial. Las entradas x_1, x_2, \dots, x_n son introducidas en una neurona artificial. Estas entradas, definidas por un vector X , corresponden a las señales de la sinapsis de una neurona biológica. Cada señal se multiplicará por un peso asociado w_1, w_2, \dots, w_n , antes de ser aplicado el sumario etiquetado por Σ . Cada peso corresponde a la fuerza de una conexión sináptica, es decir el nivel de concentración iónica de la sinapsis, y se representa por un vector W .

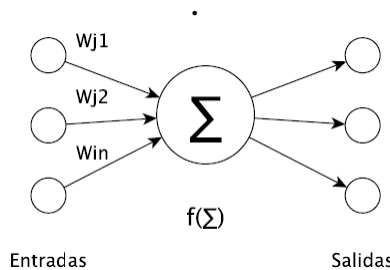


Figura 3. Representación de una neurona artificial.

La sumatoria, que corresponde al cuerpo de la neurona, suma todas las entradas ponderadas algebraicamente, produciendo una salida que se denomina E , así:

$$E = x_1w_1 + x_2w_2 + \dots + x_nw_n$$

Aprendizaje de la red neuronal.

El aprendizaje de una red neuronal artificial consiste en la determinación de los valores precisos de los pesos para todas sus conexiones, que la capacite para la resolución eficiente de un problema. El proceso general de aprendizaje consiste en ir introduciendo paulatinamente todos los ejemplos del conjunto de aprendizaje, y modificar los pesos de las conexiones siguiendo un determinado esquema de aprendizaje.

Una vez introducidos todos los ejemplos se comprueba si se ha cumplido cierto criterio de convergencia; de no ser así se repite el proceso y todos los ejemplos del conjunto vuelven a ser introducidos. El criterio de convergencia depende del tipo de red utilizado o del tipo de problema a resolver. La finalización del periodo de aprendizaje se puede determinar:

- Mediante un número fijo de ciclos.
- Cuando el error descienda por debajo de una cantidad preestablecida.
- Cuando la modificación de los pesos sea irrelevante.

Control con red neuronal *feedforward*.

Para poder controlar la humedad y la temperatura se diseñó la red neuronal *feedforward* con algoritmo *backpropagation*, se entrenó en MatLab versión 8.6.0.267246 (R2015b), para calcular los pesos y los umbrales adecuados utilizando como datos de entrada la información proporcionada por el Biólogo Reiner Rincón Rosales que es investigador del área de posgrado del Instituto Tecnológico de Tuxtla Gutiérrez, quien actualmente está estudiando la cochinilla de la laca.

La arquitectura del Perceptrón multicapa (red *feedforward*) se caracteriza porque tiene sus neuronas agrupadas en capas de diferentes niveles. Cada una de las capas está formada por un conjunto de neuronas y se distinguen tres tipos de capas diferentes: la capa de entrada, las capas ocultas y la capa de salida. Las neuronas en la capa de entrada no actúan como neuronas propiamente dichas, sino que se encargan únicamente de recibir las señales o patrones que proceden del interior y propagar dichas señales a todas las neuronas de la siguiente capa. La última capa actual como salida de la red, proporcionando al exterior la respuesta de la red para cada uno de los patrones de entrada. Las neuronas de la capa oculta realizan un procesamiento no lineal de los patrones recibidos (Galván y Isasi, 2004).

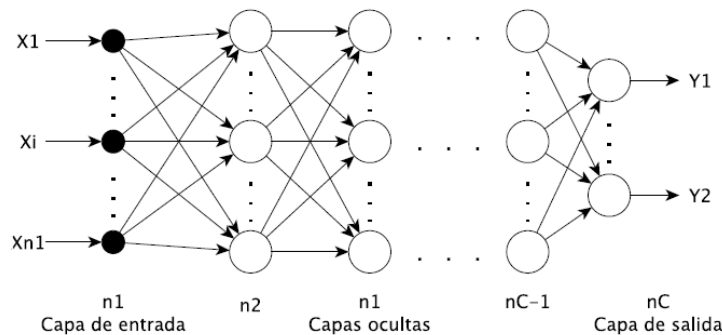


Figura 4. Arquitectura de la red feedforward.

Red neuronal para el control de temperatura.

El diseño de esta red se realizó con el fin de controlar la temperatura dentro de un vivario donde se reproduce la cochinilla de la laca. La red neuronal presenta una neurona en la primera capa, tres neuronas en la capa oculta y una neurona en la capa de salida como se muestra en la figura 5.

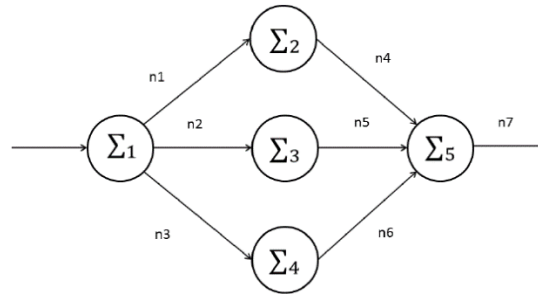


Figura 5. Red neuronal feedforward.

Para obtener los datos de la temperatura del medio ambiente, se utilizó un sensor de temperatura LM35. El rango de trabajo del LM35 es de 0 a 150 °C, se consideró controlar un rango de temperatura para la red neuronal de 15 a 34.5 °C en incrementos de 0.5 °C, como el PWM de Arduino da como máximo 255 pulsos, se divide 255/31 dando como resultado 8.5 PWM por cada incremento, como se puede observar en la tabla 1, donde se muestra la entrada y la salida de datos. En Tuxtla Gutiérrez, la capital del estado de Chiapas, el clima es tropical húmedo. En los meses más calurosos marzo a mayo, los termómetros llegan hasta los 36 grados centígrados y descienden hasta los 18 grados centígrados en los meses de noviembre a enero. Se ha observado que en las primeras lluvias de mayo la cochinilla nace. Es por ello que se presenta en la Tabla 1 los valores de salida de la red neuronal. Si la temperatura llega a 34.5 grados centígrados el sistema dejaría de funcionar y con ellos dejaría de proporcionar calor.

Tabla 1. Datos de temperatura como entrada y salida de la red neuronal.

No.	Temperatura	Salida de la red	No.	Temperatura	Salida de la red
1	15	255	21	25	135
2	15.5	249	22	25.5	129
3	16	243	23	26	123
4	16.5	237	24	26.5	117
5	17	231	25	27	111
6	17.5	225	26	27.5	105
7	18	219	27	28	99
8	18.5	213	28	28.5	93
9	19	207	29	29	86
10	19.5	201	30	29.5	79
11	20	195	31	30	72
12	20.5	189	32	30.5	65
13	21	183	33	31	58
14	21.5	177	34	31.5	51
15	22	171	35	32	44
16	22.5	165	36	32.5	36
17	23	159	37	33	28
18	23.5	153	38	33.5	20
19	24	147	39	34	10
20	24.5	141	40	34.5	0

Teniendo esos datos se procedió a codificarlos en Matlab para la red neuronal feedforward con algoritmo backpropagación, utilizando una función de transferencia Tagsig en la primera capa, para las tres neuronas intermedias de la capa oculta se utilizó la función de transferencia Logsig, y en la capa de salida se utilizó la función de transferencia Purelin.

Red neuronal para el control de la humedad.

El diseño de esta red fue para controlar la humedad, fue la arquitectura de la red neuronal mostrada en la figura 5. Se tiene como entrada a la red neuronal, los datos de la humedad que es captado por un sensor dht11, y como salida la velocidad en PWM para el control de la velocidad de un motor de DC.

Para calcular los pesos de la red neuronal se utilizó el Toolbox de redes neuronales de Matlab. Donde se tiene una neurona en la capa de entrada tres neuronas en la capa intermedia y una neurona en la capa de salida. Una vez que la red neuronal ha aprendido, se tienen los pesos y umbrales adecuados, posteriormente se procedió a desarrollar la red neuronal en una placa Arduino.

El sensor de humedad utilizado genera valores entre 0 (muy húmedo) y 1023 (muy seco). En la tabla 2 se muestran los valores de humedad de entrada a la red neuronal y los valores de salida esperados.

Tabla 2. Datos de entrada de humedad y salida de la red neuronal.

CONTROL DE HUMEDAD DEL SUELO			
No.	VALORES DE ENTRADA	VALORES DE SALIDA	CONJUNTOS
1	1023	1	MUY SECO
2	1000	0.9	SECO
3	900	0.8	SECO
4	800	0.7	MEDIO HUMEDO
5	700	0.6	MEDIO HUMEDO
6	600	0.5	HUMEDO
7	500	0.4	HUMEDO
8	400	0.3	HUMEDO
9	300	0.2	HUMEDO
10	200	0.1	HUMEDO
11	100	0	MUY HUMEDO
12	50	0	MUY HUMEDO
13	0	0	MUY HUMEDO

Con estos datos se procedió a codificarlos en Matlab para la red neuronal feedforward con algoritmo backpropagación, utilizando una función de transferencia Logsig en la primera capa, para las tres neuronas intermedias de la capa oculta se utilizó la función de transferencia Logsig, y en la capa de salida se utilizó la función de transferencia Purelin. Una vez diseñadas, entrenadas y probadas las redes neuronales de temperatura y humedad, se procedió al diseño de la lógica difusa.

Lógica difusa.

La denominada lógica borrosa (fuzzy logic) permite tratar información imprecisa, como estatura media, temperatura baja o mucha fuerza en términos de conjuntos borrosos o difusos (imprecisos), la lógica difusa está basada en la teoría de conjuntos borrosos, esta teoría parte de la teoría clásica de conjuntos, añadiendo una función de pertenencia definida como un número real entre 0 y 1. Así se introduce el concepto de conjunto o subconjunto borroso asociado a un determinado valor lingüístico, definido por una palabra, adjetivo o etiqueta lingüística A. Para cada conjunto o

subconjunto borroso se define una función de pertenencia o inclusión (μ_{A^t}) que indica el grado en que la variable t está incluida en el concepto representado por la etiqueta A . Como puede verse en la Figura 6 para el valor lingüístico *estatura_de_una_persona* podrían definirse tres subconjuntos borros, cada uno identificado por una etiqueta, Bajo, Medio, Alto, y con una función de inclusión o pertenencia $\mu_{Bajo}(x)$, $\mu_{Medio}(x)$, $\mu_{Alto}(x)$. Los conjuntos borrosos permiten agrupar objetos o sucesos por el valor de una cierta magnitud (Molina, 2001).

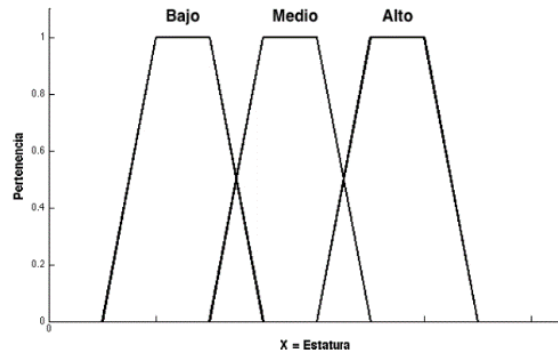


Figura 6. Lógica difusa.

En los conjuntos difusos la función de pertenencia puede tomar valores del intervalo entre 0 y 1, y la transición entre el cero y uno es gradual y no cambia de manera instantánea, como pasa con los conjuntos clásicos. Un conjunto difuso en un universo en discurso puede definirse con la siguiente ecuación:

$$A = \{(X_1, \mu_A(x)) | x \in U\}$$

Donde:

μ_A : Es la función de pertenencia entre cero y uno.

$\mu_A(x)$: Es el grado de pertenencia de la variable x y U es el universo en discurso.

Entre más cerca este A del valor 1, mayor será la pertenencia del objeto x al conjunto A .

Diseño del controlador difuso con Takagi-Sugeno.

Sugeno propuso el diseño de controladores difusos a partir de encontrar un modelo difuso de la planta que relacionara las entradas con sus salidas. Una vez que se ha diseñado un modelo difuso de la planta, es posible generar un controlador que parte de la salida deseada para determinar la entrada necesaria en el sistema, con un proceso de deducción inversa. A continuación, se explica el método por Takagi y Sugeno para la identificación difusa de sistemas y su aplicación en modelado y control. Este es un método matemático para hacer un modelo difuso de un sistema (Ponce 2010).

El primer paso es definir la estructura de las implicaciones. A continuación, se define una implicación difusa Z .

$$Z: \text{Si } f(X_1 \text{ es } A_1, \dots, X_k) \text{ entonces } y = g(X_1, \dots, X_k)$$

Donde:

y : Variable de la consecuencia, cuyo valor es inferido.

X_1, \dots, X_k : Variables de la premisa

A_1, \dots, A_k : Conjuntos difusos con funciones de membresía lineales, representando un subespacio difuso, en el cual la implicación Z puede ser usada para el razonamiento.

f: Función lógica que conecta las proposiciones en la premisa.

g: Función que implica el valor de *y* cuando X_1, \dots, X_k satisface la premisa.

3. Desarrollo.

El sistema Neuro-Difuso para el control de humedad y temperatura para la crianza de la cochinilla de la laca se describe a continuación:

1. El proyecto consta de dos etapas, un sistema de Red Neuronal para el control de la Humedad y una para el control de Temperatura, como se muestra en la figura 7.

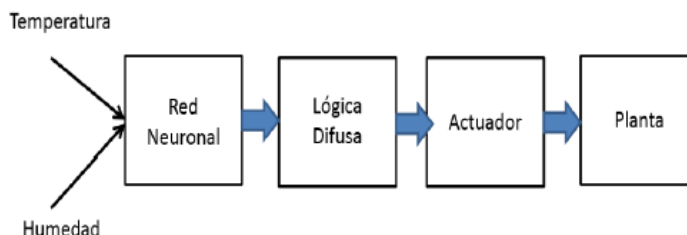


Figura 7. Modelo del sistema Neuro-Difuso.

2. La primera etapa es la red neuronal feedforward donde entran los valores de humedad y temperatura, la segunda etapa es un controlador difuso tipo Takagi-Sugeno para la acción de control de las variables de humedad y temperatura, ambos montados en una placa Arduino.
3. Se realizó la codificación en Matlab de la Red Neuronal, en ambos se utilizaron como primera capa Logsig, segunda capa Logsig y tercera capa Purelin, correspondiente a la Humedad y a la Temperatura en forma separada. Los resultados de dicho entrenamiento que son muy cercanos a los esperados, en la figura 8 se muestra el código para el entrenamiento y simulación de la red neuronal para la humedad en MatLab. En la figura 9 se muestran los pesos y las bias de la red neuronal para la humedad.

```
>> % Red neuronal de 3 capas para el control de humedad
>> % Una neurona en la capa de entrada, tres en la capa intermedia
>> % Una neurona en la capa de salida
>> entrada = [1023 1000 900 800 700 600 500 400 300 200 100 50 0];
>> salida = [1 0.9 0.8 0.7 0.6 0.5 0.4 0.3 0.2 0.1 0 0 0];
>> net = newff(minmax(entrada), [1, 3, 1], {'logsig', 'logsig', 'purelin'}, 'trainlm');
>> net = init(net);
>> net.trainparam.goal = 1e-05;
>> %entrenando la red neuronal
>> nete = train(net, entrada, salida);
>> %simulando la red neuronal
>> sim(nete, entrada)
```

Figura 8. Creación, entrenamiento y simulación de la red neuronal para el control de humedad.

```

>> %pesos y bias de la capa uno    >> % pesos y bias de la capa dos    >> % pesos de la capa tres
>> nete.iw{1,1}                    >> nete.Lw{2,1}                    >> nete.Lw{3,2}

ans =                                ans =                                ans =
    0.0070                            70.3751                            7.1293 -14.9181  10.0442
                                     -0.1713
                                     18.5974
>> %bia de la capa uno              >> % bia de la capa dos              >> %bia de la capa tres
>> nete.b{1,1}                      >> nete.b{2,1}                      >> nete.b(3)

ans =                                ans =                                ans =
   -5.0153                            -66.7228                            [-1.1910]
                                     0.2937
                                     3.2709

```

Figura 9. Pesos y bias de las capas de la red neuronal para el control de humedad.

4. Los datos de salida de la red neuronal de humedad, pasan al control difuso. La humedad ideal para la cochinilla, está entre los 200-600 (NORMAL) en este caso 500, arriba de 600 es BAJA y debajo de 200 es ALTA (muy húmedo). Para mantener la humedad a 500 se hace circular corriente por la ley de Joule. Si la humedad es baja debe generar una corriente ALTA, si la humedad es ALTA la corriente es cero. Se desea mantener la humedad NORMAL entre los 200 y 600, aplicar la lógica difusa Takagi-Sugeno.

Reglas:

- Si la humedad es BAJA entonces la corriente es ALTA.
- Si la humedad es NORMAL entonces la corriente es PEQUEÑA.
- Si la humedad es ALTA entonces la corriente es BAJA.

El diagrama de entrada de la lógica difusa para el control de humedad se muestra en la figura 10.

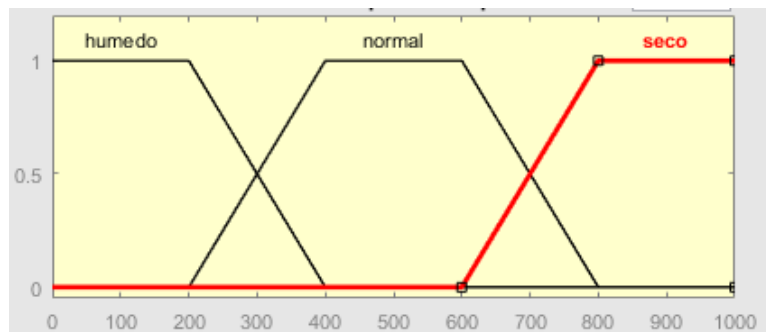


Figura 10. Diagrama de los subconjuntos de la lógica difusa de la humedad.

5. Los datos de salida de la lógica difusa Takagi-Sugeno son números concretos.

Las reglas de control son:

Cuando la temperatura es BAJA la corriente=2.0

Cuando la temperatura es NORMAL la corriente=0.5

Cuando la temperatura es ALTA la corriente=0.0

6. El mismo procedimiento se aplica para el control de temperatura, por lo que no se explica.

Conclusiones.

Los resultados obtenidos en el diseño e implementación del sistema Neuro-Difuso, demuestran que el sistema cumple con las características para el proceso de la crianza de la cochinilla de la laca. Los controladores lógicos o difusos surgen de la necesidad de resolver problemas complejos con resultados imprecisos, para los cuales las matemáticas y la lógica clásica no son suficientes y que no son de fácil caracterización. El desarrollo de este trabajo servirá para futuros trabajos de investigación con la aplicación de los sistemas Neuro-Difusos. Para visualizar de manera gráfica los procesos del control de humedad y temperatura, se puede pasar a LabView la red neuronal y la lógica difusa, quedando la placa Arduino como una tarjeta de adquisición de datos.

Créditos.

Los autores agradecen al Dr. Reiner Rincón Rosales por la colaboración brindada en la parte Biotecnológica y dan créditos y agradecimientos al Tecnológico Nacional de México por el financiamiento otorgado al proyecto “Control neurodifuso para el cultivo y producción del insecto Llaveia Axin (NIIJ) en habitat artificial para el desarrollo sustentable de las artesanías de la laca en Chiapas, México” clave 5473.14-P del I. T. de Tuxtla Gutiérrez.

Referencias Bibliográficas.

- Barrera, F. y A. Prado (2003).** Investigación y recuperación del aje como materia prima artesanal 2001-2003. *Ukata: Revista del Arte Popular Michoacano*, Editorial Grupo IBISA. México. 51 p.
- Colegimexsur.edu.mx. (2015).** *Arte Popular / SECRETARIA de TURISMO del ESTADO de CHIAPAS.* Recuperado el 12 junio 2014, desde <http://www.colegimexsur.edu.mx/ifapop.html>.
- Dreyfus, G. (2004).** *Neural Networks Methodology Applications.* (P. Eyrolles, Ed.) (2nd ed.). Paris: Springer.
- Galván, L. I. M. y Isasi, V. P. (2004).** *Introducción a las Redes de Neuronas Artificiales, Redes de Neuronas Artificiales. Un Enfoque Práctico*, Prentice Hall, Madrid, (pp. 1-14).
- Jenkins, K. D. (1970).** The fat-yielding coccid, Llaveia, a monophlebine of the Margarodidae. *Pan Pacific Entomologist* 46:79-81.
- Martínez FM. (2006).** *La laca de los Achies. Artículo de semanario de prensa libre.* <http://www.prensalibre.com/pl/domingo/archivo/revistad/2006/marzo06/190306/dfondo.shtml>; 89:18-21.
- Molina A. S. (2001).** *Redes Neuronales y sistemas difusos.* Alfaomega, Mexico DF, pp. 197-225.
- Ponce C. P. (2010).** *Inteligencia Artificial con aplicaciones a la ingeniería.* México: Alfaomega Grupo Editor, S.A de C.V.

Información de los autores.



José Alberto Morales Mancilla es Maestro en Ciencias de la Computación, egresado del Centro Nacional de Investigación y Desarrollo Tecnológico CENIDET, en Cuernavaca, Morelos. Es profesor de tiempo completo en el área de Ingeniería en Sistemas Computacionales del Instituto Tecnológico de Tuxtla Gutiérrez y es investigador en el I.T. de Tuxtla Gutiérrez desde 1991, pertenece al cuerpo académico “Tecnologías de información para el desarrollo regional”, tiene el reconocimiento de Perfil Deseable PROMEP junio del 2012, cuenta con el reconocimiento del Sistema Estatal de Investigadores, fundó y dirige el área de trabajo “tecnología para población con necesidades educativas especiales” capacidades.sistemastuxtla.net y colabora en el área “aplicaciones sobre mapas” www.mexmapa.com. Candidato a doctor en desarrollo tecnológico por la Universidad de Ciencia y Tecnología Descartes.



Héctor Guerra Crespo es Doctor en Sistemas Computacionales por la Universidad del Sur, Tuxtla Gutiérrez, Chiapas, en 2011. Es profesor en el área de Ingeniería en Sistemas Computacionales del Instituto Tecnológico de Tuxtla Gutiérrez y en el área de Licenciatura en Sistemas Computacionales de la Universidad Autónoma de Chiapas, en ambas desde 1995, líder del cuerpo académico “tecnologías de información para el desarrollo regional” donde dirige el área de trabajo “aplicaciones sobre mapas” y colabora en la línea “tecnología para población con necesidades educativas especiales”. www.hectorguerracrespo.com



Aída Guillermina Cossío Martínez es Maestra en Ciencias en Administración por el Instituto Tecnológico de Tuxtla Gutiérrez en 2002. Es profesora de tiempo completo del área de Ingeniería en Sistemas Computacionales del Instituto Tecnológico de Tuxtla Gutiérrez, desde 1994. Se especializa en la formulación y evaluación de proyectos, así como el emprendimiento y desarrollo de planes de negocio.



Francisco de Jesús Suárez Ruiz es Maestro en Ciencias de la Computación por la Universidad Pablo Guardado Chávez en 2008. Es profesor de tres cuartos de tiempo en el área de Ingeniería en Sistemas Computacionales del Instituto Tecnológico de Tuxtla Gutiérrez desde el año 2000. Actualmente es jefe del departamento de Sistemas y Computación del Instituto Tecnológico de Tuxtla Gutiérrez desde el 2013. Actualmente imparte las materias de programación.



Joel Gómez Pérez es Maestro en Ciencias en Ingeniería en Mecatrónica por el Instituto Tecnológico de Tuxtla Gutiérrez en el 2013, colabora con la Universidad Politécnica de Chiapas y en el área de Ingeniería en Mecatrónica de la Universidad Valle de México. Es profesor de tiempo completo en el área de Ciencias Básicas del Instituto Tecnológico de Tuxtla Gutiérrez desde el 2010. Se especializa en el área de automatización y control. Es miembro del Sistema Estatal de Investigadores.



Jorge Antonio Orozco Torres es Maestro en Ciencias Computacionales por la Universidad Autónoma del Estado de Hidalgo (UAEH) en el 2008. Es profesor de tiempo completo del área de Ingeniería Industrial del Instituto Tecnológico de Tuxtla Gutiérrez. Se especializa en Computación Educativa. Candidato a doctor en desarrollo tecnológico por la Universidad de Ciencia y Tecnología Descartes. Evaluador por CACEI A.C. en las áreas de Ingeniería Industrial y Ciencias Computacionales.