

Modelo neurodifuso para el control de humedad del suelo en cultivo hidropónico para la planta de tomate.

Neurofuzzy model for the soil moisture control in hydroponic crop for tomato plant.

Mario Culebro de la Piedra (1).
Estudiante, Instituto Tecnológico de Tuxtla Gutiérrez.
mculebrodlp@hotmail.com.

Marcos Rafael Gómez Gómez (2), estudiante, I.T. de Tuxtla Gutiérrez, marcosaguila_11@hotmail.com.

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

Joaquín Adolfo López Molina (4), I.T. de Tuxtla Gutiérrez, jmontes@ittg.edu.mx.

Néstor Antonio Morales Navarro (5), I.T. de Tuxtla Gutiérrez, nstrmorales@gmail.com.

Jorge Octavio Guzmán Sánchez (6), I. T. de Tuxtla Gutiérrez, jogs78@gmail.com.

Artículo recibido en septiembre 26, 2016; aceptado en diciembre 08, 2016.

Resumen.

Este trabajo presenta el diseño de un sistema neurodifuso para un cultivo hidropónico de la planta de tomate, que controla la humedad del suelo mediante un riego automatizado. Este modelo utiliza una red neuronal tipo feedforward con algoritmo backpropagation (topología multicapa) y el sistema difuso tipo Takagi-Sugeno. La red neuronal emplea los valores de entradas de los sensores de humedad y el sistema difuso los valores de salida para el control de humedad del cultivo. Los algoritmos del sistema neurodifuso fueron desarrollados y simulados en MatLab e implementados en una placa Arduino.

Palabras clave: Modelo neurodifuso, cultivo hidropónico, redes neuronales, lógica difusa.

Abstract.

This work presents the design of a neurofuzzy system for a hydroponic crop of the tomato plant, which controls the soil moisture through an automated irrigation. This model uses a feedforward neural network with backpropagation algorithm (multiplayer topology) and the Takagi-Sugeno type diffuse system. The neural network uses the input values of the moisture sensors and diffuse system the output values for the control of crop moisture. The algorithms of the neurofuzzy system were developed and simulated in MatLab and implement on Arduino board.

Keywords: Neurofuzzy model, hydroponic crop, neural networks, fuzzy logic.

1. Introducción.

El tomate es uno de los productos agrícolas con mayor valor económico a nivel mundial; además cabe destacar que es uno de los primeros cultivos producidos por el método hidropónico. Es importante mencionar que gracias a los altos estándares de producción que presenta el tomate mexicano, es una de las hortalizas con mayor demanda tanto a nivel nacional como internacional ya que cuenta con un alto grado de calidad e inocuidad, que lo hacen, una de las

especies vegetales con más rendimiento y rentabilidad. De acuerdo con datos arrojados por el Servicio de Información Agroalimentaria y Pesquera (SIAP), a lo largo del 2014 México produjo cerca de 2.8 millones de toneladas de jitomate, razón por la cual su producción ocupa el segundo lugar después del cultivo de chile.

En México existen seis millones de hectáreas de riego y 15 a 16 millones de hectáreas de temporal, por lo tanto la frontera productiva está muy limitada y los sistemas tradicionales de producción no alcanzan a cosechar lo que nuestro país necesita de alimentos. Un gran reto para el país es el de generar tecnologías que ayuden al desarrollo de una mejor producción, manteniendo los altos estándares de calidad adecuados a las condiciones que hoy en día se presentan.

Ante ello están surgiendo sistemas alternativos que pueden ayudar a cubrir el déficit en la producción de alimentos del país, como lo es la hidroponía, una forma de producción que no necesita de suelos y por lo mismo no depende de fenómenos meteorológicos, con ello permite reducción de costos de manera considerable y cosechas fuera de estación, además de que brinda elevados rendimientos y alta calidad en los alimentos. Lo anterior se expuso en el marco del Primer Curso-Congreso Internacional de Hidroponía en México, organizado por la Asociación Hidropónica Mexicana, presidida por Gloria Samperio Ruiz.

El sistema de hidroponía consiste en efectuar cultivos prescindiendo de la tierra, utilizando sustratos inertes (grava, arena, cascarillas, etc.) e implica la alimentación de la planta mediante una solución nutritiva disuelta en agua, así como el máximo aprovechamiento de este vital líquido al utilizar riego por goteo o un sistema de reciclaje.

Samperio (2014) señaló que entre sus proyectos a corto plazo está el de seguir enseñando la hidroponía en forma básica para cubrir el renglón de autoconsumo y después ir a un nivel más avanzado de venta para que posteriormente se logre la exportación a través de muchos productores.

Entre los avances de los propios cultivos de la Asociación Hidropónica Mexicana se tiene el de que en diciembre de 1998 se cultivaron 109 toneladas de jitomate en cinco mil metros. Samperio explicó que existen tres niveles de la hidroponía: el de autoconsumo, el de comercio y el de industrial. Para una producción de autoconsumo, se recomienda conocer siempre la información de lo que se va a cultivar y siguiendo una serie de procedimientos sencillos a más tardar en 90 días se tiene una cosecha para autoconsumo.

2. Métodos.

Cultivo hidropónico del tomate.

El sustrato utilizado en el cultivo hidropónico se conformó de una mezcla entre *peat moss* con arena, el *peat moss* es un material con buena retención de humedad, buena aireación y alto contenido de materia orgánica, por su parte la arena es uno de los sustratos que más se emplean en los cultivos hidropónicos por su facilidad de uso, granulometría y porque genera un buen drenaje general al homogeneizarse bien con el resto de componentes del sustrato.

El crecimiento y desarrollo del jitomate comprende de 3 a 5 etapas, las cuales tienen una duración diferente dependiendo del ambiente y las técnicas de producción, pero sobre todo, dependiendo del hábito de crecimiento (determinado o indeterminado). En igualdad de condiciones lo normal es que la duración de cada etapa sea mayor en las variedades indeterminadas. Las etapas de germinación, crecimiento, floración y fructificación se dan mejor bajo un ritmo alternante de temperatura entre el día y la noche que a una temperatura constante (Maroto, 1989).

Resh (2002) considera que el tomate es una planta con exigencias relativamente bajas en cuanto a la humedad del suelo, lo cual es debido a la armonía estructural entre el sistema radical, que absorbe agua con facilidad, y el sistema foliar, que gasta agua con dificultad. Una deficiencia de humedad provoca reducción del crecimiento, reduce la etapa de crecimiento y el periodo funcional de las hojas.

La humedad del sustrato del cultivo hidropónico es controlada a través de un riego automatizado, el funcionamiento del sistema de riego es activado cuando los sensores detectan una baja humedad dentro del sustrato (suelo) y es apagado cuando los sensores detectan un porcentaje alto de humedad.

Sistemas neurodifusos.

Para este proyecto se utilizó un sistema de control Neuro-Difuso, los sistemas basados en modelos neurodifusos combinan las fortalezas de las redes neuronales y la lógica difusa; por un lado tenemos que las redes neuronales son buenas en el reconocimiento de patrones, capacidad adaptativa, aprendizaje, paralelismo y generalización, en cambio la lógica difusa provee un mecanismo de inferencia bajo incertidumbre cognitiva de una forma como lo realiza el ser humano.

Redes neuronales.

Las redes neuronales artificiales son un campo muy importante dentro de la inteligencia artificial. Romero (2007) describe, inspirándose en el comportamiento conocido del cerebro humano (principalmente el referido a las neuronas y sus conexiones), se permite resolver problemas que no pueden ser solucionados usando algoritmos convencionales.

Las redes neuronales pueden ser clasificadas según el tipo de aprendizaje (supervisado, no supervisado), el tipo de aplicación y la arquitectura de la conexión (monocapa, multicapa). Una de las principales características de las redes neuronales, es su capacidad de aprendizaje, a la hora de entrenar la red neuronal, se busca conseguir que una aplicación determinada, para un conjunto de entradas produzca el conjunto de salidas deseadas o mínimamente consistentes. Este proceso de entrenamiento consiste en la aplicación secuencial de diferentes conjuntos o vectores de entrada para que se ajusten los pesos de las interconexiones según un procedimiento predeterminado.

En las redes neuronales artificiales existen dos capas con conexiones, con el mundo exterior. Una capa de entrada, donde se presentan los datos a la red, y una capa de salida que mantiene la respuesta de la red a una entrada. El resto de las capas reciben el nombre de capas ocultas. La Figura 1, muestra el aspecto de una red neuronal artificial.

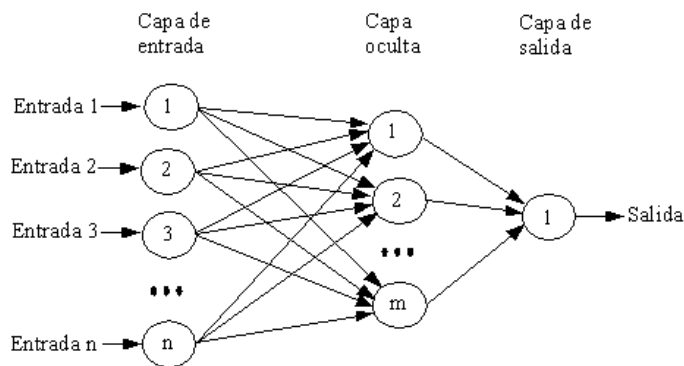


Figura 1. Arquitectura de una Red Neuronal Artificial Multicapa.

Red neuronal tipo feedforward.

La red neuronal tipo feedforward, como su nombre lo indica, en este tipo de redes se empieza con un vector de entradas el cual es equivalente en magnitud al número de neuronas de la primera capa de la red, las cuales procesan dicho vector elemento por elemento en paralelo. La información, modificada por los factores multiplicativos de los pesos en cada neurona, es transmitida hacia adelante por la red pasando por las capas ocultas para finalmente ser procesada por la capa de salida.

Es importante mencionar que las redes feedforward son las más sencillas en cuanto a implementación y simulación, pero su desempeño es bueno para aplicaciones en los que no se requiera que la red retenga información de eventos pasados como ayuda para evaluar eventos futuros. Cada vector de entrada presentado como entrenamiento para este tipo de redes es una entidad aislada del resto y, al final de dicho periodo de prueba, la red estará lista para comenzar a identificar y clasificar patrones o cualquier otra aplicación que se le quiera dar. (Haykin 1998).

Red neuronal para el control de humedad

Se creó una red neuronal tipo feedforward, como se muestra en la figura 2, para controlar la humedad del cultivo hidropónico contando con 5 neuronas de entradas en la primera capa, 4 neuronas en la capa oculta y una capa de salida con 1 neurona.

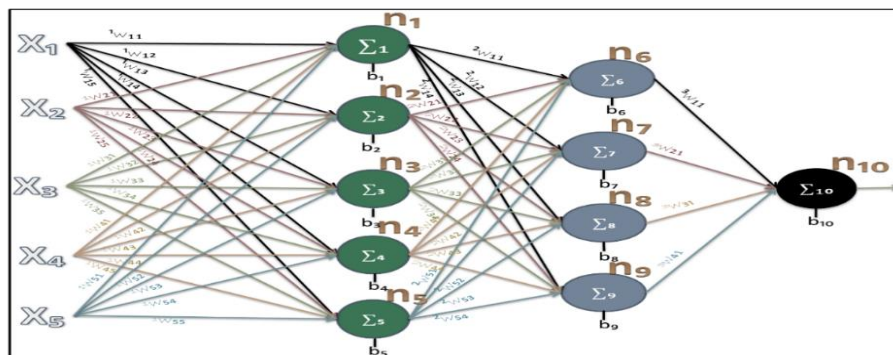


Figura 2. Red neuronal tipo feedforward [5 4 1].

La red neuronal artificial fue creada con el programa MATLAB, utilizando el paquete de la “Neural Network Toolbox”, que contiene una serie de funciones para crear y trabajar con redes de neuronas artificiales que comúnmente se entrenan por aprendizaje supervisado, aunque también soporta el aprendizaje no supervisado y el diseño directo, en MATLAB se realizó el proceso de aprendizaje, entrenamiento y las funciones de transferencia.

Para los valores de entrada de la red neuronal se utilizaron 5 sensores de humedad FC-28, los valores de los sensores de humedad fueron establecidos en la siguiente tabla.

Tabla 1. Escala de valores de entrada y salida de la red neuronal.

No	Valores de entrada	Valores de salida	Observaciones
1	1023	100	Seco
2	1000	100	Seco
3	900	100	Seco
4	800	75	Seco
5	700	75	Seco
6	600	50	Seco
7	500	50	Optimo
8	400	30	Optimo
9	300	30	Optimo
10	200	0	Húmedo
11	100	0	Húmedo
12	0	0	Húmedo

Con los datos establecidos se codificó en MATLAB la red neuronal, realizando el proceso de entrenamiento, aplicando el algoritmo backpropagation y utilizando las funciones de transferencia para generar los valores de entrada a la lógica difusa.

Lógica difusa.

Los sistemas lógicos difusos son sistemas basados en reglas, las cuales son expresadas como implicaciones lógicas, es decir, en forma de sentencias SI-ENTONCES. La implicación refleja la relación que guarda un hecho derivado de otro y pertenece a una rama de las matemáticas conocida como lógica (Chahuara, 2005).

La lógica difusa permite tratar información imprecisa, como estatura media o temperatura baja, en términos de conjuntos borrosos que se combinan en reglas para definir acciones: si la temperatura es alta entonces enfriar mucho. De esta manera, los sistemas de control basados en lógica difusa combinan variables de entrada, definidas en términos de conjuntos difusos, por medio de grupos de reglas que producen uno o varios valores de salida. Los más populares sistemas de lógica difusa que se encuentran son los siguientes tipos: sistemas difusos tipo Mamdani (con fuzzificador y defuzzificador) Sistemas difusos tipo Takagi-Sugeno.

Sistema difuso tipo Takagi-Sugeno.

El modelo difuso de Takagi-Sugeno fue propuesto por Takagi, Sugeno y Kang, en un esfuerzo para formalizar un método sistemático para generar reglas difusas a partir de un conjunto de datos de entradas y salidas.

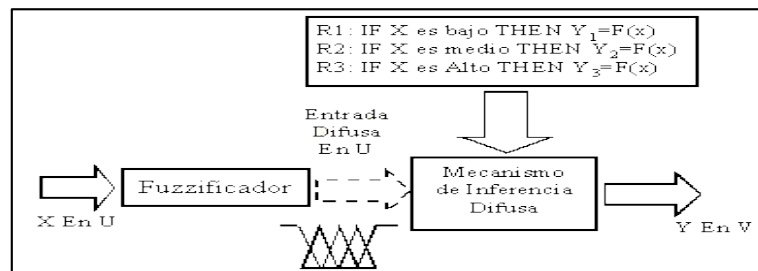


Figura 3. Sistema difuso Sugeno procesamiento general.

Los sistemas difusos Sugeno, como se muestra en la figura 3, se distinguen las siguientes partes:

- Fuzzificador, la entrada de un sistema de lógica difusa tipo Sugeno normalmente es un valor numérico proveniente, por ejemplo, de un sensor; para que este valor pueda ser procesado por el sistema difuso se hace necesario convertirlo a un "lenguaje" que el mecanismo de inferencia pueda procesar. Esta es la función del fuzzificador, que toma los valores numéricos provenientes del exterior y los convierte en valores "difusos" que pueden ser procesados por el mecanismo de inferencia. Estos valores difusos son los niveles de pertenencia de los valores de entrada a los diferentes conjuntos difusos en los cuales se ha dividido el universo de discurso de las diferentes variables de entrada al sistema.
- Mecanismo de inferencia difusa, teniendo los diferentes niveles de pertenencia arrojados por el fuzzificador, los mismos deben ser procesados para generar una salida difusa. La tarea del sistema de inferencia es tomar los niveles de pertenencia y apoyado en la base de reglas generar la salida del sistema difuso.
- Base de reglas difusas, en el sistema difuso tipo Sugeno, los valores que arrojan los consecuentes de las diferentes reglas que se han activado en un momento determinado ya son valores numéricos por lo que no se necesita una etapa de defuzzificación. Para calcular la salida del sistema difuso se ponderan los diferentes consecuentes teniendo en cuenta el valor que se activó el antecedente de cada una de las reglas.



Figura 4. Bases de reglas difusas.

3. Desarrollo.

El desarrollo del proyecto se dividió en 4 etapas y se utilizó un modelo de lazo cerrado (figura 5), con una entrada de referencia, aplicando el sistema neurodifuso (red neuronal y lógica difusa) para tener una salida controlada, con el propósito de retroalimentar el sistema.

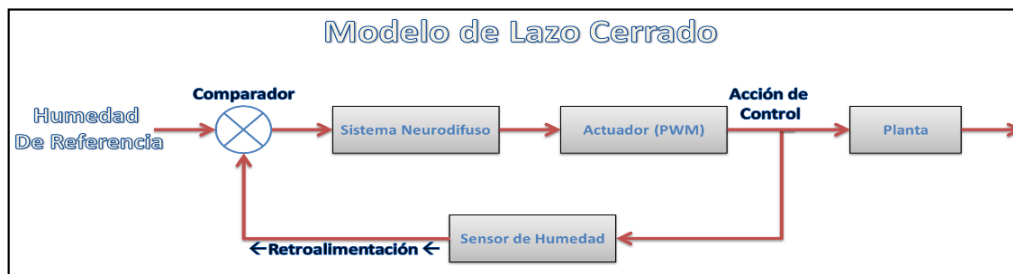


Figura 5. Modelo de lazo cerrado.

1.- Creación de la red neuronal artificial en Matlab.

- a) Se crea la variable “h” (humedad) que almacena los valores de las 5 entradas de los sensores de humedad, los cuales reciben valores desde 0 = muy húmedo, hasta 1023 = muy seco.
- b) Crear la variable “hs” es el porcentaje de salida, con valores entre 0 a 100.
- c) Se crea la red neuronal “net” agregando la matriz con los valores mínimos y máximos, tamaño de la capa, funciones de transferencia y entrenamiento.

```
>> h=[1023 1000 900 800 700 600 500 400 300 200 100 0; 1023 1000 900 800 700 600 500 400 300 200 100 0;
1023 1000 900 800 700 600 500 400 300 200 100 0; 1023 1000 900 800 700 600 500 400 300 200 100 0;
1023 1000 900 800 700 600 500 400 300 200 100 0];
>> hs = [100 100 100 75 75 50 50 30 30 0 0 0];
>> net = newff(minmax(h),[5 4 1],{'tansig','tansig','purelin'},'trainlm');
```

Figura 6. Creación de la red neuronal.

- d) Se inicializa la red “net” mediante el comando “init”.
- e) Se agrega el error máximo permitido.
- f) Se procede a entrenar la red “net” con el comando “train”, con lo que MatLab entrenara la red hasta alcanzar el performance esperado.

```
>> net = init(net);
>> net.trainparam.goal = 1e-05;
>> net = train(net, h, hs);
```

Figura 7. Inicialización y entrenamiento de la red neuronal.

- g) Simulación de la red neuronal, una vez entrenada la red neuronal, podemos aplicar un patrón real a la entrada y comprobar en la salida los valores que nos proyecta.

```
>> output = sim(net,h)

output =

Columns 1 through 11

55.4417 55.4384 55.4084 55.3052 54.9394 54.0435 52.9038 52.0429 51.5570 51.3217 51.2160

Column 12

-0.0035
```

Figura 8. Simulación de la red neuronal para comprobar los resultados.

- h) Por último procedemos a obtener los pesos y bias de cada una de las capas de la red neuronal con los comandos “net.iw{1,1} y net.b{1}” respectivamente. El mismo procedimiento se aplica para las capas 2 y 3, por lo que no se explica.

```
>> net.iw{1,1}

ans =

0.1383 0.1401 0.1390 0.1418 0.1406
0.0580 0.0625 0.0626 0.0616 0.0607
0.0013 -0.0019 0.0002 -0.0003 -0.0016
0.0685 0.0664 0.0704 0.0666 0.0674
0.1667 0.1638 0.1668 0.1660 0.1626

>> net.b{1}

ans =

0.0211
2.6979
1.4129
0.2235
1.6849
```

Figura 9. Pesos y bias de la capa 1.

2.- Desarrollo de lógica difusa aplicando el modelo Takagi-Sugeno, debido a que ya están definidas las salidas del voltaje en relación a la entrada, estableciendo los valores definidos en la tabla 1 de la escala de control de humedad, se generan 3 subconjuntos trapezoidales.

- a) Se ejecuta el comando “Fuzzy” en Matlab para iniciar la interfaz gráfica y se establecen los valores de entrada, que estarán en un rango de 0 a 100.

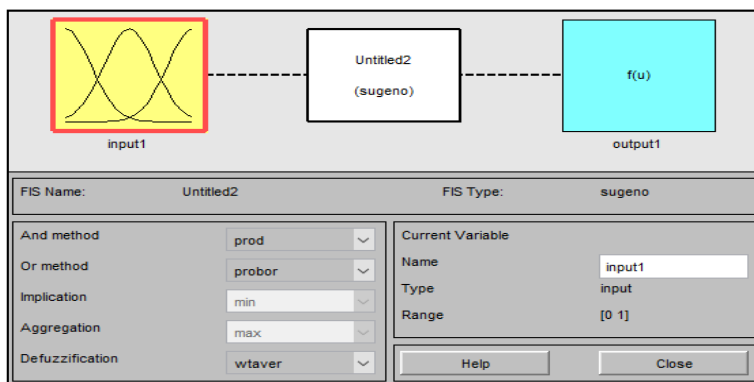


Figura 10. Interfaz gráfica en MatLab modelo Sugeno.

- b) Creación del primer subconjunto difuso “Humedo” con un rango de $[-30 0 10 30]$, función trapmf.

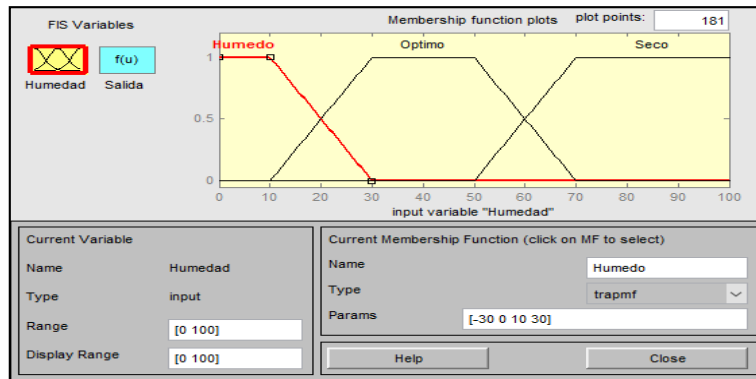


Figura 11. Subconjunto trapezoidal Humedo de la lógica difusa.

- c) Creación del segundo subconjunto difuso “Óptimo” con un rango de $[10 30 50 70]$, función trapmf.

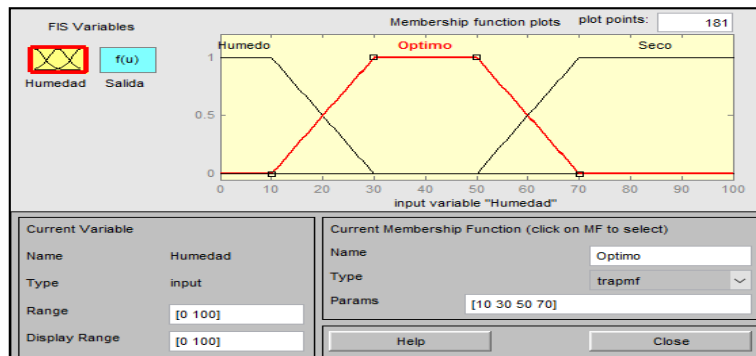


Figura 12. Subconjunto trapezoidal Óptimo de la lógica difusa.

- d) Creación del primer subconjunto difuso “Seco” con un rango de $[50 70 100 120]$, función trapmf.

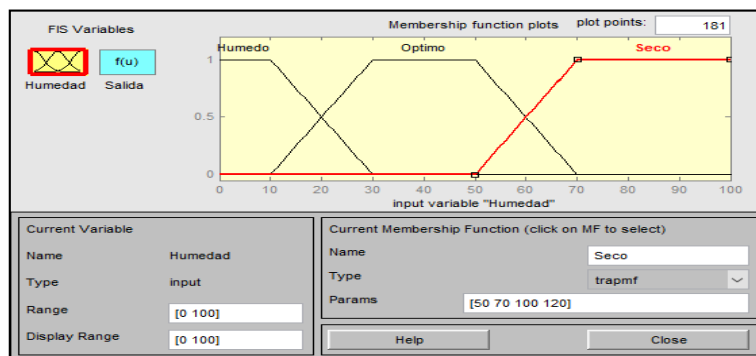


Figura 13. Subconjunto trapezoidal Seco de la lógica difusa.

- e) Se asignan los siguientes valores a la salida de lógica difusa Sugeno (estos valores serán la salida para el PWM del código en Arduino, debido a que el circuito creado utiliza estos valores para hacer funcionar a la bomba sumergible):

- Cuando la humedad es BAJA se activara la señal del PWM con un salida de 150.
- Cuando la humedad es MEDIA se activara la señal del PWM con una salida de 50.
- Cuando la humedad es ALTA se activara la señal del PWM con una salida de 0.



Figura 14. Variables de salida modelo Sugeno.

- f) A continuación se mencionan las reglas de inferencia utilizadas en el sistema difuso:
- Si la humedad es ALTA entonces la salida es BAJA.
 - Si la humedad es OPTIMA entonces la salida es MEDIA.
 - Si la humedad es BAJA entonces la salida es ALTA.

3.- Desarrollo de código del sistema neurodifuso en el IDE Arduino con los valores obtenidos en la red neuronal y en la lógica difusa, y al finalizar su respectiva quema en una placa Arduino.

- a) Declaración de variables:
- Se declaran cinco variables para recibir los datos de los cinco sensores de humedad FC-28.
 - Se declaran diez variables para las neuronas de la red neuronal.
 - Se declaran nueve variables para aplicar las funciones de transferencia.
 - Se declaran tres variables con la función de pertenencia tipo trampf.
 - Se declaran tres variables más para la salida del PWM (la intensidad de corriente que recibirá la bomba).
 - Se declara una variable para la constante “e”.
- b) Los resultados de salida obtenidos de la red neuronal, son los valores de entrada que se usan para el proceso de la lógica difusa, todo esto codificado en el IDE Arduino, a continuación de muestra una parte del desarrollo.
- Se leen los puertos analógicos, asignando este valor a cada variable perteneciente a cada sensor.
 - Se asignan los valores de pesos y bias a cada neurona de nuestra red en específico de nuestras tres capas totales de la red neuronal.
 - Se aplica la función de transferencia a cada neurona de nuestras tres capas, iniciando en la primer capa, en seguida a la segunda capa y al final a la tercer capa (capa de salida).
 - Se prosigue con el desarrollo de la parte de lógica difusa, que nos generará una salida por el pin 9 (PWM) de nuestra placa Arduino, esta última salida generada por el sistema será la encargada de activar y/o desactivar la bomba de agua, encargada del riego del cultivo.

```

Sistema_Neurodifuso $
#include <math.h> //Libreria para operaciones matematicas
float h1, h2, h3, h4, h5; // Variables de los sensores de humedad
float n1, n2, n3, n4, n5, n6, n7, n8, n9, n10; //Variables de las neuronas de la red
float s1, s2, s3, s4, s5, s6, s7, s8, s9; // Variables de las funciones de transferencia
float p_bajo, p_normal, p_alto, v; //Variable de las potencias
const int v1 = 0, v2 = 50, v3 = 150; //Salida del PWM
float e = 2.71828182;

void setup() {
  Serial.begin(9600);
}

void loop() {
  h1 = analogRead(0); //Se leen los valores de entrada
  h2 = analogRead(1);
  h3 = analogRead(2);
  h4 = analogRead(3);
  h5 = analogRead(4);
  
```

Figura 15. Desarrollo en el IDE Arduino.

4.- Implementación de todos los componentes para la puesta en marcha del sistema.

- a) Se utilizó un contenedor con medidas de 138.5 cm de largo, 78.5 cm de ancho y 30 cm de profundidad.
- b) Se utilizó una mezcla de peat moss con arena, en cantidades de 70% y 30% respectivamente como sustrato para nuestro cultivo, colocado en el contenedor.
- c) Se realizó una práctica de laboratorio para conocer la capacidad de retención de agua (CRA) presente en el sustrato.
- d) Se realizó la siembra en charola de las semillas de tomate, para posteriormente de 12 a 15 días realizar el trasplante de las plántulas al contenedor.
- e) Instalación de los 5 sensores de humedad Fc-28 a lo largo y ancho del contenedor.

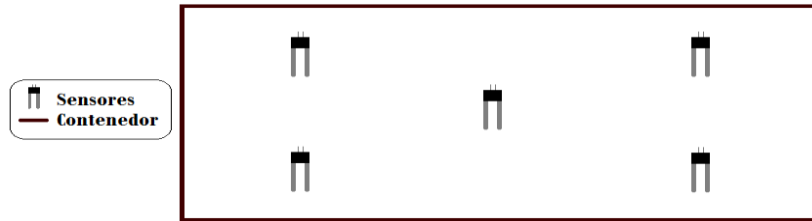


Figura 16. Esquema de distribución de los sensores de humedad FC-28.

- f) Instalación del sistema de riego, utilizando 8 metros de manguera de cristal de 3/4" y para obtener un riego más uniforme sin perder potencia en la salida de agua, se adaptó un empalme para conectar tres partes de la manguera y así poder distribuirse mejor por el contenedor.

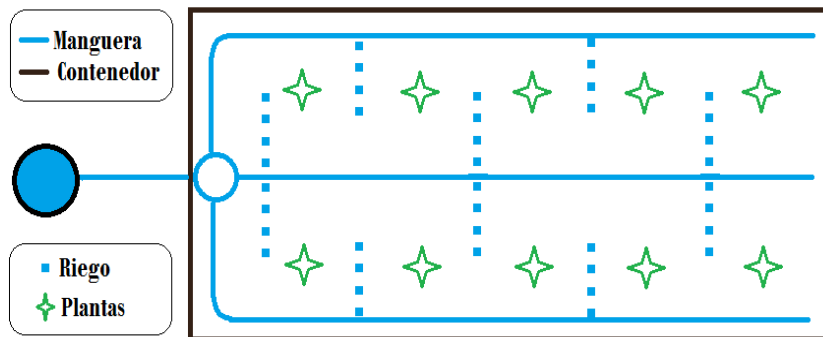


Figura 17. Esquema de distribución del sistema de riego.

- g) Se realizó un circuito necesario para el funcionamiento de la bomba.

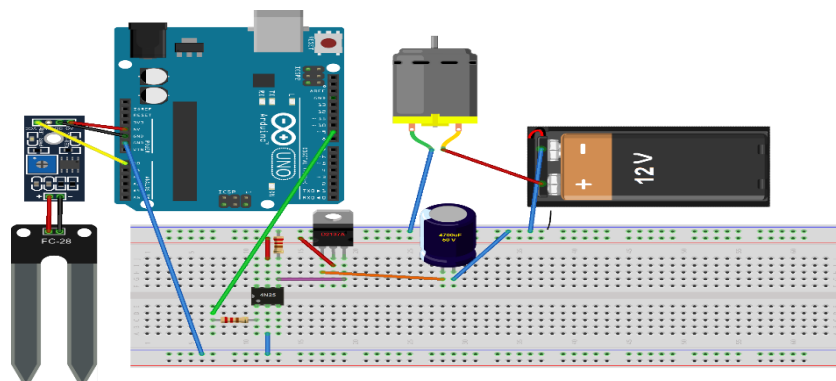


Figura 18. Diagrama esquemático de las conexiones del circuito.

- h) La bomba de agua fue colocada en un recipiente con una capacidad de más de 10 litros de agua, la bomba debe ir conectada a una alimentación de 12v, y su entrada negativa a la salida del circuito, el cual su valor es dado por la salida del pin 9 (PWM) del Arduino.
- i) Se alimentó a la placa Arduino con una power bank (batería externa) de 3000 mAh, que cuenta con un puerto USB, donde se conectó el Cable USB Tipo A-B, la cual lo alimenta correctamente para su funcionamiento.
- j) Una vez que el sistema cuente con alimentación necesaria, 5v para la placa Arduino y 12v para la bomba, el sistema comenzara a trabajar autónomamente, censando y activando la bomba según la necesidad del suelo.

Conclusiones.

El sistema neurodifuso aplicado al cultivo hidropónico donde se determinaron los valores de 40% a 60% de humedad como óptimos, cumple con la característica de mantener dichos valores en los rangos establecidos, los valores de entrada del sistema neurodifuso son tomados de las mediciones que realizan los sensores de humedad en tiempo real, y el valor de salida que activa el riego automático, es obtenido a través del valor de salida del mismo sistema. La red neuronal nos permitió encontrar la combinación de parámetros que mejor se ajustan al control de humedad requerida, en la logica difusa se generaron las reglas para realizar las acciones de control para la activación del riego, de acuerdo a los valores establecidos.

Créditos.

Los autores agradecen y dan créditos al Instituto Tecnológico de Tuxtla Gutiérrez por todo el apoyo y las facilidades para la realización de este proyecto.

Referencias Bibliográficas.

Chahuara, J. (2005). Sistemas Neuro-difusos, Virtual data, Recuperado 25 de Agosto del 2016, desde <http://sisbib.unmsm.edu.pe/bibvirtualdata/tesis/ini/chahuara.htm>

Haykin, S. (1998). Neural Networks, A comprehensive Foundation, New York: MacMillan College Publishing Company.

Maroto B., J. V. 1989. Horticultura Herbácea Especial, Editorial Mundi-Prensa, Madrid, España. 589 p.

Resh, H. M. 2002. Cultivos Hidropónicos, Ediciones Mundi-Prensa, Madrid, España. 369 p.

Romero, A. (2007). Redes Neuronales, Gestión de recursos Informáticos del Departamento de Informática y Automática. Recuperado el 26 de Agosto de 2016, desde <http://avellano.fis.usal.es/~lalonso/RNA/index.htm>

Sampeiro, G. 2014. "México Puede Solucionar El Déficit De Producción De Alimentos Mediante El Sistema De Hidropónia". Asociación Hidropónica Mexicana. Recuperado el 9 de Agosto de 2016, desde <http://hidroponia.org.mx>

Información de los autores.



Mario Culebro de la Piedra es egresado del Instituto Tecnológico de Tuxtla Gutiérrez en diciembre de 2015, de la carrera de Ingeniería en Sistemas Computacionales, se especializa en el área de inteligencia artificial con enfoque a los sistemas neurodifusos (sistemas de control), desarrollo de proyectos web y sistemas embebidos.



Marcos Rafael Gómez Gómez es egresado del Instituto Tecnológico de Tuxtla Gutiérrez en diciembre de 2015, de la carrera de Ingeniería en Sistemas Computacionales, se especializa en el área de inteligencia artificial en la rama de sistemas expertos orientado a los sistemas neurodifusos, desarrollo de aplicaciones móviles y proyectos web.



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, actualmente es perfil deseable y trabaja en la línea de investigación Tecnología de Información y Base de Datos.



Joaquín Adolfo Montes Molina graduado en la Licenciatura en Químico Farmacéutico Industrial por la Escuela Nacional de Ciencias Biológicas en 1976. Graduado como Maestro en Ciencias en Biotecnología por la Universidad Autónoma de Chiapas (UNACH). Graduado como Doctor en Ciencias en la Especialidad de Biotecnología por el Centro de Investigación y Estudios Avanzados (CINVESTAV). Es miembro del Sistema Nacional de Investigadores nivel 1. Actualmente es docente-investigador en el Instituto Tecnológico de Tuxtla Gutiérrez desde 1976 a la fecha. Su área de interés es la biotecnología.



Néstor Antonio Morales Navarro es Maestro en Ciencias en Ingeniería Mecatrónica por el Instituto Tecnológico de Tuxtla Gutiérrez en 2010. Es profesor en el área de Ingeniería en Sistemas Computacionales del Instituto Tecnológico de Tuxtla Gutiérrez desde 2012 y en el área de Licenciatura en Ingeniería Mecatrónica de la Universidad Valle de México Campus Tuxtla desde 2011. Se especializa en el área de Visión e Inteligencia Artificial.



Jorge Octavio Guzmán Sánchez tiene la Maestría en Ciencias de la Computación, especialidad bases de datos y sistemas de información, es Ingeniero en Sistemas Computacionales, profesional certificado por *Microsoft* en la administración de servidores con *Windows*. Ejerce la docencia desde hace más de una década, actualmente docente del Instituto Tecnológico de Tuxtla Gutiérrez así como en la Universidad Descartes.

