

SEP

TECNM

DITD



---

---

# INSTITUTO TECNOLÓGICO SUPERIOR DE ATLIXCO

*Organismo Público Descentralizado del Gobierno del Estado de Puebla*

## DISEÑO E IMPLEMENTACIÓN DE UN SISTEMA DE INFERENCIA DIFUSO EN VHDL CON APLICACIÓN AL CONTROL DE TEMPERATURA EN UN INVERNADERO DE JITOMATE

OPCIÓN I.

TESIS PROFESIONAL

QUE PARA OBTENER EL TÍTULO DE:

**INGENIERO MECATRÓNICO**

PRESENTA:

**ANA MARÍA RODRÍGUEZ GONZÁLEZ**

**ASESOR: DRA. MARIANA NATALIA IBARRA BONILLA**

ATLIXCO, PUE. MARZO DE 2019

<b>INTRODUCCIÓN .....</b>	<b>3</b>
<b>CAPÍTULO 1. PROTOCOLO DE INVESTIGACIÓN.....</b>	<b>5</b>
1.1 SITUACIÓN PROBLEMÁTICA .....	5
1.2 PLANTEAMIENTO DEL PROBLEMA.....	5
1.3 OBJETIVOS.....	6
1.3.1 OBJETIVO GENERAL .....	6
1.3.2 OBJETIVOS PARTICULARES.....	6
1.4 HIPÓTESIS.....	6
1.5 JUSTIFICACIÓN.....	6
1.6 ALCANCES Y LIMITACIONES.....	8
<b>CAPÍTULO 2. MARCO TEÓRICO .....</b>	<b>9</b>
2.1 INVERNADEROS .....	9
2.1.1 CONTROL CLIMÁTICO .....	11
2.1.2 DISPOSITIVOS PARA EL CONTROL CLIMÁTICO.....	13
<b>CAPÍTULO 3. LÓGICA DIFUSA.....</b>	<b>16</b>
3.1 CONCEPTOS BÁSICOS DE LÓGICA DIFUSA.....	18
3.1.1 CONJUNTOS DIFUSOS .....	18
3.1.2 FUNCIÓN DE MEMBRESÍA (PERTENENCIA).....	20
3.2 SISTEMAS DE CONTROL DIFUSO (BORROSO) .....	21
3.2.1 FUZIFICADOR (BORROSIFICADOR) .....	22
3.2.2 BASE DE REGLAS .....	22
3.2.3 INFERENCIA .....	22
3.2.4 DEFUZIFICADOR (DEBORROSIFICADOR) .....	23
3.3 HERRAMIENTAS .....	24
3.3.1 MATLAB.....	24
3.3.2 XFUZZY .....	26
<b>CAPÍTULO 4. FPGA (FIELD- PROGRAMMABLE GATE ARRAY) .....</b>	<b>28</b>
4.1 VHDL .....	28
4.1.1 ESTRUCTURA.....	29
4.2 VERILOG .....	30
<b>CAPÍTULO 5. METODOLOGÍA.....</b>	<b>32</b>
5.1 DISEÑO DEL SISTEMA DIFUSO.....	32
5.1.1 VARIABLES LINGÜÍSTICAS .....	32

5.1.2 VALORES LINGÜÍSTICOS (ENTRADAS) .....	32
5.1.3 VALORES LINGÜÍSTICOS (SALIDA) .....	32
5.1.4 MATRIZ ASOCIATIVA DIFUSA (FAM) .....	33
5.2 IMPLEMENTACIÓN DEL SISTEMA DIFUSO EN MATLAB .....	33
5.2.1 SIMULINK .....	35
5.3 IMPLEMENTACIÓN DEL SISTEMA DIFUSO EN XFUZZY .....	36
5.4 IMPLEMENTACIÓN DEL SISTEMA DIFUSO EN FPGA (VHDL) .....	38
5.4.1 ETAPA DE FUZIFIZACIÓN .....	39
5.4.2 ETAPA DE INFERENCIA .....	40
5.4.3 ETAPA DE DEFUZIFIZACIÓN .....	41
5.5 IMPLEMENTACIÓN EN HARDWARE .....	42
5.6 RESULTADOS .....	44
<b>CONCLUSIONES .....</b>	<b>51</b>
<b>APÉNDICES .....</b>	<b>52</b>
APÉNDICE A. PROTOTIPO INVERNADERO AUTOMÁTICO .....	52
APÉNDICE B. PRUEBAS VHDL-MATLAB .....	53
<b>REFERENCIAS.....</b>	<b>58</b>

## INTRODUCCIÓN

El presente trabajo tiene como principal objetivo diseñar e implementar un controlador difuso y codificarlo en VHDL para su aplicación en el sector agrícola, específicamente para mantener una temperatura óptima para el cultivo bajo cubierta de jitomate.

Los principales motivos que me llevaron a enfocar esta tesis al diseño de un sistema difuso es el interés personal en la lógica difusa, así como lo referente a la inteligencia artificial y lo controversial de sus aplicaciones. La elección de implementarlo para el control de temperatura en un invernadero fueron los resultados obtenidos con un prototipo de un sistema de monitoreo y control de un invernadero de jitomate a través de una página web trabajado en los últimos semestres de la licenciatura (Ver Apéndice A), a continuación se mencionan los eventos en que se presentó dicho proyecto:

- II Demostración de Innovación Tecnológica y Emprendimiento 2018 (Etapa Regional).
- Sexto encuentro de Jóvenes investigadores 2018.
- XVII Congreso Nacional de Ingeniería Electromecánica y de Sistemas (CNIES 2018) en la modalidad Artículo (Aceptado).

Este trabajo describe el proceso de diseño del controlador, desde la definición de valores y variables lingüísticas hasta la codificación en VHDL. Asimismo realiza una comparación entre los valores de salida del controlador dados por una FPGA y la simulación.

***Un invernadero es toda aquella estructura cerrada cubierta por materiales transparentes, dentro de la cual es posible obtener unas condiciones artificiales de microclima, y con ello cultivar plantas fuera de estación en condiciones óptimas [1].*** Esto último puede beneficiarse de la continua evolución de la ciencia y la tecnología, haciendo uso de sensores, actuadores, dispositivos programables (microcontrolador, PLC o FPGA) y diferentes técnicas de control.

Con el desarrollo tecnológico constante, la incapacidad de ser 100% precisos en sistemas cada vez más complejos se vuelve más evidente, por esto el

ingeniero Iraní-estadounidense Lotfy A. Zadeh introdujo el concepto de conjunto difuso con la conjetura que el pensamiento humano es cualitativo y no cuantitativo, es decir, se basa en etiquetas lingüísticas más no en números como la lógica clásica lo representa. La lógica difusa o Fuzzy logic en inglés, es capaz de ser flexible, tiene cierta tolerancia con la imprecisión y se basa en el lenguaje natural, por lo que la convierte en la herramienta matemática que puede ser aplicable en el campo de la automatización en el sector agrícola.

En el primer capítulo de este trabajo se presenta la caracterización del proyecto realizado, se define la situación problemática, el planteamiento del problema, la hipótesis, la justificación y los objetivos (general y particulares) los cuales se cumplieron al finalizar el proyecto.

En el capítulo dos se exponen las principales características de un invernadero y la clasificación según la estructura. También aborda las variables más relevantes a controlar dentro de un invernadero, la importancia del control climático, así como algunos dispositivos que se encuentran en el mercado actualmente utilizados para el mismo fin.

El capítulo tres introduce los conceptos necesarios sobre lógica difusa para comprender el sentido y alcance de este proyecto. Allí se presentará una breve descripción del origen de la teoría de conjuntos, definición de los conceptos básicos: conjunto difuso, función de membresía, operaciones con conjuntos, controlador difuso y sus diferentes etapas (fuzificación, inferencia y defuzificación).

En el capítulo cuatro se describe el proceso de diseño e implementación del sistema difuso. Primero se presenta la definición de los valores y variables lingüísticos utilizados dentro del sistema, también se describe la implementación en la Fuzzy toolbox de Matlab, Simulink y XFuzzy. Posteriormente se detalla la codificación en VHDL del sistema difuso implementado en los software anteriormente nombrados. Se concluye el capítulo haciendo al análisis y la interpretación de los resultados obtenidos a través de las pruebas físicas y de la simulación realizada.

# CAPÍTULO 1. PROTOCOLO DE INVESTIGACIÓN

## 1.1 SITUACIÓN PROBLEMÁTICA

El cultivo en invernáculos puede representar un gran desafío para un emprendedor inexperto, *los recursos disponibles de mano de obra para la industria de vegetales y frutas frescas se están volviendo cada vez más escasos. Solo en Estados Unidos la fuerza de trabajo agrícola ha caído desde 3.4 millones a un millón durante el último siglo y un 20% entre 2002 y 2014, y los costos han aumentado 51%. [2]*

Aunque la instalación de un invernadero conlleva una alta inversión inicial y costos de mantenimiento las ventajas que se pueden obtener son mayores, por mencionar algunas: mayor calidad en el producto, aumento del volumen de la cosecha por año y mejor control del estado del cultivo.

Entre los elementos que se pueden controlar dentro de un invernadero son:

- Riego.
- Temperatura.
- Humedad relativa.
- Dióxido de Carbono.
- Iluminación.

La temperatura y humedad relativa son los parámetros físicos que pueden ser más variantes, ya que las condiciones ambientales del exterior tienen la capacidad de interferir en el microclima dentro del invernadero y afectar parcial o totalmente el cultivo si no se tiene un control climático adecuado.

La temperatura es uno de los parámetros físicos que más impacta en el desarrollo de un cultivo, puede interferir en el proceso de fotosíntesis, permeabilidad de la membrana celular y en la absorción de agua y nutrientes, teniendo como consecuencia un desarrollo y crecimiento deficiente.

## 1.2 PLANTEAMIENTO DEL PROBLEMA

¿Cómo mantener un control adecuado de la temperatura dentro de un invernadero de jitomate con las nuevas tecnologías?

## **1.3 OBJETIVOS**

### **1.3.1 OBJETIVO GENERAL**

Diseñar un controlador difuso tipo Mamdani para mantener las condiciones climáticas óptimas dentro de un invernadero de jitomate e implementarlo en un FPGA Spartan6.

### **1.3.2 OBJETIVOS PARTICULARES**

- Definir los parámetros de entrada y salida del controlador, así como las condiciones apropiadas para el cultivo de jitomate dentro del invernadero.
- Diseñar un controlador difuso por el método de Ebrahim Mamdani.
- Implementar el diseño en MATLAB y Simulink para la visualización gráfica del comportamiento del controlador.
- Codificar el sistema difuso en lenguaje VHDL (Very High Speed Integrated Circuit Hardware Description Language) para su implementación en una FPGA Spartan6.
- Realizar pruebas físicas y de simulación para la validación del controlador.

## **1.4 HIPÓTESIS**

El diseño de un algoritmo basado en lógica difusa en una tecnología FPGA asegura un control preciso que requiere operación en tiempo real.

## **1.5 JUSTIFICACIÓN**

La optimización de invernaderos ha permitido el cultivo de diversas especies de flora durante todo el año, en lugares y climas que no serían aptos con un cultivo en el exterior.

La agricultura ha sido una de las actividades más importantes en México, actualmente los principales cultivos de consumo interno y exportación son de maíz, trigo, aguacate y jitomate. En lo que respecta al jitomate, México ocupa el primer lugar en cuanto a exportación a nivel continental, gran parte de esto se debe al estado de Sinaloa que tiene la mayor producción a nivel nacional y a nivel mundial ocupa el octavo lugar, abasteciendo principalmente a parte de México, Canadá y Estados Unidos.

Existen tres variables importantes dentro de un invernadero para asegurar un desarrollo y crecimiento óptimo del jitomate que son: la temperatura ambiental y de suelo, la humedad relativa y el porcentaje de dióxido de carbono. Dentro de éstos la temperatura se considera el factor ambiental más relevante para dar paso a las reacciones biológicas que tienen mayor repercusión en el desarrollo, maduración, crecimiento y coloración del producto.

Por otro lado la humedad relativa influye directamente con el crecimiento de la planta, es decir, cada especie tiene un umbral en el que se puede desarrollar. En niveles altos puede ser un factor de riesgo para el cultivo completo ya que favorece la propagación de hongos, enfermedades, plagas e incluso puede verse dañada la estructura del invernadero.

Para obtener un correcto control de clima en agricultura protegida se acude a la implementación de sistemas inteligentes que logren evitar errores humanos, sean autónomos y principalmente incrementen la calidad y cantidad de producción; así mismo cumpliendo con tres factores que harán al controlador un artículo factible para el productor que son: mantener el clima según las exigencias del cultivo, costo accesible y materiales resistentes a las condiciones de un invernadero.

Un invernadero es un sistema que requiere operación en tiempo real, y la implementación del controlador difuso requiere del procesamiento de muchas operaciones, por lo que se necesita de una tecnología que sea capaz de integrar una gran cantidad de operaciones lógicas en un solo circuito integrado. El uso de un sistema inteligente con la tecnología de una FPGA trae como beneficios, tener un dispositivo flexible fácilmente reprogramable y rápido por

realizar tareas paralelizadas, lo que lo hace rentable para proyectos enfocados a la automatización y control de sistemas complejos.

## **1.6 ALCANCES Y LIMITACIONES**

### **ALCANCES**

El alcance de este proyecto está previsto para obtener el controlador difuso en una implementación en hardware Aunque la elaboración del proyecto va dirigido para el cultivo de jitomate, el sistema está diseñado para ser flexible y pueda ser fácilmente modificado dependiendo de las especificaciones y variables que quiera controlar y monitorear.

### **LIMITACIONES**

No llegar a la implementación. Las pruebas de evaluación del desempeño del controlador no son realizadas en un invernadero real.

## CAPÍTULO 2. MARCO TEÓRICO

### 2.1 INVERNADEROS

Un invernadero o invernáculo se define como un recinto en el que se mantienen condiciones ambientales adecuadas para favorecer el cultivo de plantas, según la Real Academia Española (RAE). Generalmente la estructura es semicilíndrica, cubierta por plástico grueso o incluso vidrio para proteger a las plantas de las condiciones exteriores (Ver Figura 2.1).



Figura 2.1. Invernadero.

Existen diferentes factores a considerar para elegir el tipo de invernadero que es más conveniente construir, por ejemplo:

- Exigencias bioclimáticas específicas del cultivo.
- Mano de obra y disponibilidad económica
- Radiación solar.
- Características de la zona.

Este último elemento suele pasar desapercibido pero es uno de los más importantes; ya que se debe considerar el tipo de suelo, inclinándose por suelos con buen drenaje y de alta calidad, la dirección, intensidad y velocidad de los vientos en la zona, así mismo la topografía, ya que son más favorecedores los lugares con una pequeña pendiente orientados de norte a sur, es decir, un invernadero en un pie de loma, junto a arroyos o ríos no es aconsejable puesto que el terreno puede ser anegadizo en algunas épocas del año.

El material de la cobertura puede cambiar según los puntos antes mencionados, se puede utilizar polietileno, cloruro de polivinilo, vidrio o policarbonato. Igualmente la estructura puede ser de materiales variados, como: madera, hormigón, acero, aluminio u otro metal.

Los invernaderos pueden clasificarse según la forma de su estructura:

- De cristal o tipo Venlo: este invernadero industrial es de estructura metálica prefabricada con cubierta de vidrio.
- Túnel o semicilíndrico: es la forma más común por su capacidad de tener un mejor control de factores climáticos, rápida instalación y su resistencia a vientos fuerte. Su estructura es metálica.
- Su techo tiene la facilidad de desalojar el agua de lluvia puesto que está formado por uno o dos planos inclinados, su construcción es fácil.
- Doble capilla: se conforma por dos naves yuxtapuestas con aberturas entre estas con malla mosquitera para una mejor ventilación.
- Plano o tipo Parral: Cuenta con dos estructuras, una vertical con soportes rígidos y una horizontal que cuenta con dos mallas de alambre galvanizado. Esta estructura se aconseja usar en zonas poco lluviosas.
- Asimétrico: Busca la inclinación suficiente que permita que la radiación solar caiga sobre la cubierta al mediodía. Se orienta en sentido este-oeste.

## 2.1.1 CONTROL CLIMÁTICO

Es bien sabido que la temperatura es uno de los factores principales causantes de plagas y enfermedades en plantas. *Jingyuan Kia destacó que las pestes ya son responsables actualmente de la pérdida del 8% de los rendimientos del maíz del 14% de los del arroz a nivel mundial, porcentajes que crecerán al 10% y al 17%, respectivamente, si la temperatura global sube en dos grados centígrados respecto a los niveles preindustriales. [1]* El uso de invernaderos en el sector agrícola permite obtener productos de alta calidad en comparación a los procesos tradicionales en campo abierto. El control ambiental en invernáculos se concentra en manejar todos los dispositivos instalados (calefacción, ventilación, sistema de riego, suministro de fertilización, etc.) según las necesidades específicas del cultivo, así asegurando el buen rendimiento, precocidad sin perder de vista la calidad del producto.

Las condiciones bioclimáticas dentro de un invernadero deben de ser cuantificadas en cuatro periodos a lo largo del día para conocer el posible impacto sobre el desarrollo de las plantas, estos intervalos son: periodo diurno, periodo nocturno y sus dos transiciones entre ambos.

### 2.1.1.1 Principales factores a controlar dentro de un invernadero

Es esencial que el productor agrícola conozca las condiciones bioclimáticas necesarias de su cultivo para que éste no se vea afectado (Ver Figura 2.2), uno de los factores más importantes a controlar dentro de un invernadero es la temperatura, ya que influye en el proceso de fotosíntesis lo que afecta en la calidad de la planta e incluso se pueden presentar pérdidas en la producción. Es de gran importancia conocer los requerimientos del cultivo, algunos de los conceptos que nos pueden auxiliar para mantener un control óptimo son:

- Temperatura mínima letal: los cultivos pueden presentar daños severos en su fisiología.
- Temperatura nocturna y diurna: rangos aconsejados para asegurar un buen desarrollo del cultivo.

- Temperatura máxima y mínima biológica: valor donde alguna fase del desarrollo del cultivo (floración, fructificación, etc.) se puede ver afectada.

El control de la temperatura de un cultivo de jitomate se podría considerar como un factor de gran repercusión, ya que este cultivo en especial tiene requerimientos climáticos bastante altos en comparación con otros, por ejemplo:

- Temperatura diurna sobre 20-30°C.
- Temperatura nocturna sobre los 15-17°C.
- Temperatura mínima vegetativa de 12°C.
- Temperatura máxima vegetativa de 35°C
- Temperatura óptima para la fecundación entre 12°C y 25°C.

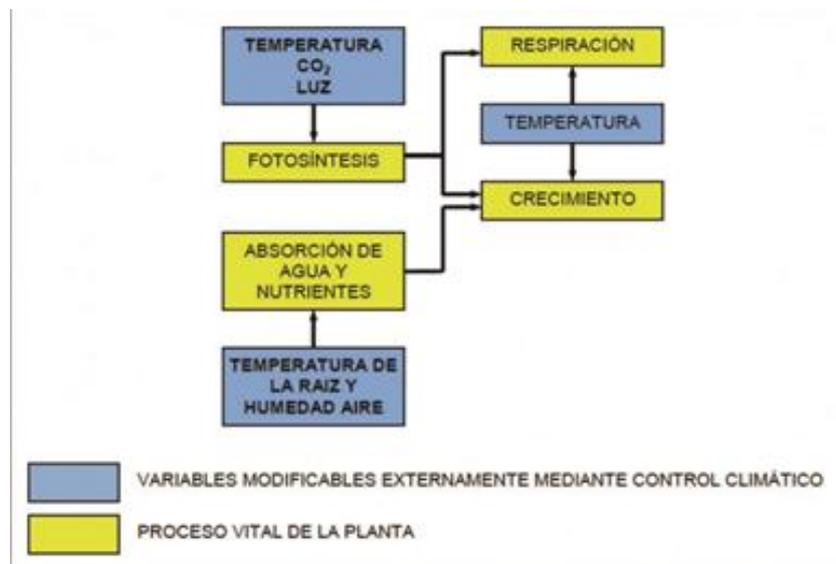


Figura 2.2. Variables importantes dentro de un invernadero.

La humedad relativa es otra variable a controlar dentro de un invernadero, se define como el porcentaje de agua presente en el aire en relación a la máxima que podría tener a la misma temperatura. Al igual que la temperatura influye en el proceso de cultivo y tiene una estrecha relación con ésta.

Si se alcanza entre el 80 y 85% el cultivo puede verse dañado, específicamente para el jitomate se necesita mantener un rango entre 60-80%, no mantener la humedad en un rango apropiado puede ocasionar daños severos, como: crecimiento de hongos, enfermedades, plagas, daño en la estructura del invernadero, baja tasa fotosintética, cierre de estomas y en casos de campo abierto dificulta el proceso de polinización.

Por otro lado, *las plantas consumen dióxido de carbono en la reacción de fotosíntesis y lo combinan con agua para formar azúcares y oxígeno. La concentración de dióxido de carbono de las plantas y, por lo tanto, el nivel de dióxido de carbono debe monitorearse y controlarse para alcanzar un nivel óptimo de crecimiento. [3]*

Al igual que la temperatura y la humedad cada especie necesita diferentes grados de concentración de dióxido de carbono, las plantas pueden ser más sensibles a niveles inadecuados en comparación a los humanos y los daños que pueden presentar son quemaduras o un crecimiento deficiente. El valor óptimo en la mayoría de los cultivos es 1000 ppm (partes por millón), este valor es más fácil de mantener durante el verano únicamente ventilando el invernadero.

Todos estos factores pueden alterar el desarrollo de la planta, a continuación se mencionan algunas afecciones que puede presentar el jitomate.

- Pudrición radical: Evita la germinación y causa la muerte de la plántula por exceso de humedad y una temperatura de 12 a 17°C.
- Pudrición de cuello y raíz: Lesiones alargadas oscuras o negras.
- Marchitez: Color amarillento presente en los primeros frutos y caída de hojas por suelo contaminados por lluvias, trasplantes, semilla, agua de riego temperaturas de 21 a 35°C.
- Pudrición de la corona y raíz: Necrosis vascular en los tallos, pudrición seca en la raíz y numerosas lesiones. Causada por hongos y temperaturas de 15 a 20°C.

## **2.1.2 DISPOSITIVOS PARA EL CONTROL CLIMÁTICO**

En la actualidad ya existen diferentes dispositivos capaces de controlar las condiciones climáticas, a continuación se mencionan algunos:

Sistema de control de riego y clima de Metaliser: Esta empresa ofrece un conjunto de sensores de temperatura y humedad relativa que se instalan dentro del invernadero, los cuales activan los sistemas de ventilación y calefacción, además del control del riego. Si el cliente lo requiere el sistema cuenta con un software de apoyo para mostrarlas condiciones del invernadero de manera gráfica.

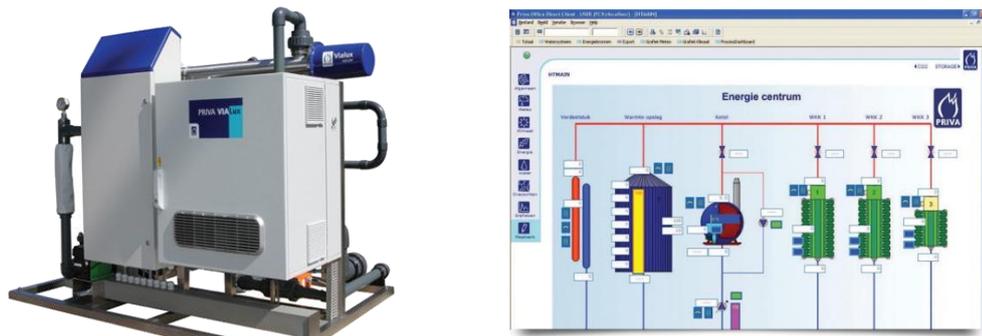


Figura 2.3. Sistema de control de riego y clima Metaliser

Calefactor y ventilador de Metaliser: Dispositivo que puede ser operado como calentador o recirculador de aire según las necesidades de invernadero.



Figura 2.4. Calefactor y ventilador de Metaliser.

Sistema de ventilación forzada de ECOFAN: Es un ventilador de costo accesible, fácil instalación y gran potencia.

Control climático de HARNOIS: Este equipo aparte de controlar los parámetros de la temperatura, es capaz de monitorear la lluvia, humedad y utilizar estos

valores para verificar las consecuencias en el proceso de crecimiento del cultivo.



Figura 2.5. Ejemplo de control climático.

PRIVA es una empresa internacional dedicada a la instalación de invernaderos y a la venta de equipos para el control de estos.

### CAPÍTULO 3. LÓGICA DIFUSA

Existen diversos métodos para atacar la incertidumbre en el modelado del conocimiento humano, el primer método del que se tiene registros es el razonamiento por defecto, un método cualitativo que valida sentencias hasta encontrar una razón para creer lo contrario. A principios del siglo XIX se introdujo uno de los primeros métodos numéricos, la probabilidad, se basa en el grado de creencia que se le asocia a una regla, este valor depende de la percepción del agente inteligente y las probabilidades pueden cambiar si el conocimiento de éste cambia. Algunos de las técnicas utilizadas son: redes bayesianas, diagramas de influencia, método Bayesiano subjetivo y factores de certeza.

La teoría de Dempster-Shafer es uno de los métodos no probabilistas que también utiliza grados de creencia pero por intervalos. ***La lógica difusa es igualmente definirse como una extensión de la lógica multivaluada que facilita enormemente el modelado de información cualitativa de forma aproximada.[4]*** Su gran ventaja ante otros métodos es la posibilidad de resolver problemas muy complejo.

Como indica Zadeh, cualquier problema puede resolverse con un conjunto de variables de entrada y obteniendo un valor adecuado de salida. En términos generales la lógica difusa hace posible esto a través del modelado enfocándose en valores lingüísticos. Este método también es llamada como lógica borrosa o teoría de conjuntos borrosos dada por el ingeniero eléctrico iraní-estadounidense, Lofti A. Zadeh.

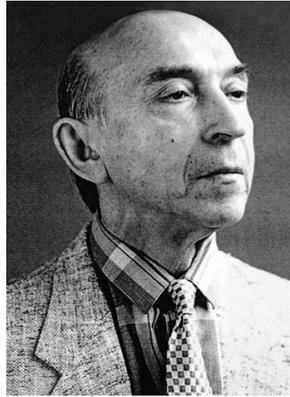


Figura 3.1. Lotfi A. Zadeh.

El primer ejemplo en que Zadeh utilizó el concepto de conjunto difuso fue para ilustrar “hombres altos”. Según la lógica clásica el conjunto hombres altos es un conjunto de hombres con una estatura mayor a cierto valor, por ejemplo 1.70 metros, y todos los hombres con una talla menor quedan fuera de dicho conjunto. Analizando más a fondo, si un hombre mide 1.71 metros pertenece al conjunto “hombres altos” pero otro que mide 1.69 metros ya no. Sin embargo, en el razonamiento humano no es lógico hacer dicha clasificación con tan solo 2 centímetros de diferencia. La lógica difusa considera posibles conjuntos intermedios y determinados grados de pertenencia a cada conjunto, es decir, el hombre con 1.69 metros podría tener un grado de pertenencia a cada conjunto, es decir, el hombre con 1.69 metros podría tener un grado de 0.8 de pertenencia al conjunto “hombres altos” y uno con 1.50 metros con un grado de 0.3 al mismo conjunto.

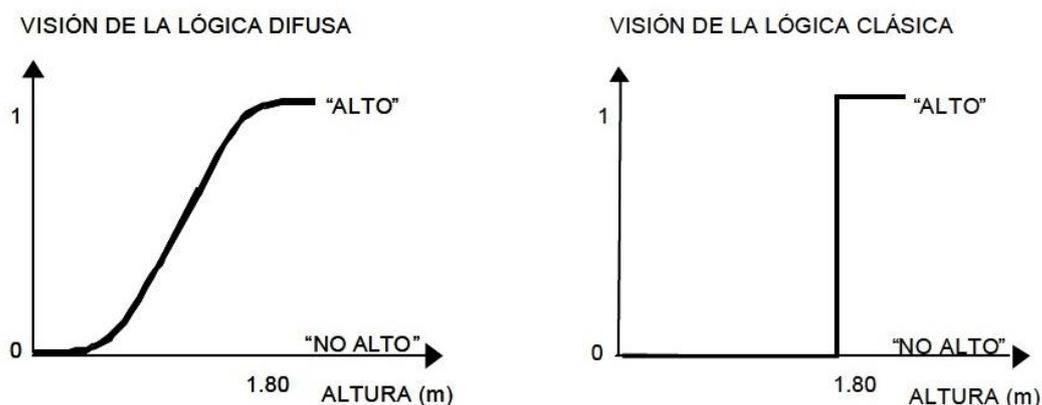


Figura 3.2. Lógica clásica y lógica difusa.

A partir de este principio, se puede decir que la lógica difusa tiene la capacidad de trabajar con sentencias en lenguaje natural (Valores lingüísticos) con cierto

grado de incertidumbre y transformarlo a un lenguaje matemático formal. Sus principales características son:

- Es flexible, se puede implementar en cualquier sistema lógico.
- Se basa en la experiencia de expertos conocedores del problema.
- Tolerante a la imprecisión, se basa en el lenguaje humano.
- Combina expresiones lingüísticas con datos numéricos.
- Puede modelar funciones no lineales, sistemas lógicos complejos.
- Conceptualmente de fácil comprensión.

### 3.1 CONCEPTOS BÁSICOS DE LÓGICA DIFUSA

#### 3.1.1 CONJUNTOS DIFUSOS

*En términos generales un conjunto difuso puede definirse como una clase en la que hay una progresión gradual desde la pertenencia al conjunto hasta la no pertenencia [4], es decir, una sentencia no es totalmente cierta o falsa sino que un conjunto difuso ofrece una transición suave entre los límites de un conjunto clásico, como a continuación se muestra.*

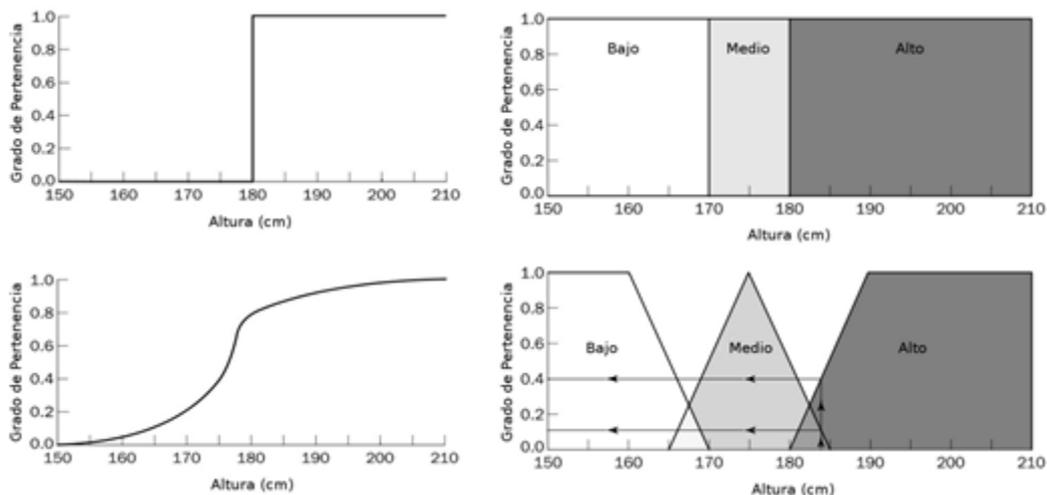


Figura 3.3. Conjunto clásico (arriba) y conjunto difuso (abajo).

Existen tres operaciones básicas sobre conjuntos clásicos (complemento, unión e intersección), estas operaciones se comportan de la misma manera en conjuntos difusos.

1. La unión es el valor de membresía más grande entre dos conjuntos. Sean A y B dos intervalos difusos, la unión entre estos dos es el conjunto difuso C:

$$A \cup B = \max[\mu_A(x), \mu_B(x)]$$

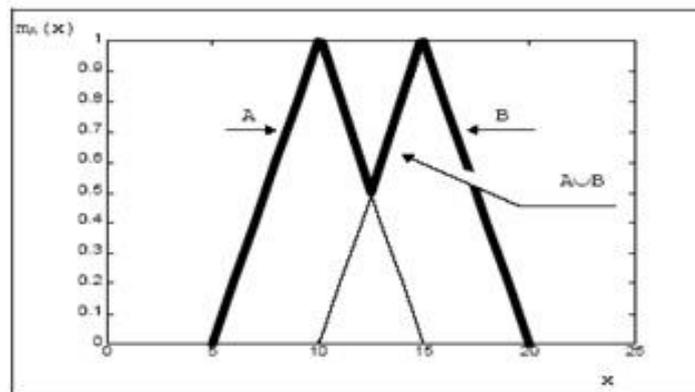


Figura 3.4. Función de membresía del conjunto C

2. El complemento de un conjunto difuso, es un opuesto a este conjunto:

$$\overline{\mu_A}(x) = 1 - \mu_A$$

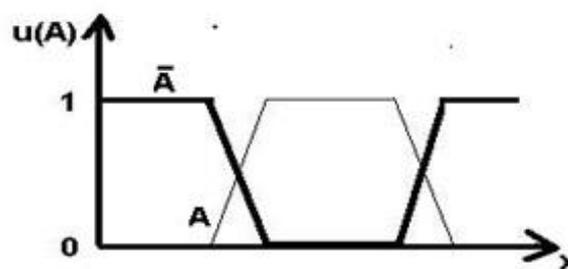


Figura 3.5. Complemento del conjunto difuso A

3. La intersección es el valor de membresía más bajo entre dos conjuntos. La intersección de los conjuntos difusos A y B es el conjunto difuso C:

$$\mu_A \cap \mu_B = \min[\mu_A, \mu_B]$$

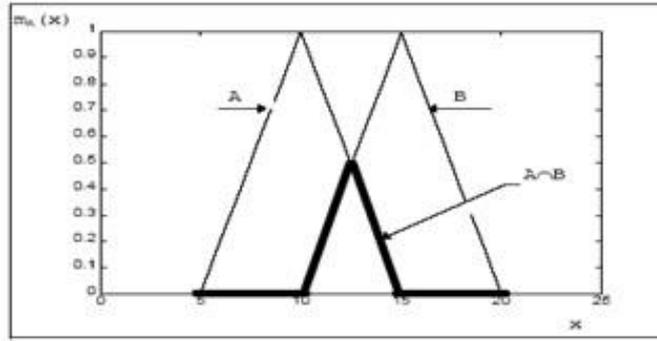


Figura 3.6. Función de membresía del conjunto C

### 3.1.2 FUNCIÓN DE MEMBRESÍA (PERTENENCIA)

La función de membresía es la función que caracteriza de manera gráfica a un conjunto difuso sobre el universo del discurso a través del grado de pertenencia de cada elemento.

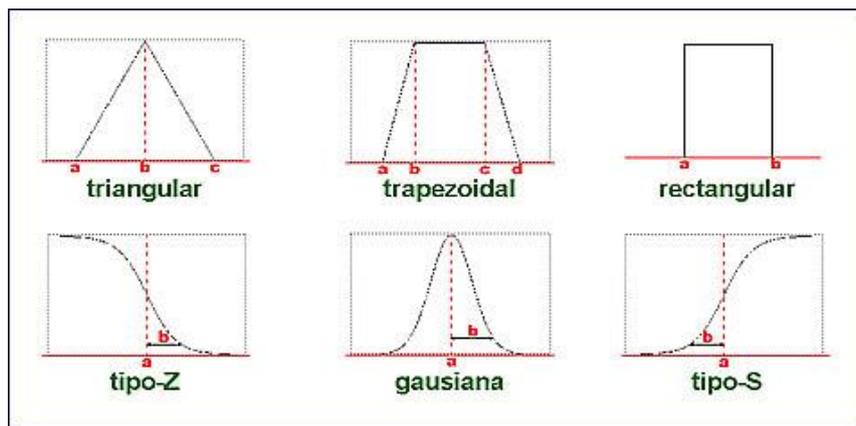


Figura 3.7. Tipos de funciones de membresía.

En la figura anterior se muestran algunos de los tipos de funciones de membresía. Las formas más utilizadas en sistemas difusos son: triangulares, trapezoidales y singleton.

Forma triangular: Esta función depende de 3 parámetros escalares (Límite inferior, límite superior y el valor modal).

$$f(x) = \begin{cases} 0 & \text{si } x \leq a \\ (x - a)/(b - a) & \text{si } x \in (a, b] \\ (c - x)/(c - b) & \text{si } x \in (b, c) \\ 1 & \text{si } x \leq c \end{cases}$$

Forma Trapezoidal: Esta función depende de cuatro parámetros escalares (límites inferior, superior y de soporte inferior y superior).

$$f(x) = \begin{cases} 0 & \text{si } (x \leq a) \text{ o } (x \geq b) \\ (x-a)/(b-a) & \text{si } x \in (a, b] \\ 1 & \text{si } x \in (b, c) \\ (d-x)/(d-c) & \text{si } x \in (b, d) \end{cases}$$

Forma singleton: tiene valor de 1 solo para un punto a y 0 para el resto.

$$f(x) = \begin{cases} 0 & x \neq a \\ 1 & x = a \end{cases}$$

### 3.2 SISTEMAS DE CONTROL DIFUSO (BORROSO)

El término control automático es usado para referirse a la interacción humano-máquina, esto para reducir riesgos de operación y/o aumentar la eficiencia del sistema. La introducción de la teoría de conjuntos al control ha permitido ampliar la complejidad de los sistemas a controlar, facilitar su interpretación al expresarse en lenguaje "natural" y modelarlo en menor tiempo. En comparación con un controlador convencional, el controlador difuso usa en el conocimiento de un experto en lugar de ecuaciones diferenciales, esta base de conocimientos permitirá al sistema la capacidad de tomar decisiones según las circunstancias (Ver Figura 3.8). Para esto se deben de tener en cuenta los siguientes puntos para la etapa de diseño de un controlador:

1. La solución debe ser la más simple.
2. Se debe conocer en su totalidad el sistema a controlar.
3. Debe garantizar la estabilidad en sistemas más simples.

Se les llama dispositivos de inferencia difusa a los sistemas que interpretan las reglas de tipo IF-THEN, también conocidos como: Controlador de lógica difusa, FLC (Fuzzy Logic Controller) o Sistemas de inferencia difusa, FIS (Fuzzy Inference System).

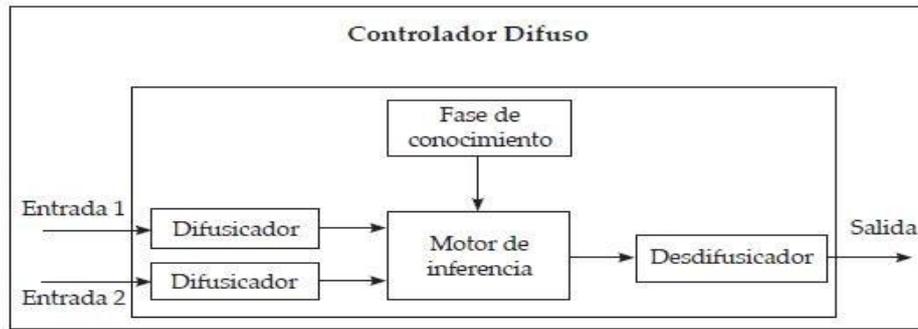


Figura 3.8. Estructura básica de un controlador difuso.

### 3.2.1 FUZIFICADOR (BORROSIFICADOR)

En esta etapa del controlador se realiza la conversión de valores reales (crisp) a valores subjetivos (difusos) mediante la asignación de un grado de pertenencia a cada conjunto difuso. Este bloque tiene como entradas valores medidos y como salida grados de pertenencia.

*En la definición de los conjuntos borrosos es muy importante el conocimiento del problema que posea el experto, aunque es importante también tener en cuenta que si el número de conjuntos borrosos definidos sobre la variable lingüística es elevado entonces obtendremos una gran resolución, pero el coste computacional también será alto. [5]*

### 3.2.2 BASE DE REGLAS

Este bloque contiene la base de conocimientos del experto que entrada se expresa en un conjunto de reglas que satisfacen los objetivos del control, tiene como entrada grados de pertenencia de cada conjunto difuso y de salida conjuntos difusos asociados a las variables de salida. Una regla de producción difusa son sentencias del tipo:

IF x es A THEN y es B.

Donde x e y son variables de entrada del sistema, por otra parte A y B son términos lingüísticos que definen a los conjuntos difusos (alto, bajo, malo, bueno, regular, etc.)

### 3.2.3 INFERENCIA

El proceso de inferencia relaciona los conjuntos difusos entrada y salida empleando la teoría de conjuntos. Se utiliza la información de la base de reglas y puede utilizar el modelo Mamdani o el TSK (Takagi, Sugeno y Kang).

### Modelo Mamdani

Es el mecanismo de inferencia más utilizado, planteado en 1975 por el profesor Ebrahim H. Mamdani. Se caracteriza porque cada consecuente de la base de reglas está formado por un conjunto difuso.

P: IF  $x_1$  is A AND  $x_2$  is B AND  $x_3$  is C THEN  $u_1$  is D,  $u_2$  is E

Si una regla contiene más de un antecedente se hace uso del operador AND (unión, T-Conorma) u OR (intersección, T-Norma), posteriormente las salidas del bloque de base de reglas se unifican para tener un único consecuente por cada variable de salida del sistema.

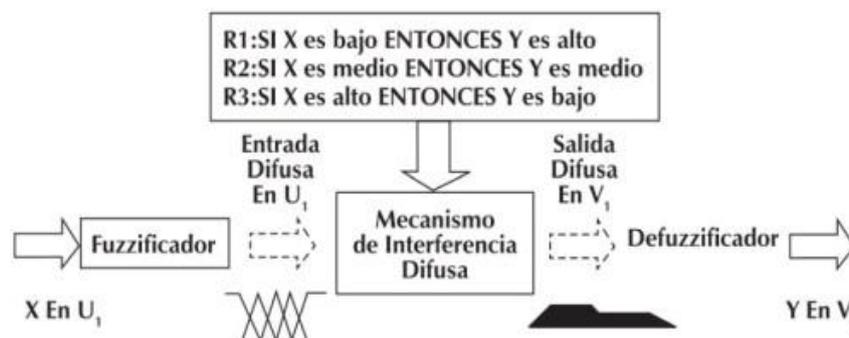


Figura 3.9. Estructura básica modelo Mamdani.

### Modelo TSK

Este método de inferencia es computacionalmente eficiente y se adapta mejor a técnicas lineales (PID) pero no representa de manera natural el conocimiento humano puesto que cada consecuente de la base de reglas contiene una ecuación respecto a las variables de entrada.

P: IF  $x$  es A AND  $y$  es B THEN  $z$  es  $f(x, y)$

#### 3.2.4 DEFUZIFICADOR (DEBORROSIFICADOR)

Esta es la última etapa del controlador, donde los datos difusos obtenidos en los bloques anteriores se convierten en valores reales (crisp) a través de métodos matemáticos como:

- Criterio de máximo (MC).
- Media de máximos (MOM, middle of maximum).
- Máximo más chico (SOM, smallest of maximum).
- Máximo más grande (LOM, largest of maximum).
- Primer Máximo (FOM, first of maximum).
- Método de Yager (YM).
- Método “Level Granding” (LGM).
- Media Difusa (FM).
- Media Difusa Ponderada (WFM).
- Centro de área (COA) o Centro de Gravedad (COG).

Este último es el método más utilizado y como su nombre lo indica calcula el punto donde una línea vertical divide el conjunto en dos áreas con masas iguales.

$$z_0 = \frac{\sum_{i=1}^N X_i \cdot \mu_x(X_i)}{\sum_{i=1}^N \mu_x(X_i)}$$

Donde N representa el número de valores difusos.

### **3.3 HERRAMIENTAS**

#### **3.3.1 MATLAB**

MATLAB o “Laboratorio de matrices” es una herramienta matemática que cuenta con su propio lenguaje de programación (lenguaje M), alguna de las funciones que ofrece son:

- Operación con matrices.
- Representación de datos.
- Creación de interfaces de usuario (GUI).
- Comunicación con otros programas y dispositivos.

- Cajas de herramientas (Toolboxes). Plataforma de simulación multidominio (Simulink).

**Fuzzy Logic Toolbox proporciona funciones, aplicaciones y un bloque Simulink® para analizar, diseñar y simular sistemas basados en lógica difusa. El producto lo guía a través de los pasos para diseñar sistemas de inferencia difusos. Se proporcionan funciones para muchos métodos comunes, incluyendo agrupamiento difuso y aprendizaje neurofuzzy adaptativo. [6]**

La aplicación para el diseño del FIS da la oportunidad de elegir el tipo de mecanismo de inferencia (Mamdani o TSK), el tipo de operador, la forma de las funciones de membresía, así mismo se puede visualizar el comportamiento de la base de reglas con entradas determinadas.

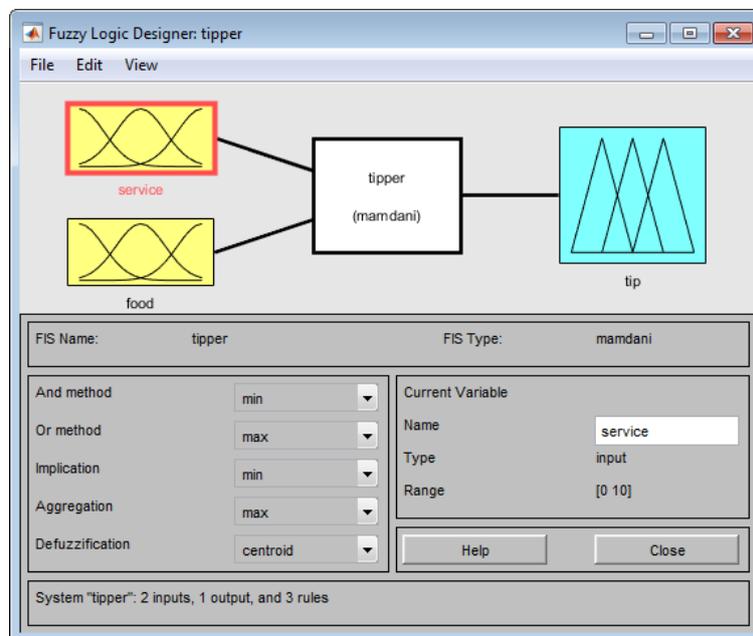


Figura 3.10. Fuzzy Logic Designer App.

La siguiente figura muestra el bloque de Simulink, con esta herramienta es posible simular el sistema de inferencia.

**Temperature Control in a Shower**  
Copyright (c) 2002-2018 The MathWorks, Inc.

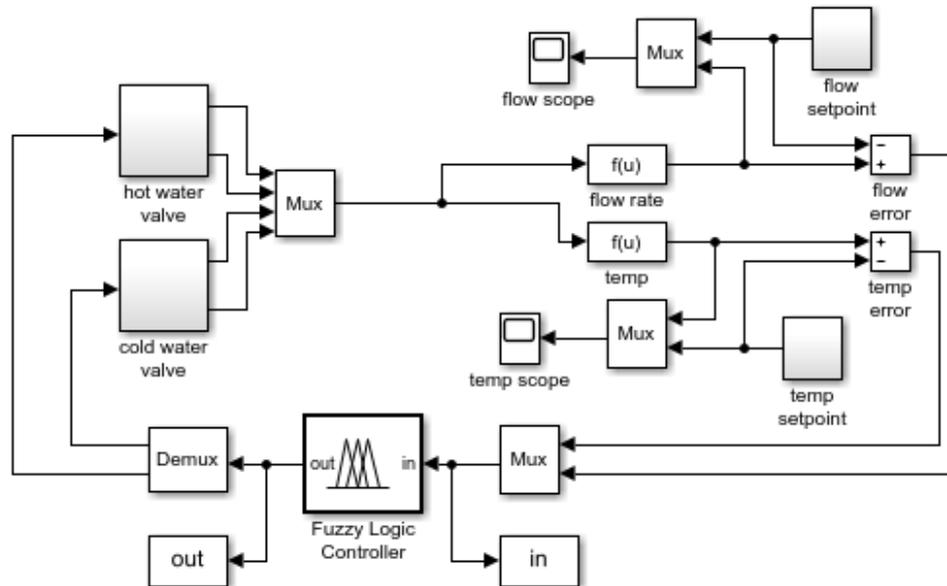


Figura 3.11. Ejemplo de FIS en Simulink.

### 3.3.2 XFUZZY

Xfuzzy es una interfaz gráfica programada en Java para desarrollar sistemas de control difuso complejos, ofreciendo la representación gráfica en 2D, la superficie de control (3D), simulación y síntesis en python, java, c, c++ y VHDL.



Figura 3.12. XFUZZY 3.0.

*Xfvhd traslada la especificación de un sistema difuso escrita en XFL3 en una descripción VHDL que puede ser sintetizada e implementada sobre un dispositivo programable o como un circuito integrado para aplicaciones específicas.[7]*

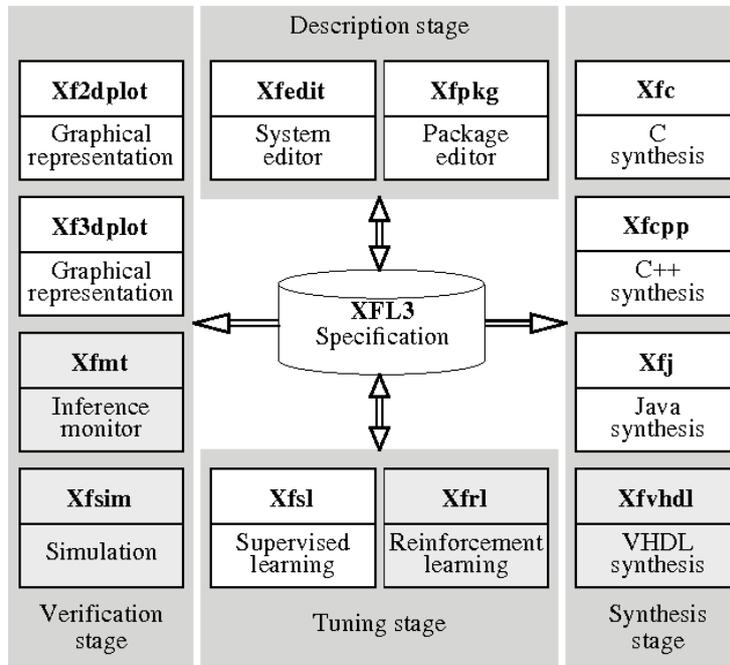


Figura 3.13. Flujo de diseño de Xfuzzy.

## CAPÍTULO 4. FPGA (FIELD- PROGRAMMABLE GATE ARRAY)

Como su nombre lo dice es un arreglo de compuertas lógicas programables que a diferencia de los procesadores operan de manera paralela, lo cual ahorra recursos de procesamiento. **Cada chip FPGA está compuesto por un número finito de recursos predefinidos con interconexiones programables para implementar un circuito digital reconfigurable y bloques de ETS para permitir que el circuito acceda al mundo exterior, [8]** (Ver Figura 4.1). Cabe destacar que las FPGAs permiten describir el comportamiento y estructura que se desea mediante un lenguaje específico (VHD, Verilog)

Las principales ventajas del uso de FPGAs sobre los procesadores son:

- Rapidez de procesamiento.
- Potencia computacional.
- generación de prototipos y verificación sin el proceso de fabricación del diseño personalizado de ASIC.
- Actualización rápida.

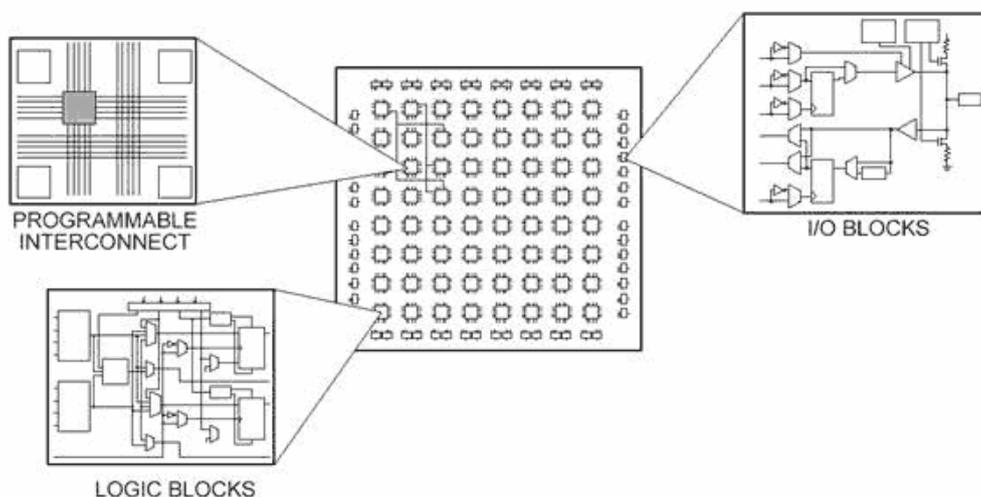


Figura 4.1. Estructura FPGA.

### 4.1 VHDL

VHDL surge entre los años 1984 y 1987 gracias a una colaboración entre el IEEE y el Departamento de Defensa de los Estados Unidos. El significado de las siglas VHDL viene de VHSIC HDL (Very High Speed Integrated Circuits Hardware Description Language), esto quiere decir que no es un lenguaje de

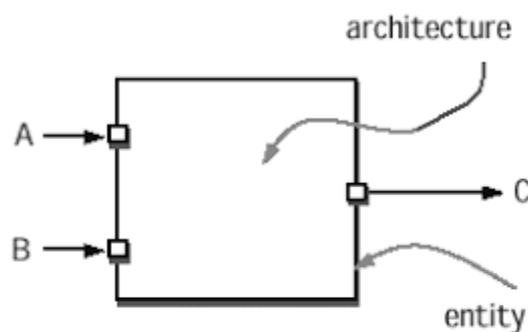
programación sino un lenguaje de diseño de hardware que permite describir la estructura y comportamiento de circuitos síncronos y asíncronos.

Algunas características importantes que se deben tomar en cuenta para implementar un sistema en este lenguaje son:

- Cada elemento tiene una interfaz única que hace posible su simulación.
- El diseño es jerárquico.
- Cualquier diseño puede ser simulado.
- Tener en cuenta el diseño de cada elemento: Combinacional o secuencial.
- Evitar uso de relojes condicionales.
- VHDL ignora saltos de línea y espacios.
- No reconoce la diferencia entre mayúsculas y minúsculas de palabras reservadas.

#### 4.1.1 ESTRUCTURA

La estructura general de un sistema diseñado en VHDL se basa en el diseño estructural utilizado en Pascal y ADA, por lo que está formado por dos partes la entidad y la arquitectura.



4.2 Estructura archivo VHDL.

En la entidad (Entity) se describe de manera externa al circuito, es decir, se declaran entradas y salidas (in, out, inout o buffer), se les asigna un nombre, se describe el tipo de dato (std\_logic, std\_logic\_vector, integer, etc).

Por otro lado la arquitectura describe el comportamiento de la entidad, en otros términos define al circuito internamente a través de constantes, funciones o procesos.

Para un correcto diseño e implementación de un sistema en VHDL existen dos etapas. La primera etapa es de desarrollo y consiste en lo siguiente:

1. Realización de un diagrama de bloques que represente al sistema.
2. Codificación del código VHDL.
3. El compilador comprueba la sintaxis para convertir el programa en un objeto.
4. Verifica la correcta funcionalidad del circuito con condiciones reales.
5. Simulación del comportamiento de cada elemento del sistema.

La segunda etapa es de realización y se define en los siguientes pasos:

1. El proceso de Síntesis modela lo descrito en el programa a través de ecuaciones booleanas, listas o mallas de compuertas lógicas.
2. El enrutamiento o mapeo se encarga de interconectar cada recurso necesario según el código VHDL.
3. Finalmente se realiza una verificación total del circuito.

## **4.2 VERILOG**

Al igual que VHDL, Verilog es un lenguaje para describir e implementar circuitos electrónicos, fue inventado en 1985 por el ingeniero Phil Moorby. Su sintaxis también es similar al lenguaje de programación C, ofrece la opción de utilizar números negativos, trabajar con diferentes bases (octal, decimal, hexadecimal, binario) y la estructura general de un archivo Verilog consta de tres secciones: la declaración de las entradas y salidas del módulo, declaración de señales, variables y constantes internas que se utilizarán y la descripción del módulo. En la siguiente figura se muestra un ejemplo en VHDL en contraste con el mismo ejemplo descrito en Verilog.

## VHDL

```
library ieee;
use ieee.std_logic_1164.all;

entity shift is
  port(C, SI : in std_logic;
        SO : out std_logic);
end shift;

architecture Behav of shift is
  signal tmp: std_logic_vector(7 downto 0);
begin
  process (C)
  begin
    if (C'event and C='1') then
      for i in 0 to 6 loop
        tmp(i+1) = tmp(i);
      end loop;
      tmp(0) = SI;
    end if;
  end process;
  SO = tmp(7);
end Behav;
```

## Verilog

```
module shift (C, SI, SO);
  input C,SI;
  output SO;
  reg [7:0] tmp;

  always @(posedge C)
  begin
    tmp = tmp << 1;
    tmp[0] = SI;
  end
  assign SO = tmp[7];
endmodule
```

Figura 4.3.Comparación VHDL vs Verilog.

Este lenguaje de diseño permite realizar la descripción del sistema a diferentes niveles de abstracción: Compuerta, RTL, Comportamiento.

- Nivel compuerta: Descripción de bajo nivel, hace uso de compuertas lógicas (and, or, not, nand, nor, xor, xnor,buf) y sus ecuaciones para crear una red lógica. Para fines de simulación se recomienda asignar un retardo por compuerta lógica.
- **Los diseños descritos a nivel RTL especifican las características de un circuito mediante operaciones y la transferencia de datos entre registros. Mediante el uso de especificaciones de tiempo las operaciones se realizan en instantes determinados. La especificación de un diseño a nivel RTL le confiere la propiedad de diseño sintetizable. [9]**
- Este último nivel es independiente a los anteriores, describe el comportamiento del sistema mediante algoritmos (sentencias, expresiones lógicas, procedimientos, etc.) en paralelo que se ejecutan de manera secuencial o estructurada.

## **CAPÍTULO 5. METODOLOGÍA**

Para el diseño adecuado del controlador difuso para el microclima dentro del invernáculo se tomaron en cuenta los datos de la sección 2.1.1, adicionalmente se establecieron los valores y variables usados dentro del sistema de control.

### **5.1 DISEÑO DEL SISTEMA DIFUSO**

#### **5.1.1 VARIABLES LINGÜÍSTICAS**

- Temperatura.
- Humedad.
- Variación de Temperatura.

#### **5.1.2 VALORES LINGÜÍSTICOS (ENTRADAS)**

##### **Temperatura**

- Muy frío (MF).
- Frío (F).
- Normal (N).
- Alta (A).
- Muy Alta (MA).

##### **Humedad**

- Muy baja (MB).
- Baja (B).
- Media (M).
- Alta (A).
- Muy Alta (MA).

#### **5.1.3 VALORES LINGÜÍSTICOS (SALIDA)**

- Descenso grande (DG).
- Descenso normal (DN).
- Descenso pequeño (DP).

- Mantener temperatura (MT).
- Ascenso pequeño (AP).
- Ascenso normal (AN).
- Ascenso grande AG).

### 5.1.4 MATRIZ ASOCIATIVA DIFUSA (FAM)

Tabla 5.1 Matriz de reglas difusas

T / HR	MB	B	N	A	MA
MF	AN	AN	AG	AG	AG
F	M	M	AP	AP	AN
N	M	M	M	M	DP
A	M	M	DP	DP	DN
MA	DP	DN	DN	DG	AG

Se eligió el modelo tipo Mamdani, la regla composicional usa el operador máx.-min y como método de defuzificación centro de área (Centroide) para aprovechar los datos generados por el sistema de control.

## 5.2 IMPLEMENTACIÓN DEL SISTEMA DIFUSO EN MATLAB

Para crear un sistema de inferencia difuso (FIS) en MATLAB se necesitan insertar y editar las variables de entrada y salida, para este caso son dos variables de entradas y una de salida.

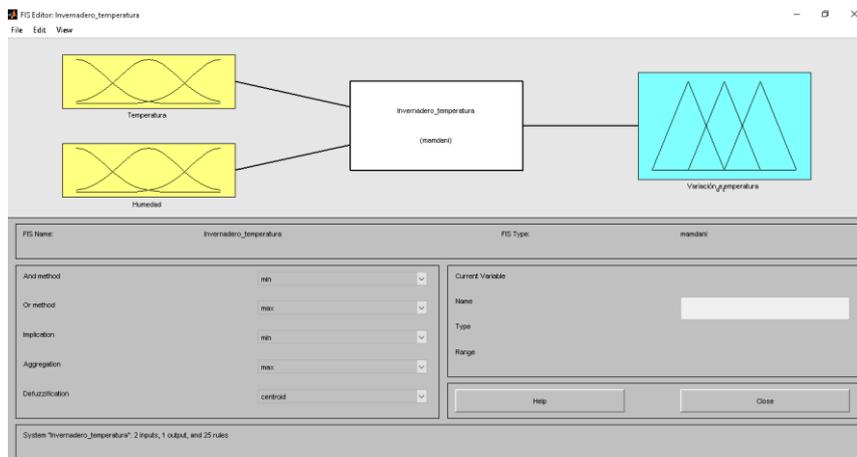


Figura 5.1. Ventana Fuzzy Logic Toolbox de MATLAB.

La primer entrada “Temperatura”, cuenta con cinco funciones de membresía tipo triangular simétricas traslapadas que describen los 5 estados posibles del sistema (MF, F, N, A, MA) como se muestra a continuación.

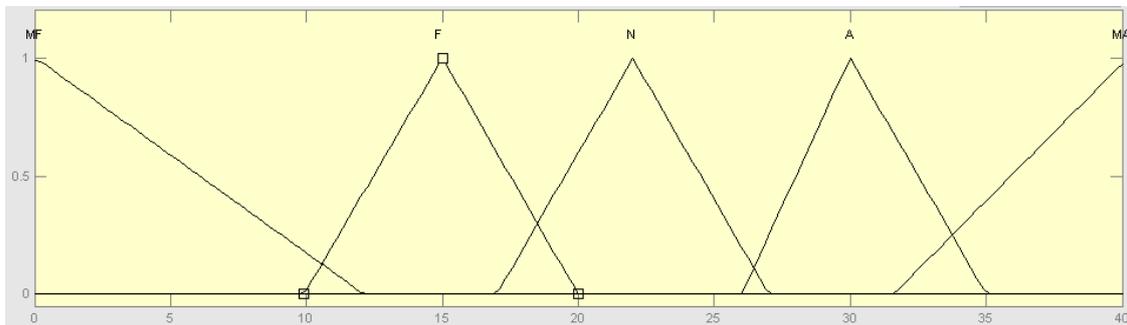


Figura 5.2. Función de membresía de la variable Temperatura.

La entrada Humedad igualmente se modela con cinco funciones de membresía (MB, B, N, A, MA) triangulares simétricas como se indica en la Figura 5.3.

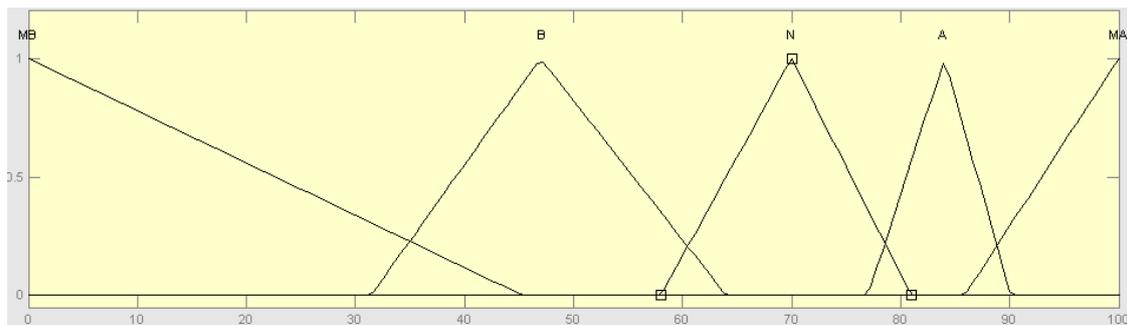


Figura 5.3. Función de membresía de la variable Humedad.

Por otra parte, la salida del sistema se conforma de siete funciones de membresía (DG, DN, DP, M, AP, AN, AG) que describen la variación de temperatura que requiere el sistema, como se ilustra en la Figura 5.4.

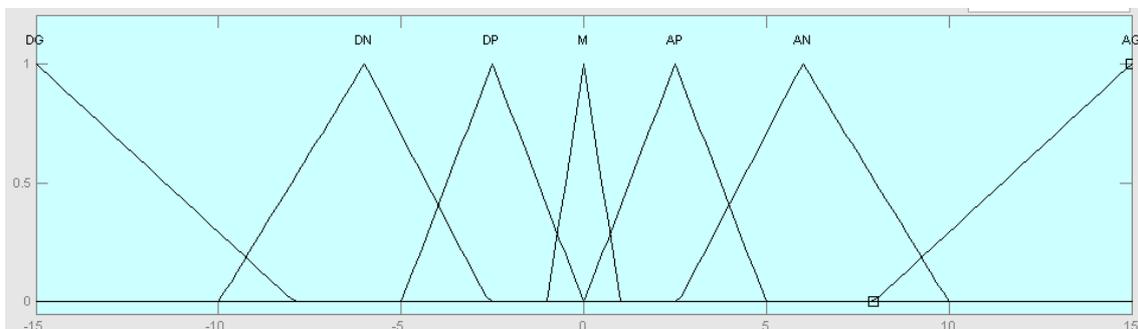


Figura 5.4. Función de membresía de la salida.

Finalmente se definen las veinticinco reglas que conforma la base de conocimientos seleccionando el conector adecuado, en este caso se utilizó AND (Ver Figura 5.5).

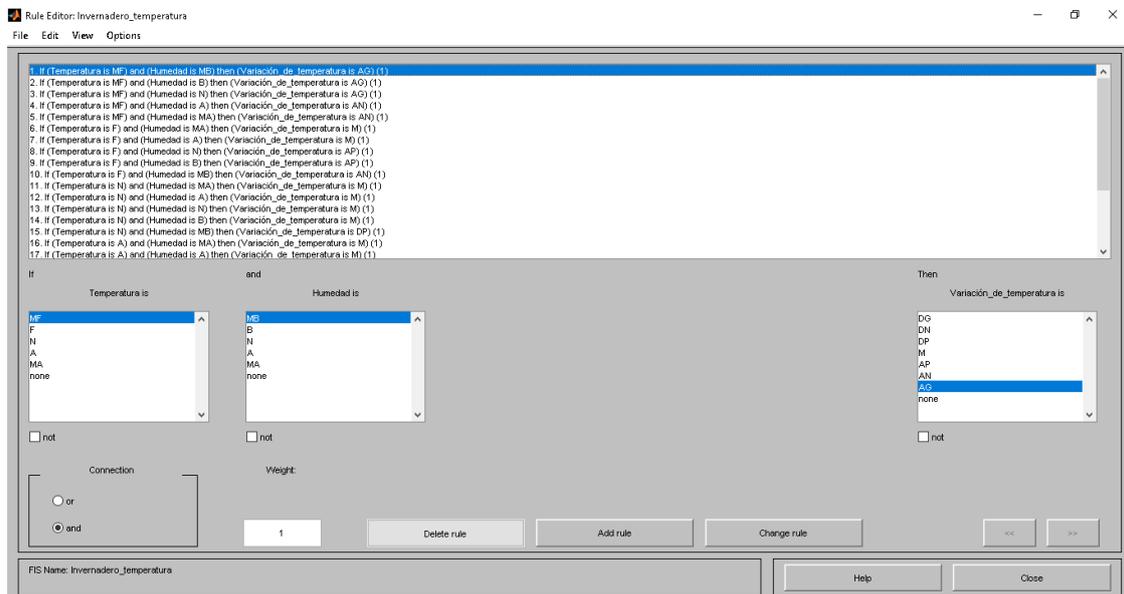


Figura 5.5. Editor de reglas del sistema en MATLAB.

### 5.2.1 SIMULINK

El bloque de Simulink se utilizó como apoyo para validar el comportamiento del controlador, es decir conocer la respuesta del sistema a un par de entradas tipo rampa. Para esto es necesario definir las entradas/salidas y agregar desde la ventana de trabajo el FIS creado por la Fuzzy Toolbox como se menciona anteriormente.

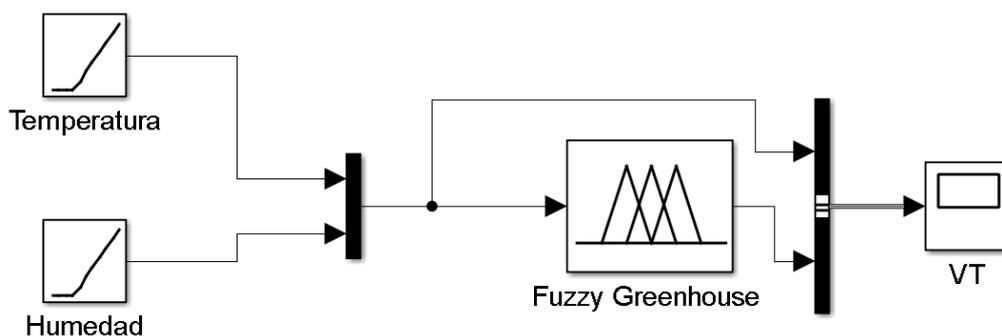


Figura 5.6. Modelo en Simulink del sistema difuso.

La herramienta Simulink es capaz de generar un diagrama que explica el funcionamiento de cada bloque presente dentro del modelo. En la Figura 5.6 se muestra el diagrama esquemático generado por Simulink del bloque que contiene el sistema de inferencia difuso.

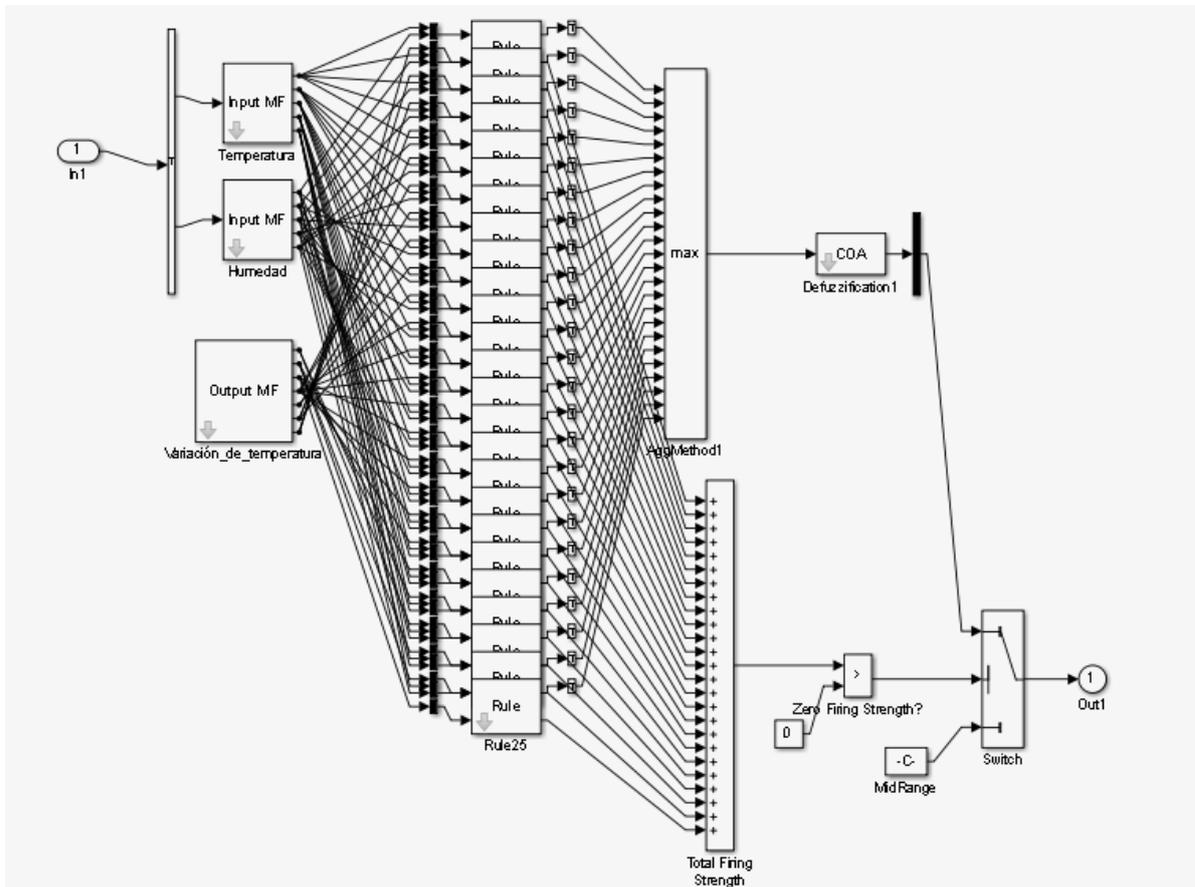


Figura 5.7. Diagrama esquemático del modelo de Simulink.

En el diagrama se puede notar los bloques que representan las 2 entradas, la salida, los 25 bloques que conforman la base de conocimientos (reglas), así como el bloque encargado de la defuzificación y los operadores del sistema.

### 5.3 IMPLEMENTACIÓN DEL SISTEMA DIFUSO EN XFUZZY

En la ventana de Xfuzzy se creó un nuevo sistema. Dentro de la ventana emergente, en el apartado “Tipos” se agregaron las tres variables del sistema para después definir las como entrada o salida según sea el caso. La interfaz de Xfuzzy también permite la definición de la base del conocimiento (Base de reglas).

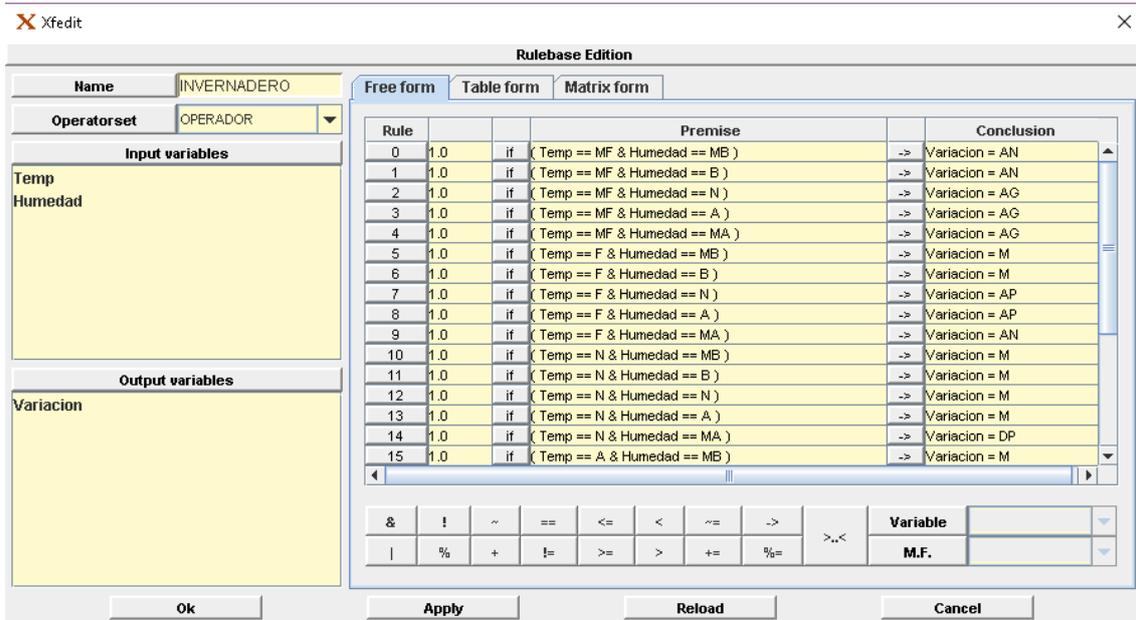


Figura 5.8. Base de reglas en Xfuzzy.

En comparación de la Toolbox de Matlab, Xfuzzy ofrece más opciones en la sección de los operadores a utilizar, así como el método de defuzificación que al usuario le convenga. En este caso, al igual que en Matlab se eligió como operador máx.-min para la regla composicional y defuzificación por centroide.

Finalmente se conectó las entradas y la salida a través del módulo que representa el sistema difuso (Ver Fig. 5.9). Al guardar el sistema se puede acceder a algunas herramientas de simulación, verificación, graficación y síntesis.

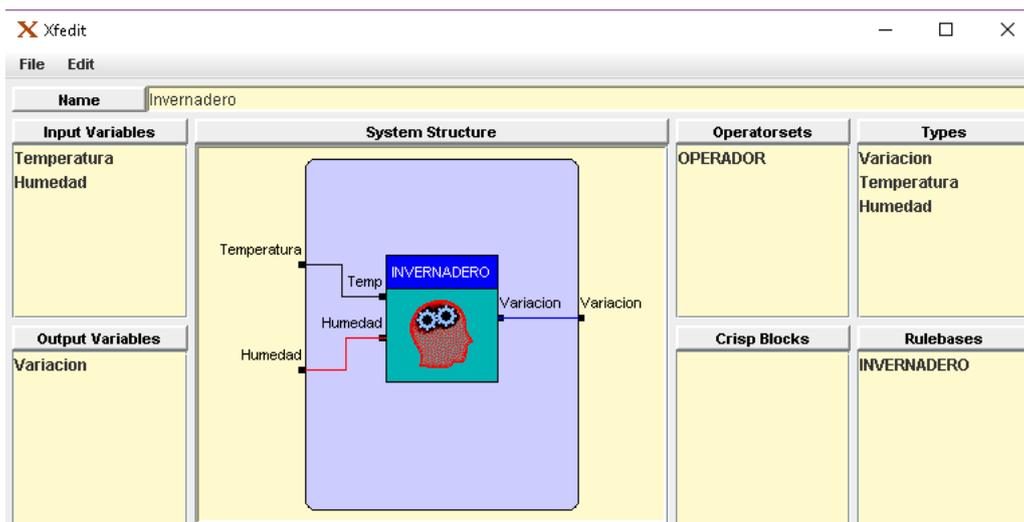


Figura 5.9. Ventana Xfedit del sistema de control Invernadero.

## 5.4 IMPLEMENTACIÓN DEL SISTEMA DIFUSO EN FPGA (VHDL)

La implementación en el entorno de XFuzzy permite el uso de una librería desarrollada por el Instituto de Microelectrónica de Sevilla (IMSE-CNM) que es útil para la codificación del sistema difuso en VHDL. El sistema difuso contiene cuatro entradas: la temperatura (in1), la humedad (in2), el reloj del sistema (CLK), el restablecimiento del controlador (RST) y una bandera para el funcionamiento adecuado de los diferentes subsistemas. Cada una de las entradas del FIS (in1/in2) se encuentran en un rango de 0 a 255 para un controlador de 8 bits.

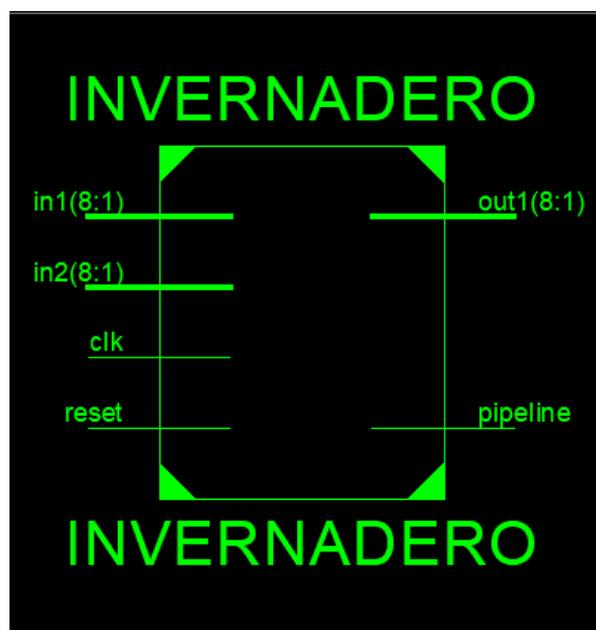


Figura 5.10. Esquemático RTL general del controlador difuso.

El diseño se trabajó con bloques independientes para cada etapa del controlador y posteriormente fueron integrados, es decir un diseño estructural. El esquemático general del controlador se presenta en la Figura 5.11. Los dos bloques denominados Arithmetic se usan para el proceso de fuzificación de las entradas Temperatura y Humedad. La inferencia, responsable de relacionar los valores de entrada con los de salida a través de la base de reglas es realizado por los bloques Rules y Minimum. Finalmente la defuzificación que acondiciona los valores difusos en valores reales es implementada por los bloques Weighted Fuzzy Mean y Division.

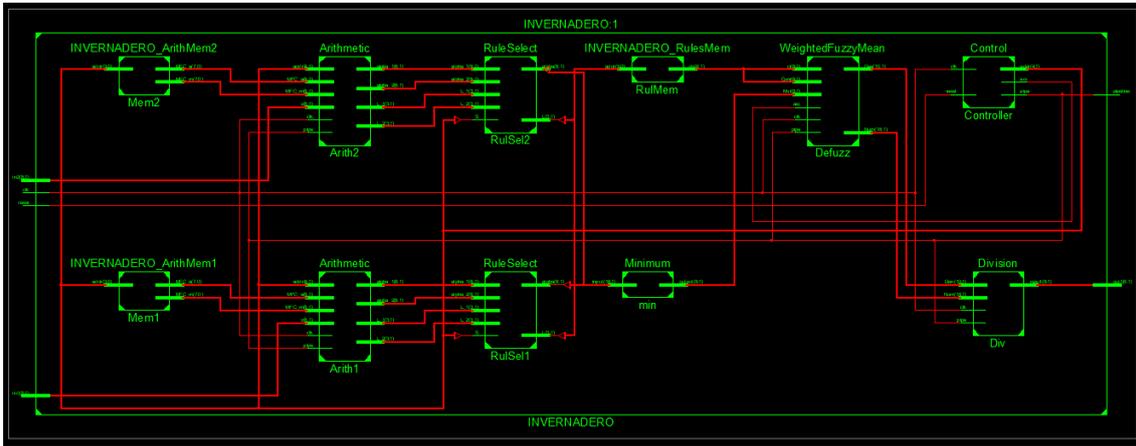


Figura 5.11. Esquemático RTL del controlador difuso.

### 5.4.1 ETAPA DE FUZIFICACIÓN

Se diseñaron dos bloques en VHDL para la etapa de fuzificación de cada entrada, el primer bloque se encarga de definir cuáles funciones de membresía son activadas por la entrada (valor lingüístico), y así enviar a la salida el valor de quiebre y la pendiente de dicha función. Este proceso se realiza usando la estructura de control switch-case. De este modo en el siguiente bloque se calcula el grado de pertenencia de la entrada correspondiente a través de la ecuación de la recta despejada (Ver Fig. 5.12 y 5.13).

```

Read_mem : process(s_addr)
begin
  case s_addr is
    when "000" => s_data <= "0000000011111111";
    when "001" => s_data <= "0011111111111111";
    when "010" => s_data <= "0111111111111111";
    when "011" => s_data <= "1011111111111111";
    when "100" => s_data <= "1111111110000000";
    when others => s_data <= "-----";
  end case;
end process Read_mem;

```

Figura 5.12. Selección de las funciones de membresía afectadas por el valor de entrada.

$$s\_mult \leq ((x - MFC\_a) * MFC\_m);$$

Figura 5.13. Cálculo del grado de membresía.

## 5.4.2 ETAPA DE INFERENCIA

La etapa de inferencia está conformada por: un bloque que contiene las 25 reglas de la base de conocimiento, un bloque de cada entrada para seleccionar las reglas activadas por los valores lingüísticos de la etapa anterior y otro que realiza la operación MIN.

En la figura 5.14 se muestra parte del código del bloque de la base de reglas del sistema difuso que tiene como salida una etiqueta que representa al consecuente de cada regla.

```
case addr is
  when "000000" => s_data <= "11010100";
  when "000001" => s_data <= "11010100";
  when "000010" => s_data <= "11111111";
  when "000011" => s_data <= "11111111";
  when "000100" => s_data <= "11111111";
  when "001000" => s_data <= "01111111";
  when "001001" => s_data <= "01111111";
  when "001010" => s_data <= "10101010";
  when "001011" => s_data <= "10101010";
  when "001100" => s_data <= "11010100";
  when "010000" => s_data <= "01111111";
  when "010001" => s_data <= "01111111";
  when "010010" => s_data <= "01111111";
  when "010011" => s_data <= "01111111";
  when "010100" => s_data <= "01010101";
  when "011000" => s_data <= "01111111";
  when "011001" => s_data <= "01111111";
  when "011010" => s_data <= "01010101";
  when "011011" => s_data <= "01010101";
  when "011100" => s_data <= "00101010";
  when "100000" => s_data <= "01010101";
  when "100001" => s_data <= "00101010";
  when "100010" => s_data <= "00101010";
  when "100011" => s_data <= "00000000";
  when "100100" => s_data <= "11111111";
  when others => s_data <= "-----";
end case;
```

Figura 5.14. Base de reglas.

El código de la operación mínimo (min) se realiza mediante una rutina de comparación, tal como se muestra en la Figura 5.15.

```

function mmin(A, B : std_logic_vector(grad downto 1))
  return std_logic_vector is

  variable C : std_logic_vector(grad downto 1);

begin

  if (A > B) then
    C := B;
  else
    C := A;
  end if;
  return C;

end mmin;

```

Figura 5.15. Rutina de comparación para realizar la operación mínimo.

### 5.4.3 ETAPA DE DEFUZIFIZACIÓN

Finalmente el diseño del defuzificador por método del centro de gravedad se basa en dos bloques. El primer bloque realiza la sumatoria del producto de cada consecuente entregado por la etapa de inferencia y su respectivo peso.

En la figura 5.16 los términos s\_mult 1 y 2 es el producto de los valores arrojados por las dos etapas anteriores del controlador.

```

s_Num <= ext( s_mult2, bitsn) + ext(s_Num, bitsn);
s_Den <= ext(s_mult1, bitsn-No) + ext(s_Den, bitsn-No);

```

Figura 5.16. Defuzificación por centroide.

Todos los bloques del controlador anteriormente descritos operan a través de una señal bandera generada por la unidad de control que se encarga del inicio y paro de cada etapa del FIS.

```

if (s_cnt = ncp - 1) then
  s_cnt <= 0;
  pipe <= '1';
else
  s_cnt <= s_cnt + 1;
  pipe <= '0';
end if;

```

Figura 5.17. Control de inicio y paro de cada bloque del sistema.

Cabe mencionar que el controlador contiene un archivo donde se definen todas las variables que describen al sistema difuso, como son: bits y número de

funciones de membresía, pendiente y punto de quiebre de las mismas, bits de las entradas y salida, etc. lo cual permite rediseñar de manera fácil las características del sistema si así se requiere.

## 5.5 IMPLEMENTACIÓN EN HARDWARE

Finalmente para la implementación física se eligió el sensor de temperatura y humedad relativa DHT11 (Ver figura 5.18) que cuenta con su propio protocolo de comunicación de un hilo, teniendo como salida 5 bytes, de los cuales el primer byte es la parte entera de la humedad relativa y el segundo la parte decimal. Así mismo ocurre con el tercer y cuarto byte, la parte entera de la temperatura y la parte decimal. Por último el byte de paridad para confirmar que no hay datos erróneos (Ver figura 5.19).

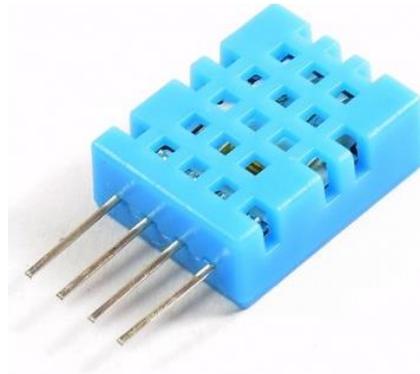


Figura 5.18. Sensor de temperatura y humedad relativa DHT11.

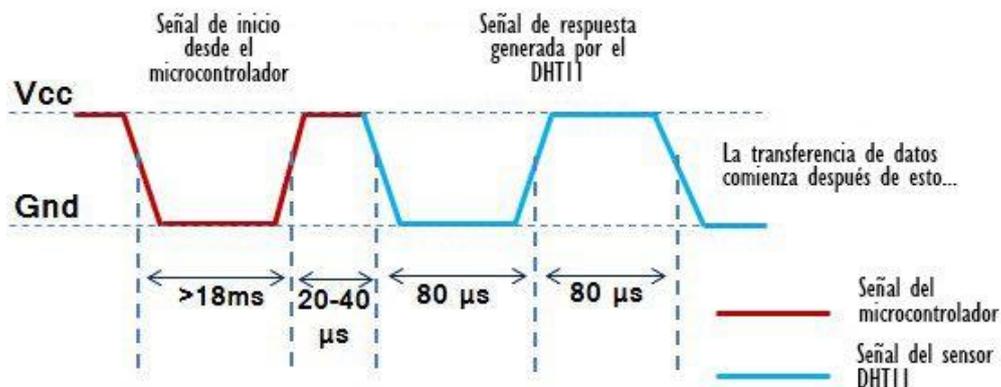


Figura 5.19. Inicialización sensor DHT11.

Al tener una comunicación de un solo hilo la lectura de datos se determina por la duración de cada pulso como se ve en la figura 5.20.

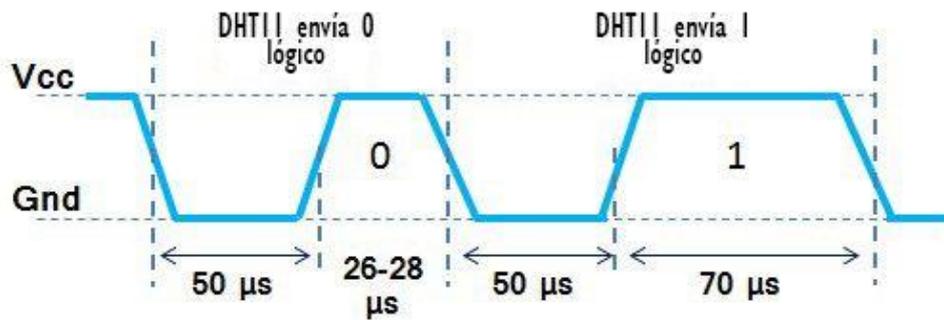


Figura 5.20. Lectura de datos sensor DHT11.

La lectura correcta del sensor se realizó por medio de una máquina de estados tipo Moore, cada estado describía el estado lógico del pin de comunicación sensor-FPGA (Ver Figura 5.21).

```

--registro (Maquina de estados)
regis_state:process (clk) begin
  if rising_edge(clk) then
    case(state) is
      when reset => -- edo inicial (inicializa las variables y establece el contador a 18ms)
        if delay_counter = 0 then
          num_bit <= data_width;
          aux <= '1';
          delay_counter <= delay_18_ms;
          state <= start_m;
        else
          delay_counter <= delay_counter - 1;
        end if;
      when start_m => -- fpga manda '1' por 18ms y pone el contador a 40us
        if delay_counter = 0 then
          aux <= '0';
          delay_counter <= delay_40_us;
          state <= wait_res_s1;
        else
          delay_counter <= delay_counter -1;
        end if ;
      when wait_res_s1 => -- espera respuesta del sensor x 40us --
        if bus_falling_edge then --
          state <= response_s1;
        end if;
    end case;
  end if;
end process;

```

Figura 5.21. Máquina de estados sensor DHT11.

En seguida se muestra el esquemático general formado por el controlador difuso, el bloque para la lectura del sensor DHT11 y un par de bloques utilizados para la visualización de los datos en una LCD.

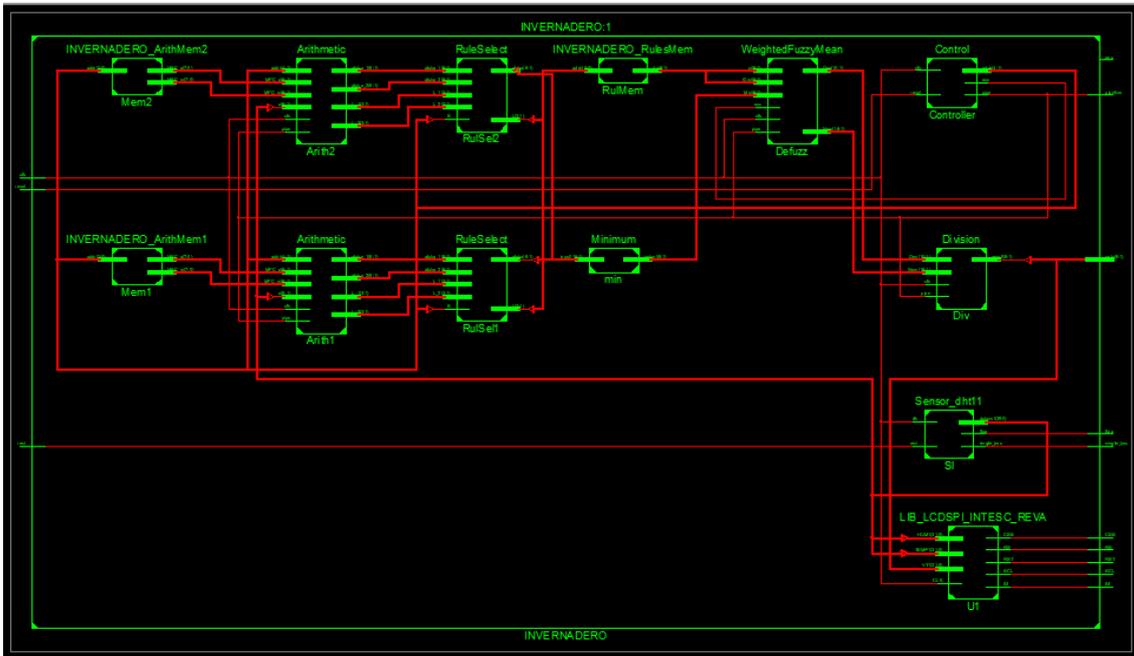


Figura 5.22. Esquemático RTL general.

## 5.6 RESULTADOS

Para validar el desempeño de nuestro sistema la implementación del controlador difuso se realizó en una FPGA Spartan6 XC6SLX16 de Xilinx, sintetizado en el software de Xilinx ISE DESIGN SUITE en su versión 14.7. A continuación se presentan los resultados obtenidos de las mediciones obtenidas del sensor en comparación con los resultados en MATLAB, cabe mencionar que el LED en la parte inferior izquierda de la pantalla en la tarjeta de desarrollo representa qué actuador se activaría (actuador de aire caliente = 0, actuador de aire frío = 1).

- Prueba 1: T= 22°C, H= 70%.

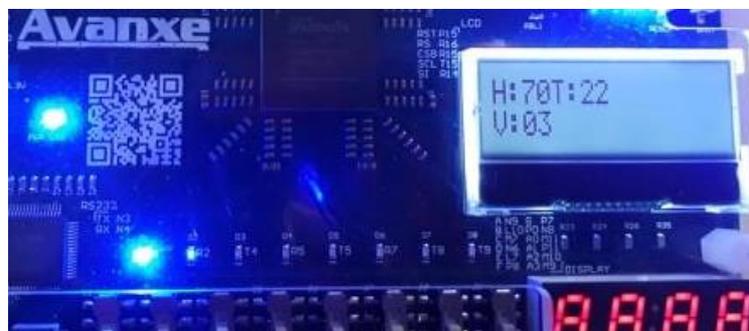
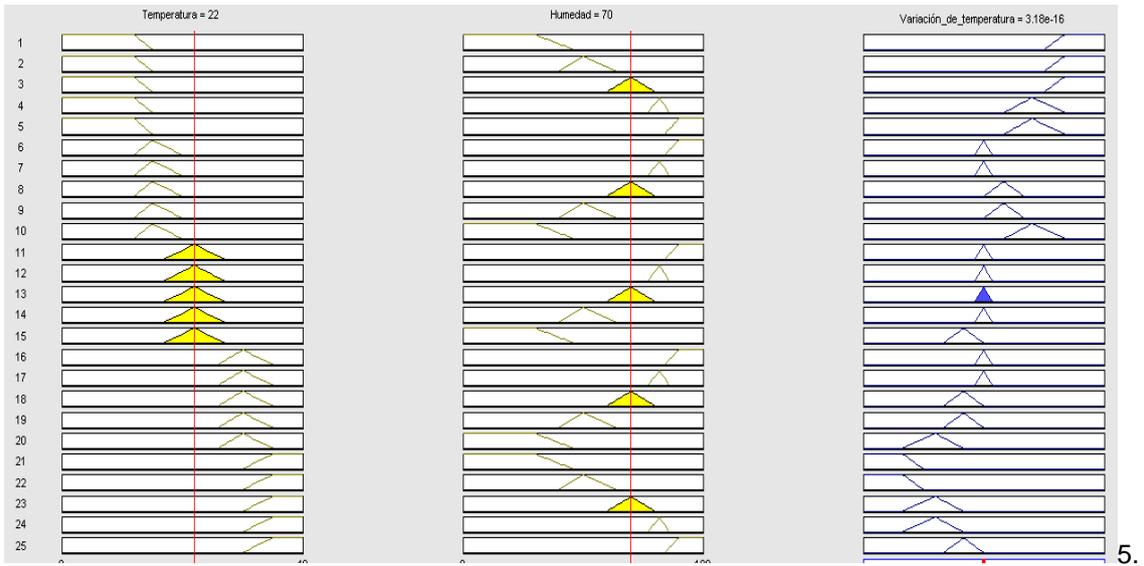


Figura 5.23. Prueba 1 en la FPGA.



24. Resultado en la Toolbox de MATLAB.

- Prueba 2: T= 25°C, H= 62%.

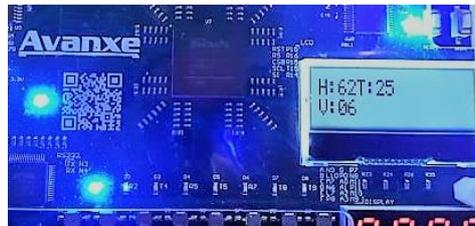


Figura 5.25. Prueba 2 en la FPGA.

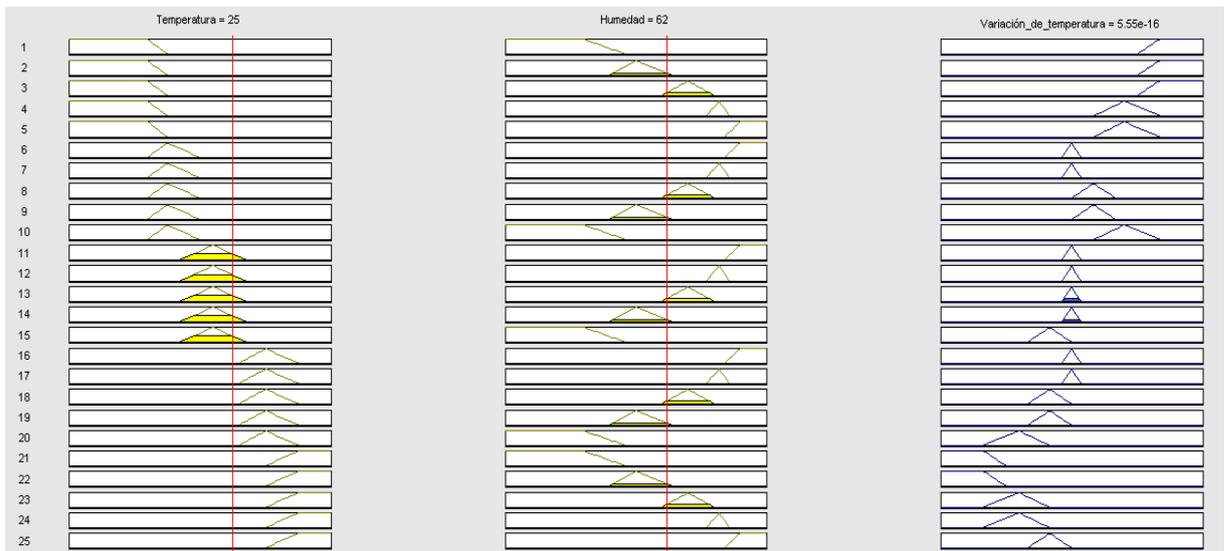


Figura 5.26. Resultado en la Toolbox de MATLAB.

- Prueba 3: T=1 9°C, H= 47%.

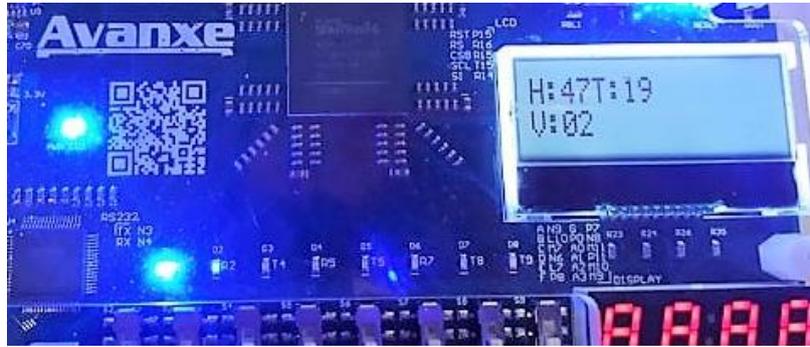


Figura 5.27. Prueba 3 en la FPGA.

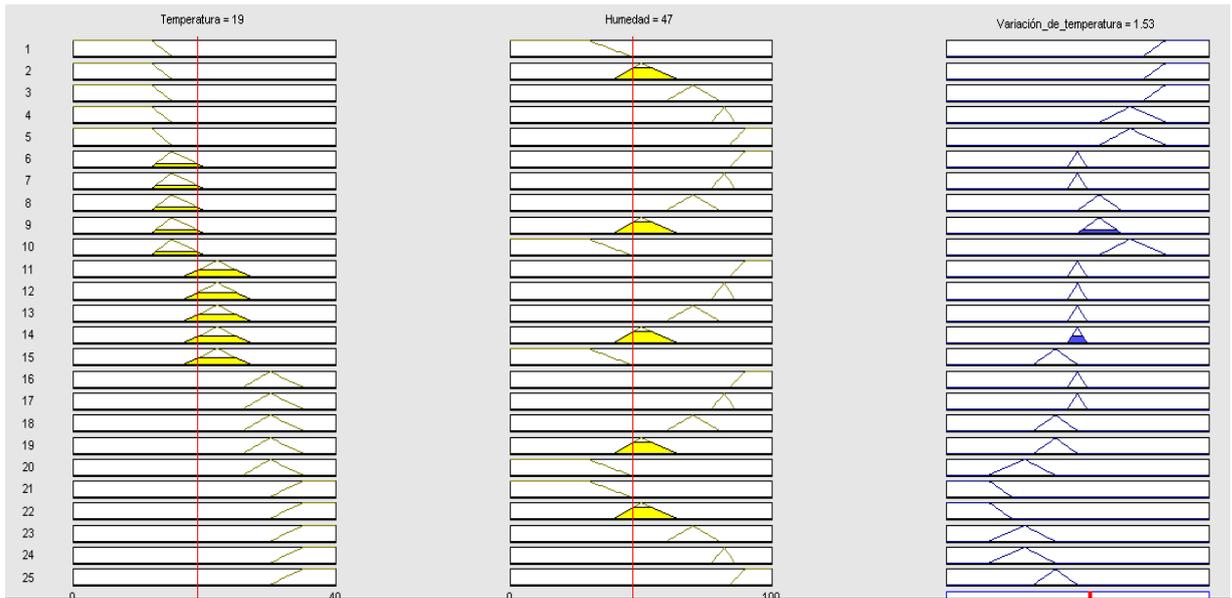


Figura 5.28. Resultado en la Toolbox de MATLAB.

- Prueba 4: T= 21°C, H= 36%.

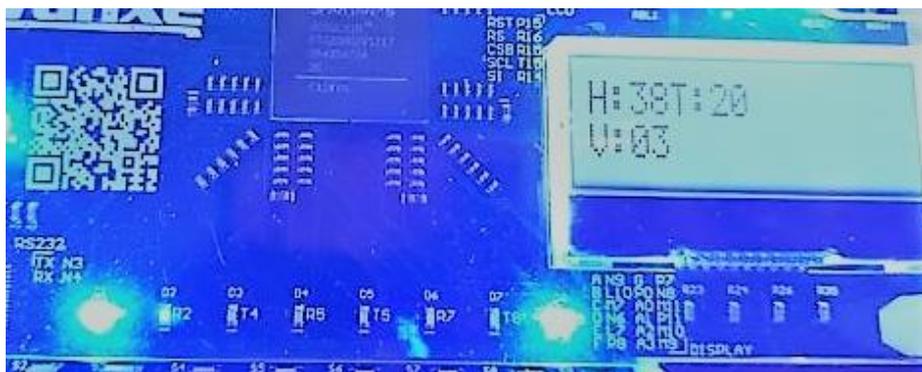


Figura 5.29. Prueba 5 en la FPGA.

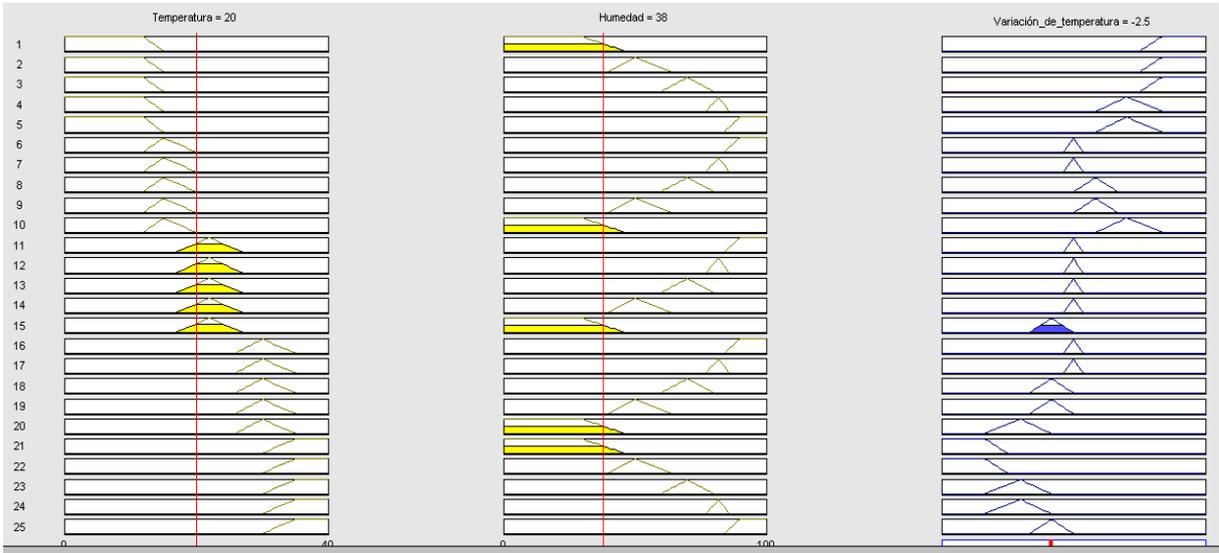


Figura 5.30. Resultado en la Toolbox de MATLAB.

- Prueba 5: T= 19°C, H= 21%.

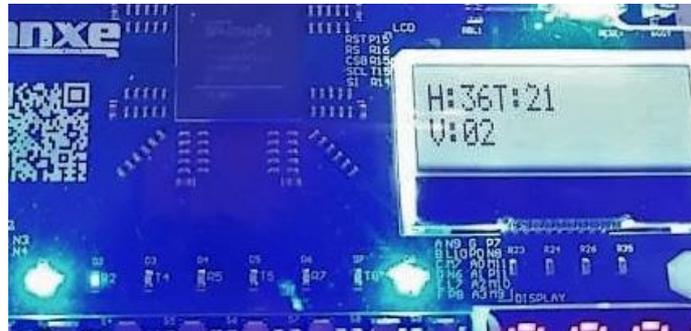


Figura 5.31. Prueba 5 en la FPGA.

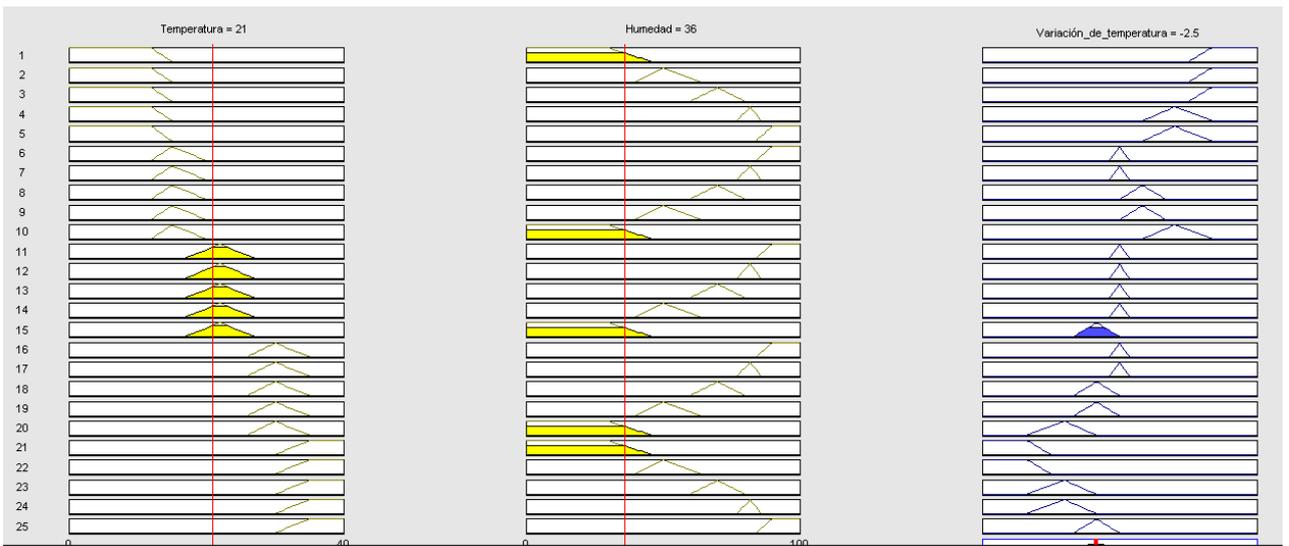


Figura 5.32. Resultado en la Toolbox de MTLAB.

Para lograr corroborar con distintas entradas sin un clima controlado se elaboró un Testbench donde se estableció un contador ascendente para cada entrada.

La salida final del sistema de control es un vector de 8 bits, donde: los 4 bits menos significativos (LSB) es la variación calculada por el controlador y el bit más significativo (MSB) nos indica si se trata de un entero positivo o negativo, es decir, si se necesita activar el actuador de aire caliente o frío. A continuación, se presentan las pruebas realizadas y la comparación de los resultados con los valores arrojados por MATLAB, uno de los software de simulación (Ver Tabla 5.2).

Tabla 5.2. Comparación datos.

FPGA	MATLAB	Diferencia
11	11.9	0.9
-9	-9.19	0.19
-7	-7.8	0.8
11	11.9	0.9
-8	7.73	0.22

1. Para la prueba 1, las entradas fueron 8°C y 16%, respectivamente, teniendo a la salida 00101011, es decir: una variación de 11, se requiere un ascenso de temperatura y para ello se activaría el actuador de aire caliente. (Ver Apéndice B)
2. Para la 2 prueba, las entradas fueron 32°C y 40%, respectivamente, teniendo a la salida 11111001, es decir: una variación de 9, se requiere un descenso de temperatura y se activaría el actuador de aire frío.(Ver Apéndice B)
3. Para la 3 prueba, las entradas fueron 32°C y 56%, respectivamente, teniendo a la salida 11110111, es decir: una variación de 7, se requiere un descenso de temperatura y para ello se activaría el actuador de aire frío. (Ver Apéndice B)
4. Para la 4 prueba, las entradas fueron 8°C y 0%, respectivamente, teniendo a la salida 00011011, es decir: una variación de 11, se requiere un ascenso de temperatura y se activaría el actuador de aire caliente.(Ver Apéndice B)

- Para la 5 prueba, las entradas fueron 32°C y 48%, respectivamente, teniendo a la salida 11111000, es decir: una variación de 8, se requiere un descenso de temperatura y para ello se activaría el actuador de aire frío. (Ver Apéndice B)

Con el fin de observar de manera más gráfica el comportamiento del sistema, se utilizó el modelo en Simulink, la respuesta a la rampa (0-40 y 0-100 respectivamente) se puede observar a continuación.

En la figura 5.33 se puede apreciar que en el rango de valores ideales para el cultivo (sección 2.1.1), el sistema tiene como salida una variación de 0, es decir mantiene las condiciones climáticas actuales.

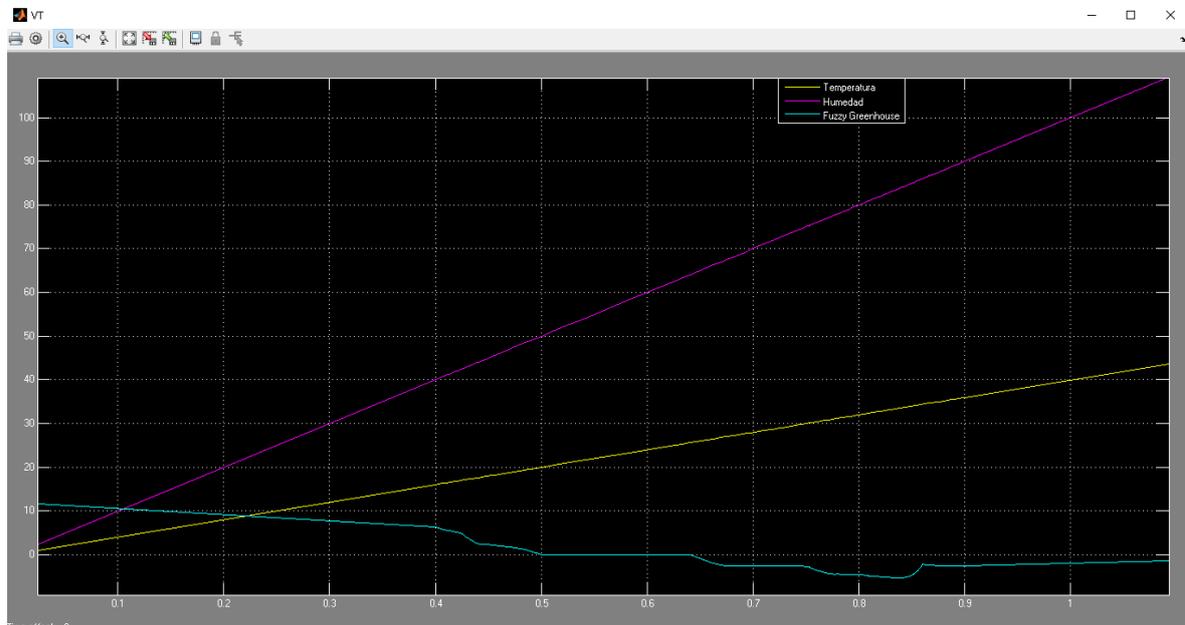


Figura 5.33. Comportamiento del sistema.

La superficie de control es la salida graficada en función de las entradas del FIS, ésta muestra el rango de posibles valores defuzificados para todas las entradas posibles. Las superficies de control obtenidas por Xfuzzy y MATLAB se presentan a continuación.

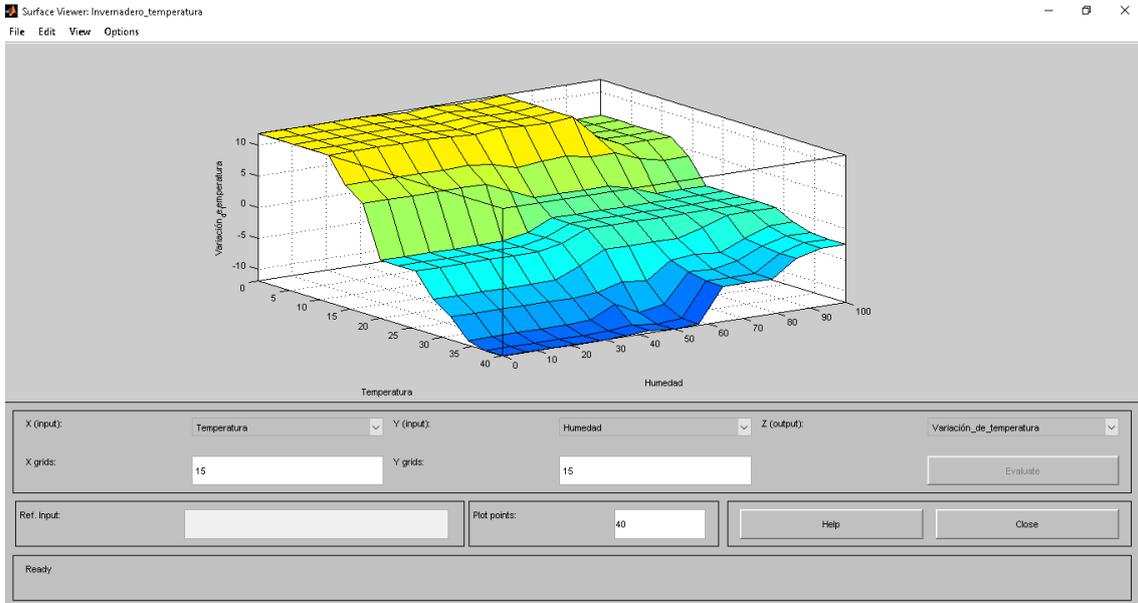


Figura 5.34. Superficie de control MATLAB.

El sistema de control obtenido muestra resultados favorables en cada software de simulación (MATLAB, Simulink y XFuzzy) que se asemejan a los datos obtenidos en la implementación física, teniendo un error promedio de 0.602.

## CONCLUSIONES

La aportación principal de este trabajo consiste en el diseño e implementación en VHDL de un sistema de inferencia difuso aplicándolo al sector agrícola, es decir, controlando la temperatura de un cultivo de jitomate bajo cubierta haciendo uso de TIC's.

Lo expuesto anteriormente permite concluir que la lógica difusa, es una técnica matemática capaz de cuantificar la vaguedad del pensamiento humano que incorporada a sistemas de control facilita el avance de la inteligencia artificial en diferentes sectores económicos del país. Ello permite que la implementación de un controlador difuso en el sector agrícola sea factible para un control más adecuado de una de las variables físicas más influyentes en el desarrollo del cultivo de jitomate. En consecuencia se mejorará la calidad del producto y aumenta el volumen de producción y beneficio económico para el usuario.

El sistema difuso presentado en el capítulo tres realiza un correcto control y monitoreo de la temperatura, a través de dos posibles actuadores (ventilación/calefacción) hace que el sistema sea eficaz y fiable para la implementación en un invernadero real.

Cabe recalcar que quedan muchas líneas de investigación por delante en la aplicación de un sistema difuso implementado en una FPGA como el que se muestra en este trabajo. Por ejemplo, en el control de la temperatura, en el campo de la domótica, en la industria con máquinas o motores dentro de líneas de producción. Así como en sistemas de riego, reconocimiento de patrones (imágenes-caracteres), etc.

## APÉNDICES

### APÉNDICE A. PROTOTIPO INVERNADERO AUTOMÁTICO



Figura A.1. Sección de riego.



Figura A.2. Vista general.

## APÉNDICE B. PRUEBAS VHDL-MATLAB

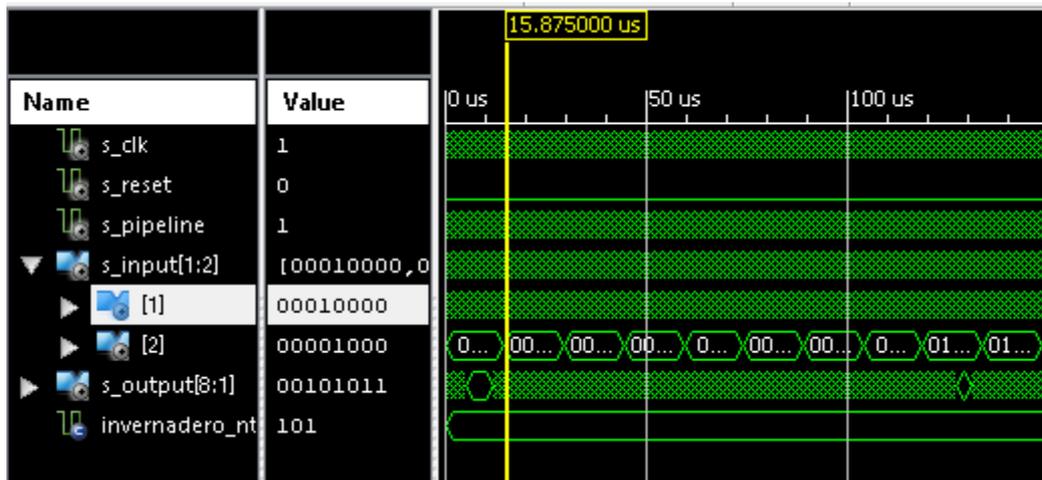


Figura B.1. Prueba 1 en el simulador de Xilinx.

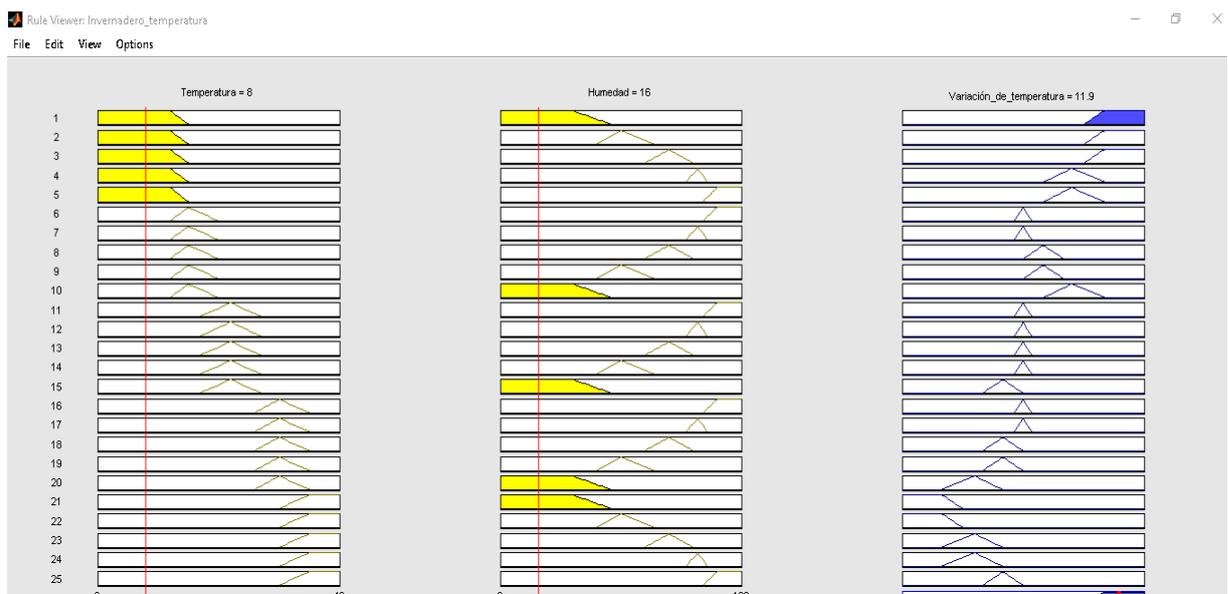


Figura B.2. Resultado 1 en la Toolbox de MATLAB.

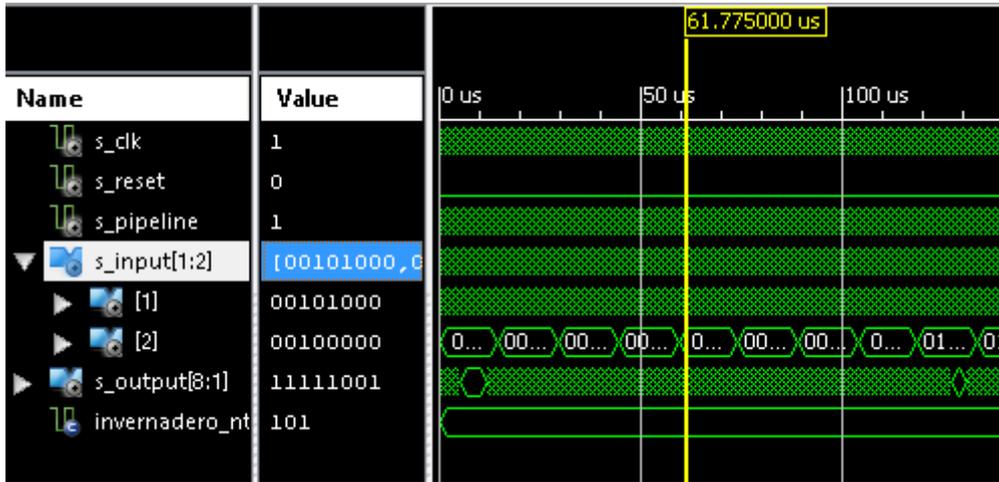


Figura B.3. Prueba 2 en el simulador de Xilinx.



Figura B.4. Resultado 2 en la Toolbox de MATLAB.

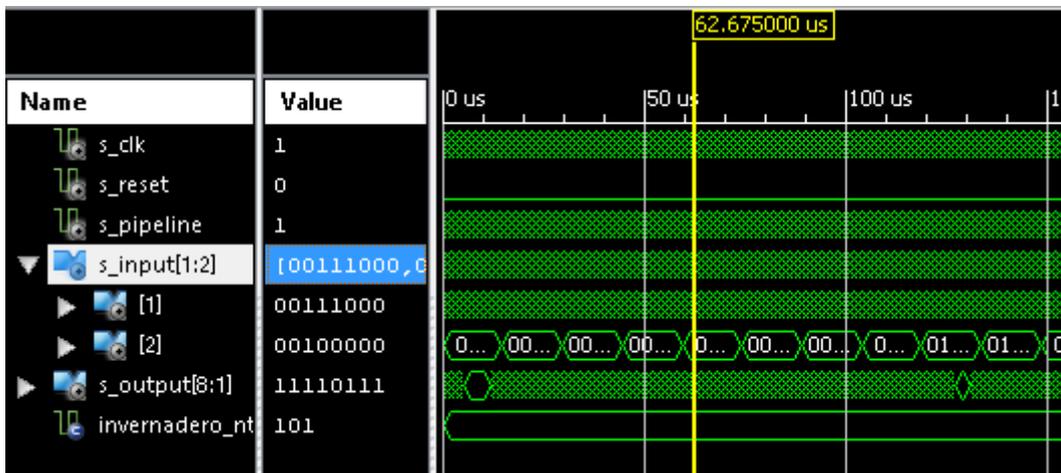


Figura B.5. Prueba 3 en el simulador de Xilinx.

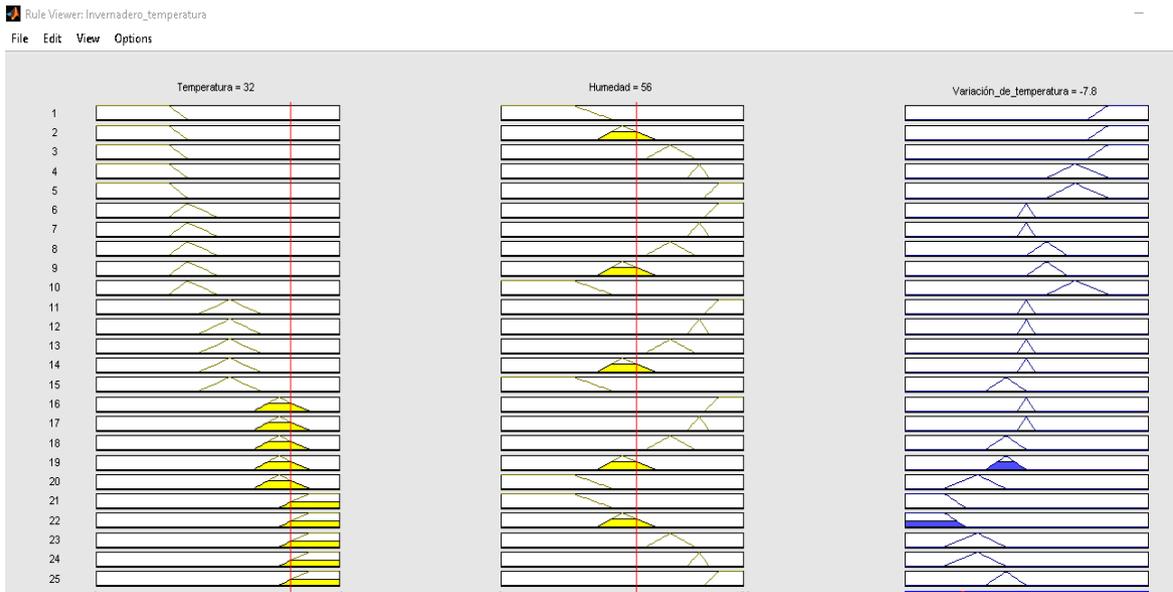


Figura B.6. Resultado 3 en la Toolbox de MATLAB.

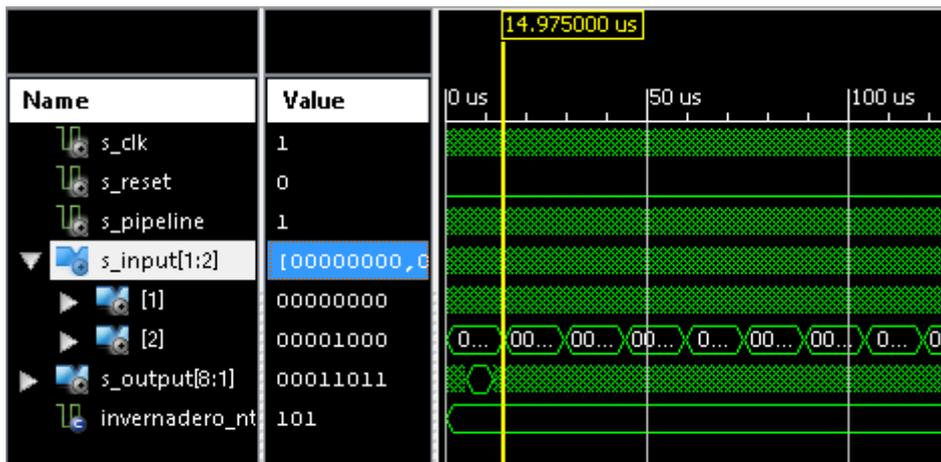


Figura B.7 Prueba 4 en el simulador de Xilinx.

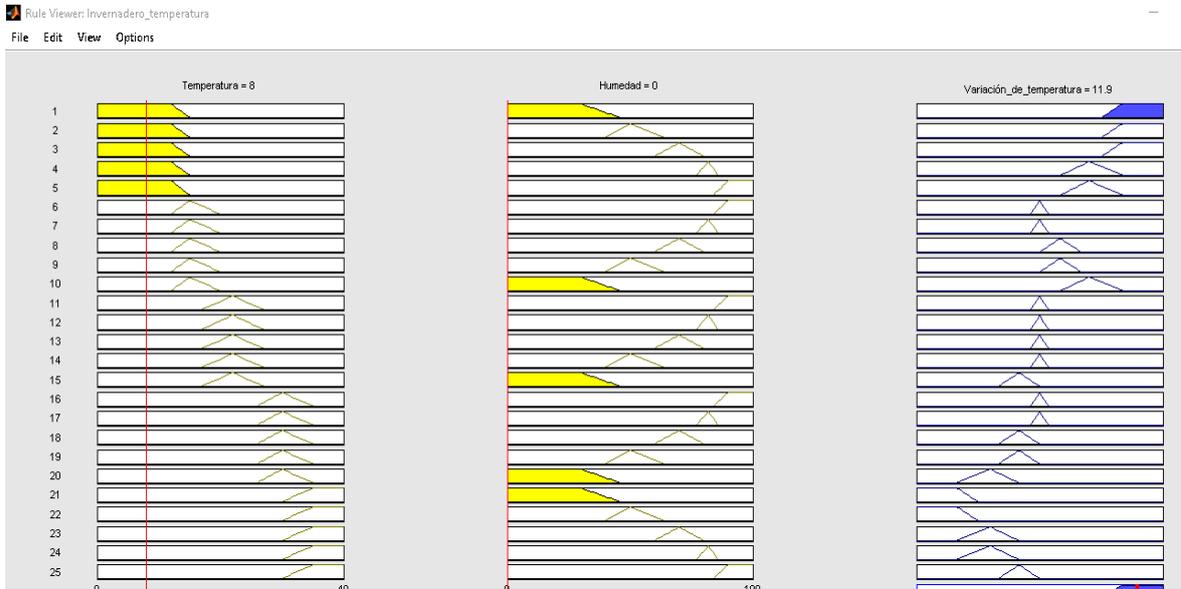


Figura B.8. Resultado 4 en la Toolbox de MATLAB.

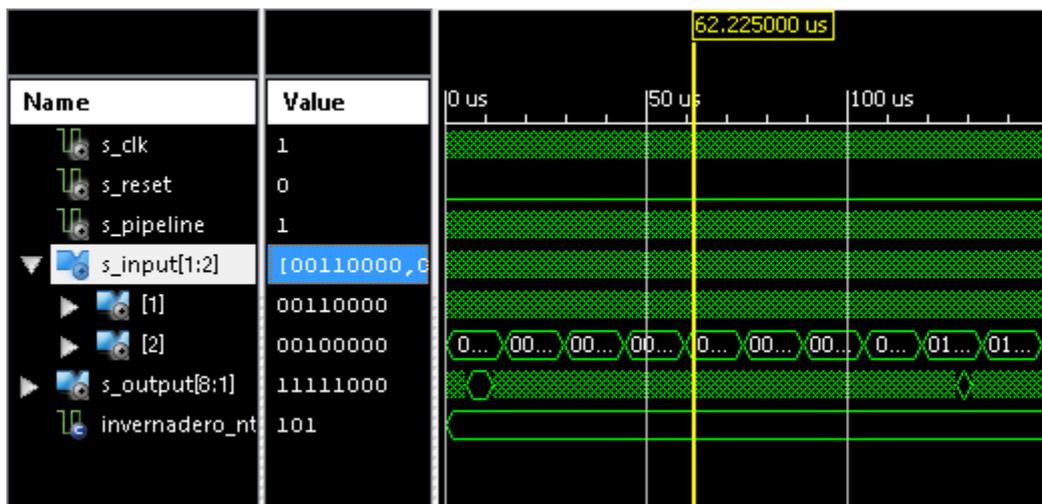


Figura B.9. Prueba 5 en el simulador de Xilinx.

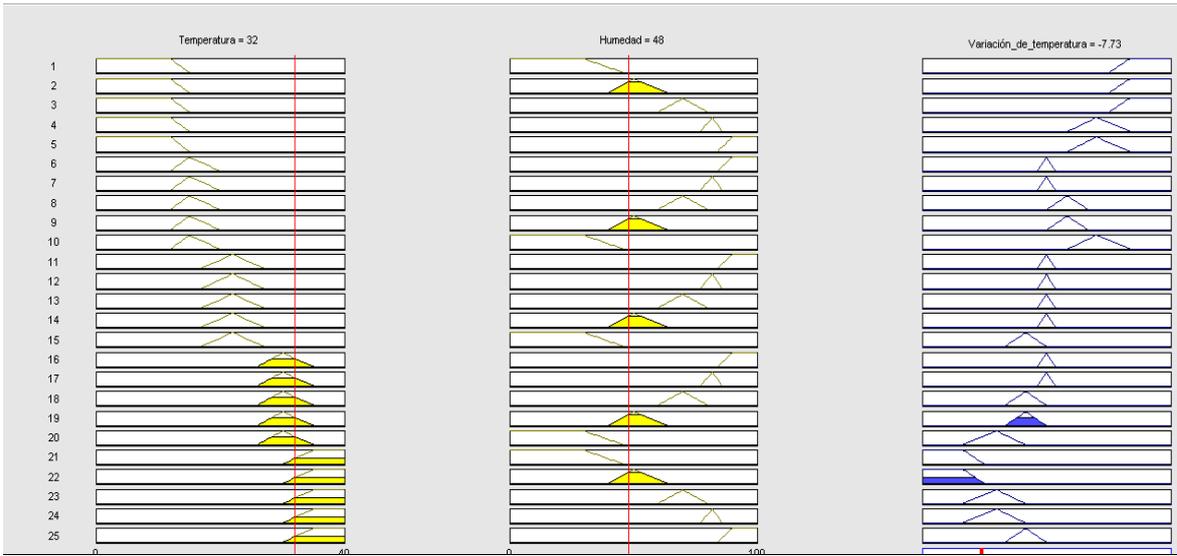


Figura B.10. Resultado 5 en la Toolbox de MATLAB.

## REFERENCIAS

- [1] INFOAGRO, “Principales tipos de invernaderos”, *Revista InfoAgro*, séptima edición, Diciembre 2016.
- [2] AGTECH(Noviembre 2017). El avance de la automatización en la agricultura, [Online]. Available: <http://www.redagricola.com>
- [3] VAISALA, “Las mediciones climáticas en los invernaderos garantizan un óptimo crecimiento de las plantas”, *Revista Vaisala*, 2013.
- [4] C. GONZÁLEZ, *Lógica Difusa: Una introducción práctica*. págs. 5–11, 2013.
- [5] M. GUIJARRO, E. TORTAJADA y F. GONZÁLEZ, “Sistema de riego inteligente borroso”, Universidad Complutense de Madrid, Madrid, 2007.
- [6] MATHWORKS. Fuzzy Logic Toolbox, [Online]. Available: <http://www.mathworks.com>
- [7] INSTITUTO DE MICROELECTRÓNICA DE SEVILLA (IMSE-CNM), “XFuzzy user guide”, Octubre 2001.
- [8] NATIONAL INSTRUMENTS. Módulo LabVIEW FPGA, [Online]. Available: <http://www.ni.com>
- [9] INSTITUTO UNIVERSITARIO DE MICROELECTRÓNICA APLICADA, “Tutorial Verilog”, Universidad de Las Palmas de Gran Canaria, España.
- [10] C. FERNÁNDEZ, I. FERNÁNDEZ, D. MOYA y M. V. LÓPEZ, “Valoración de inmuebles mediante técnicas de lógica difusa”, Universidad Complutense de Madrid, Madrid, 2008.

## TESIS

C. PRADOS, “Control de Dispositivos Externos, desde una FPGA, vía Bluetooth”, Tesis de grado en Ingeniería Electrónica Industrial y Automática, Escuela de Ingenierías Industriales, UV, Valladolid, Junio 2018.

D.A. JIMÉNEZ, “La importancia de la automatización en invernaderos para el cultivo de jitomate en el estado de Morelos”, Tesis de pregrado: Ingeniero Mecatrónico, Universidad Nacional Autónoma de México, México, DF, 2011.

J. AGUADO, “Desarrollo de un sistema automatizado para un invernadero”, Tesis de pregrado: Ingeniero Técnico Industrial, Universidad de Valladolid, 2012.

Y. MALDONADO, “Diseño, implementación y prueba de la etapa de fuzificación para sistemas difusos tipo 1 y tipo 2 en FPGA's”, Tesis de Maestría: Maestría en ciencias en Sistemas Digitales, Centro de Investigación y Desarrollo de Tecnología Digital, IPN, México, B.C., 2008.

## **ARTICULO**

J. FLORES, E. MEJÍA, J.I. MONTERO y A. ROJANO (2010). Análisis numérico del clima interior en un invernadero de tres naves con ventilación mecánica. *Agrociencia*, vol. 45 (No. 5). Recuperado de <http://www.scielo.org.mx>

J.J. PUERTA, (Octubre 2014). Control de temperatura y humedad de una maqueta de invernadero mediante tecnología Open Source.

D. GUZMÁN y V. M. CASTAÑO (2006). Lógica difusa en Ingeniería: Principios, aplicaciones y futuro. *Ciencia y Tecnología*, vol. 24 (No. 2).

A. CELA, J. SOTOMAYOR y F. SÁNCHEZ (Enero 2014). Diseño de un Controlador Fuzzy para guiado de un robot móvil. *EPN*, vol. 33 (No. 1).

J.C. HERRERA, M. L. OLVERA y M. T. LOZANO. Arquitectura básica para controladores de Lógica difusa a programarse en FPGAs. *Polibits*, vol. 14 (No. 1).

M. GUTIÉRREZ, A. I. BARRANCO, J. PRADO, J. A. PADILLA y F. J. PÉREZ (Noviembre 2016). Implementación de un sistema difuso en VHDL.