

CT-698-531
Don - 2013

XX(202095.1)



Centro de Investigación y de Estudios Avanzados
del Instituto Politécnico Nacional
Unidad Guadalajara

Control PSO-Neuronal para regular la temperatura de un invernadero

Tesis que presenta:

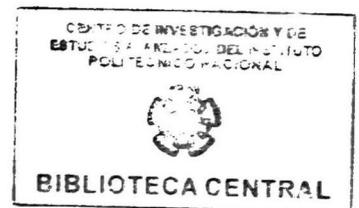
Raúl Avila Miranda

para obtener el grado de:

**Maestro en Ciencias
en la especialidad de:**

Ingeniería Eléctrica
Directores de Tesis

Dra. Ofelia Begovich Mendoza
Dr. José Javier Ruíz León



CLASS: CT00602
ADQUIS: CT-698-SS1
FECHA: 22-01-2013
PROYECTO: D02-2013
\$

Control PSO-Neuronal para regular la temperatura de un invernadero

**Tesis de Maestría en Ciencias
Ingeniería Eléctrica**

Por:

Raúl Avila Miranda

Licenciado en Matemáticas Aplicadas

Universidad Autónoma de Aguascalientes 2002-2010

Becario de Conacyt, expediente no. 243174

Directores de Tesis

Dra. Ofelia Begovich Mendoza

Dr. José Javier Ruíz León

CINVESTAV del IPN Unidad Guadalajara, Agosto de 2012.

Agradecimientos

A mis asesores por sus consejos y su guía.

A mi familia por su apoyo incondicional, especialmente a mi esposa.

Y a Conacyt por el apoyo económico, con el cual fue posible concluir mis estudios.

Resumen

En este trabajo se desarrolla una técnica de control para regular la temperatura interna de un invernadero durante el día. Esta técnica es la fusión de una red neuronal y el algoritmo de Optimización por Enjambre de Partículas (PSO): mediante un perceptrón multicapa recurrente (MLRP), entrenado con filtro de Kalman extendido (EKF), se realiza una predicción mediante la cual el algoritmo PSO calcula el tiempo en que los ventiladores deben ser encendidos y apagados, eliminando el exceso de temperatura no deseado y minimizando el tiempo en el que permanecen encendidos los ventiladores. Esta técnica puede ser generalizada y utilizada en cualquier sistema modelado con un MLRP cuya dinámica sea lenta y las entradas de control sean On-Off. El desempeño del algoritmo es probado mediante simulación.

Abstract

In this work it is developed a control technique to regulate the greenhouse's internal temperature during the day. This technique is the result of the fusion of a neural network and the particle swarm optimization algorithm (PSO): by means of a multilayer recurrent perceptron (MLRP), trained with the extended Kalman filter (EKF), it is computed a prediction through which the PSO algorithm calculate the time when the fans must be turn on and off, in order to eliminate the unwanted excess of heat and minimizing the lapse the fans remain turned on. This technique can be generalized and used in any system modeled by a MLRP which has a slow dynamic and where the control inputs are On-Off. The algorithm's performance is proved through simulation.

Índice

1. Introducción	2
1.1 Motivación	2
1.2 Estado del arte	3
1.3 Objetivos	4
1.4 Estructura de las tesis	5
2. Preliminares	6
2.1 Perceptrón Multicapa Recurrente Entrada-Salida	6
2.1.1 <i>Perceptrón</i>	6
2.1.2 <i>Perceptrón Multicapa</i>	7
2.2 Estructura de identificación serie-paralelo	9
2.3 Filtro de Kalman Extendido	10
2.4 El Perceptrón Multicapa entrenado con Filtro de Kalman Extendido	11
2.5 Optimización por Enjambre de Partículas	14
3. Preliminares para el entrenamiento de la red	16
3.1 Modelo matemático del invernadero	16
4. Entrenamiento de la Red Neuronal.....	23
4.1 Entrenamiento de la red	23
4.1.1 Estructura de la red	23
4.1.2 Parámetros de entrenamiento.....	24
4.1.3 Resultados de entrenamiento.....	26
4.2 Validación	28
5. Control PSO-Neuronal	33
5.1 Utilidad del algoritmo PSO	33
5.2 Función objetivo.....	36
5.3 Descripción de la técnica de control PSO-Neuronal	38
5.4 Simulación del control PSO-Neuronal.....	39
Conclusiones y trabajo futuro	43
Bibliografía	44

Capítulo 1

1. Introducción

1.1 Motivación

La creciente necesidad de producir cultivos de forma rápida y eficiente, así como protegerlos de las condiciones adversas a las cuales se encuentran vulnerables con los métodos de cultivo tradicionales, son aspectos fundamentales para el surgimiento de la agricultura protegida. Es por ello que el uso de los invernaderos recientemente ha tenido un importante papel en la escena agrícola mundial, como una solución viable al problema de la optimización de recursos naturales, como suelo y agua, obteniéndose además beneficios sustanciales como cultivos fuera de su ciclo natural y en menor tiempo, capaces de enfrentar con éxito plagas y enfermedades, logrando mejores rendimientos en un espacio reducido.

Estos beneficios se presentan debido a las condiciones climáticas apropiadas proporcionadas por el ambiente dentro del invernadero tales como temperatura, humedad, luminosidad y concentración de CO_2 . Esto sin mencionar la protección a plagas y enfermedades que la estructura provee.

Dichas condiciones climáticas óptimas se alcanzan mediante la manipulación adecuada de los mecanismos que inciden directa o indirectamente sobre las mismas, dicho de otra manera, mediante el correcto uso de los actuadores tales como sistemas de calefacción, humidificadores, ventiladores, y la apertura de ventanas, entre otros. La manipulación inadecuada de dichos actuadores puede desembocar en una producción pobre, o hasta en la pérdida total de la misma.

De aquí que es evidente la importancia de implementar técnicas modernas de control que de manera eficaz lleven al óptimo aprovechamiento del potencial genético inherente en cada semilla.

El creciente auge de la agricultura protegida exige modelos que describan el comportamiento de invernaderos con geometría variada, cultivos variados y en condiciones climáticas externas variadas.

1.2 Estado del arte

Se han formulado varios modelos matemáticos atendiendo aspectos muy específicos de la geometría del invernadero y de la clase de cultivo como se puede ver en (Straten *et al.*, 2011; Herrero *et al.*, 2007). Dichos modelo carecen de generalidad y muchas veces no describen de manera fidedigna el comportamiento del clima del invernadero, debido a la no-linealidad y el alto acoplamiento entre las variables a controlar.

En la actualidad existe una fuerte tendencia al uso de métodos alternos de control y optimización, conocidos como heurísticos. Dichos métodos son utilizados cuando los métodos matemáticos de análisis clásicos no son útiles o aplicables debido a la falta de modelos matemáticos explícitos o a la compleja obtención de estos.

Las Redes Neuronales y el algoritmo de optimización por enjambre de partículas, o PSO, por sus siglas en inglés (Particle Swarm Optimization), pertenecen a este conjunto de métodos.

Una de las principales características de las Redes Neuronales utilizadas en la identificación de sistemas, es la capacidad de capturar en los pesos sinápticos de su estructura, las relaciones no lineales entre los datos, y por ende la dinámica de los sistemas, la cual es muy difícil de modelar mediante las técnicas clásicas. Las Redes Neuronales, han sido utilizadas ampliamente para resolver problemas similares de otras áreas, en sistemas físicos, biológicos, químicos, económicos, etc.

Existen diversas aplicaciones de las redes neuronales estáticas a la identificación de modelos del clima de un invernadero (Salazar *et al.*, 2008; Bin Zhao *et al.*, 2010; Salazar *et al.*, 2010; Eredics *et al.*, 2010). Es el propósito de esta tesis desarrollar un modelo dinámico para la temperatura del invernadero, utilizando un perceptrón multicapa recurrente entrenado con el filtro de Kalman extendido, el cual mejora la convergencia del aprendizaje (Feldkamp *et al.*, 2003).

El algoritmo PSO es utilizado en la optimización de funciones, y es particularmente sencillo en su programación, con un costo computacional bajo, por esta característica lo hemos elegido para implementarlo en el algoritmo de control que se desarrollará posteriormente en esta tesis. En nuestro contexto, este algoritmo se ha utilizado para calcular los pesos de una red neuronal que modele el clima de un invernadero (Zhao *et al.*, 2010). Otra aplicación del PSO, en este mismo contexto, es para hacer seguimiento de

trayectoria de la temperatura mediante un sistema de calefacción regulable el cual permite un control continuo (Coelho *et al.*, 2005). Dicho control se calcula mediante una predicción en base a un modelo matemático. El núcleo central de esta tesis, es desarrollar una técnica de control, que proporcione soluciones eficientes a las necesidades actuales de la agricultura protegida de nuestra región, debido a que existen técnicas de control que solo son aplicables en invernaderos altamente tecnificados con características muy específicas, los cuales consumen mucha energía eléctrica.

1.3 Objetivos

El objetivo principal es desarrollar una técnica de control inteligente para regular la temperatura de un invernadero medianamente tecnificado, en el cual las entradas de control son ON-OFF.

La técnica aquí expuesta se enfoca en regular la temperatura interna de un invernadero mediante el uso de ventiladores, que tienen como función desplazar el aire caliente del interior al exterior del invernadero, y de este modo enfriar el ambiente interno. Por ello el control, para este análisis, funciona solo durante el día.

Es necesario señalar que esta técnica de control se puede generalizar y por ende, aplicar a todo lo referente al clima interno del invernadero.

La técnica se desarrolla en dos pasos:

- 1) Mediante información recolectada previamente, se entrena un perceptrón multicapa recurrente mediante la técnica de entrenamiento de filtro de Kalman, (Capítulo 4). Debido a que no se contaba con datos reales, estos fueron obtenidos con un modelo matemático como se explica en el Capítulo 3.
- 2) Una vez que la red neuronal fue entrenada, esta es utilizada para predecir la temperatura interna. Con esta predicción el algoritmo de PSO calcula los tiempos a los cuales los ventiladores deben ser encendidos y apagados, de tal forma que la temperatura sea regulada y el tiempo en que permanecen encendidos sea minimizado. Esto se desarrolla detalladamente en el Capítulo 5.

1.4 Estructura de las tesis

La tesis está organizada de la siguiente forma:

En el Capítulo 2 se describen los fundamentos matemáticos indispensable para comprender esta tesis.

En el Capítulo 3 se explica la manera en que se obtuvieron los datos con los cuales la red neuronal utilizada, fue entrenada.

En el Capítulo 4 se describe el procedimiento mediante el cual se entreno la red así como los resultados de dicho entrenamiento.

El Capítulo 5 describe le técnica de control PSO-Neuronal con la aplicación a la regulación de la temperatura de un invernadero.

Capítulo 2

2. Preliminares

En este capítulo se presentan las definiciones y conceptos básicos sobre el perceptrón multicapa, así como el filtro de Kalman extendido y el algoritmo de optimización por enjambre de partículas.

2.1 Perceptrón Multicapa Recurrente Entrada-Salida

2.1.1 Perceptrón

Un perceptrón es la forma más simple de red neuronal (Haykin 1994). Básicamente consiste en una sola neurona con pesos ajustables, una función de activación y un umbral como muestra la Ecuación (2.1), y la Figura 2.1.

$$v = \sum_{i=1}^p w_i x_i - \theta \tag{2.1}$$

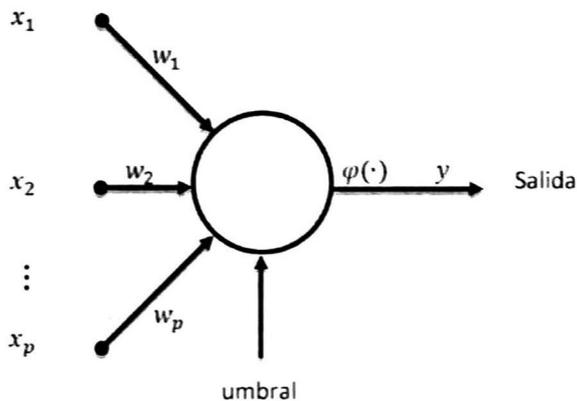


Figura 2.1 El perceptrón.

2.1.2 Perceptrón Multicapa

Un perceptrón multicapa se compone de varias capas donde a su vez cada capa está compuesta de varias neuronas como se muestra en la Figura 2.2.

Los elementos del perceptrón multicapa se muestran en la figura 2.3 y están dados por:

- Los subíndices i, j, k identifican las diferentes capas de la red.
- w_{ij} representa el peso que conecta la neurona i a la neurona j .
- v_j es el potencial de activación de la neurona j .
- $\varphi_j(\cdot)$ es la función de activación de la neurona j .
- θ_j es el umbral de la neurona j .
- $x_i(\cdot)$ representa el i -ésimo elemento de x .

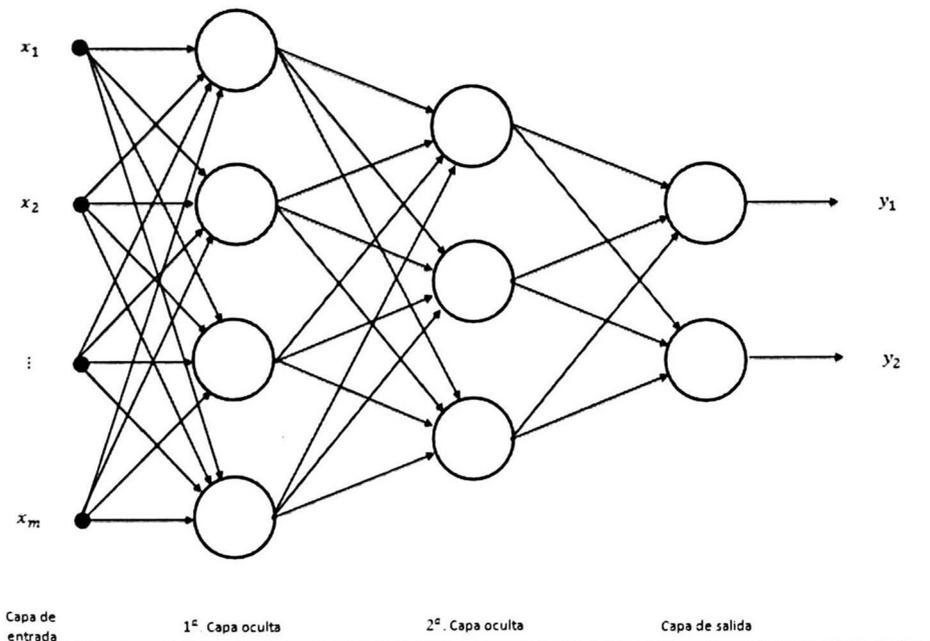


Figura 2.2 Red neuronal tipo perceptrón multicapa.

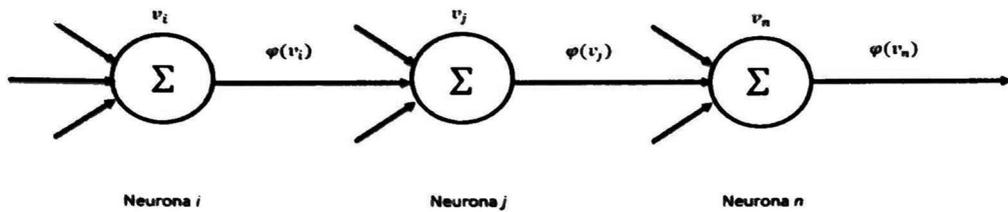


Figura 2.3 Elementos de una red neuronal tipo perceptrón multicapa.

En la figura 2.4 se muestra la arquitectura de un Perceptrón Multicapa Recurrente genérico del tipo entrada-salida. El modelo tiene solo una entrada que se aplica a la entrada de la red, aunque es posible agregar más entradas o perturbaciones. También incluye m de los retardos de la entrada; así mismo se retroalimenta a la red n de los retardos de la salida $y(k)$. De esta manera el comportamiento dinámico de la red queda determinado por:

$$y(k+1) = F(y(k), \dots, y(k-n+1), u(k), \dots, u(k-m+1)) \quad (2.2)$$

donde F es una función no lineal.

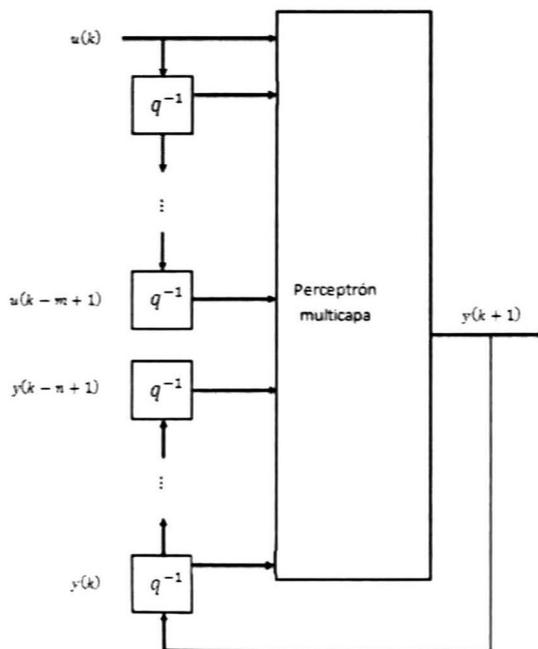


Figura 2.4 Modelo entrada-salida.

2.2 Estructura de identificación serie-paralelo

De acuerdo con la configuración de un sistema de identificación, puede haber dos estructuras de identificación: estructura paralela, y la estructura serie-paralelo (Pham *et al.*, 1995).

En la estructura serie-paralelo, la red neuronal y el sistema reciben las mismas señales de entrada externa pero la entrada del sistema es parte de las entradas de la red neuronal, como muestra la Figura 2.5. El sistema y la red neuronal no son dos procesos independientes. El comportamiento dinámico de la red es afectado por el sistema.

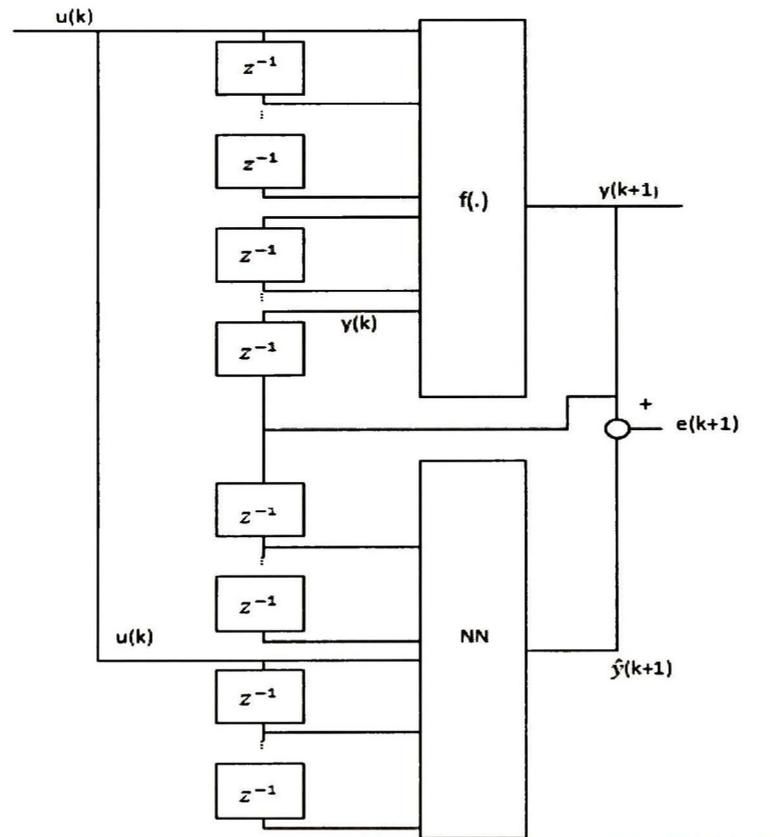


Figura 2.5 Estructura de identificación serie-paralelo.

2.3 Filtro de Kalman Extendido

Considérese un sistema dinámico no lineal descrito por el siguiente modelo en espacio de estado:

$$\begin{aligned}w(k+1) &= f(k, w(k)) + u(k) \\ y(k) &= h(k, w(k)) + v(k)\end{aligned}\tag{2.3}$$

donde $u(k)$ y $v(k)$ son ruidos blancos independientes entre si, gaussianos, con matrices de covarianza R y Q respectivamente.

La idea básica del filtro de Kalman extendido, es linealizar el modelo en espacio de estados (2.3) cada instante de tiempo alrededor del espacio estimado más reciente, el cual puede ser tomado como $\hat{w}(k)$ o $\hat{w}^-(k)$. Una vez obtenido el modelo lineal, se aplican las ecuaciones del filtro de Kalman (Haykin *et al.*, 2001).

Sean

$$\begin{aligned}F_{k+1,k} &= \left. \frac{\partial f(k, w)}{\partial w} \right|_{w=\hat{w}(k)} \\ H(k) &= \left. \frac{\partial h(k, w)}{\partial w} \right|_{w=\hat{w}(k)^-}\end{aligned}$$

Inicialización

$$\begin{aligned}\hat{w}_0 &= E[w_0], \\ P_0 &= E[(w_0 - E[w_0])(w_0 - E[w_0])^T].\end{aligned}$$

Para $k = 1, 2, \dots$, se calcula

Propagación del estado estimado

$$\hat{w}^-(k) = f(k, \hat{w}(k-1));$$

Propagación de la covarianza del error

$$P(k)^- = F_{k,k-1}P(k-1)F_{k,k-1}^T + Q(k-1);$$

Matriz de ganancia de Kalman

$$K(k) = P(k)^-H^T(k)[H(k)P(k)^-H^T(k) + R(k)];$$

Actualización del estado estimado

$$\hat{w}(k) = \hat{w}^-(k) + K(k)y(k) - h(k, \hat{w}^-(k));$$

Actualización de la covarianza del error

$$P(k) = (I - K(k)H(k))P^-(k);$$

2.4 El Perceptrón Multicapa entrenado con Filtro de Kalman Extendido

Considérese un perceptrón multicapa con L pesos sinápticos y m nodos de salida. Sea $w(k)$ el vector de pesos de la red en la interacción k . La red neuronal puede ser modelada como sigue (Sánchez et al., 2006) :

$$\begin{aligned} w(k+1) &= w(k) \\ \hat{y}(k) &= h(w(k), u(k)) \end{aligned} \tag{2.4}$$

donde el vector de pesos $w(k)$ corresponde al estado, $u(k)$ es el vector de entradas a la red neuronal, $\hat{y}(k)$ es el vector de salidas de la red neuronal y $h(\cdot)$ es la función no lineal de salida de la red neuronal.

Considerando las ecuaciones de 2.4 y el filtro de Kalman extendido expuesto en la sección 2.3, tenemos entonces que las ecuaciones necesarias para el desarrollo del algoritmo de entrenamiento de la red son:

$$\begin{aligned} K(k) &= P(k)H^T [R(k) + H(k)P(k)H^T(k)]^{-1} \\ w(k+1) &= w(k) + K(k)[y(k) - \hat{y}(k)] \\ P(k+1) &= P(k) - K(k)H(k)P(k) + Q(k) \end{aligned} \tag{2.5}$$

Donde $P(k) \in \mathfrak{R}^{L \times L}$ representan la matriz de covarianza del error de predicción; $w \in \mathfrak{R}^L$ es el vector de pesos (estados); L es el número total de pesos de la red neuronal; $y \in \mathfrak{R}^m$ es el vector de salidas; $\hat{y} \in \mathfrak{R}^m$ es la salida de la red neuronal; $K \in \mathfrak{R}^{L \times m}$ es la matriz de ganancia de Kalman; $Q \in \mathfrak{R}^{L \times L}$ es la matriz de covarianza del ruido del proceso; $R \in \mathfrak{R}^{m \times m}$ es la matriz de covarianza del ruido del medición; $H \in \mathfrak{R}^{m \times L}$ es la matriz de derivadas de la salida de la red con respecto a los pesos definida de la siguiente manera:

$$H_{ij}(k) = \left[\frac{\partial \mathcal{J}_i(k)}{\partial w_j(k)} \right]_{w(k)=\hat{w}(k+1)}, i = 1 \dots m, j = 1 \dots L \quad (2.6)$$

Considérese el Perceptrón Multicapa de la Figura 2.5 con p entradas y noc neuronas ocultas; con el vector de pesos dado por:

$$w = [w_{10}^{(1)} w_{11}^{(1)} \dots w_{1p}^{(1)} w_{20}^{(1)} w_{21}^{(1)} \dots w_{2p}^{(1)} \dots w_{noc0}^{(1)} w_{noc1}^{(1)} \dots w_{nocp}^{(1)} w_{10}^{(2)} w_{11}^{(2)} \dots w_{1noc}^{(2)}]^T$$

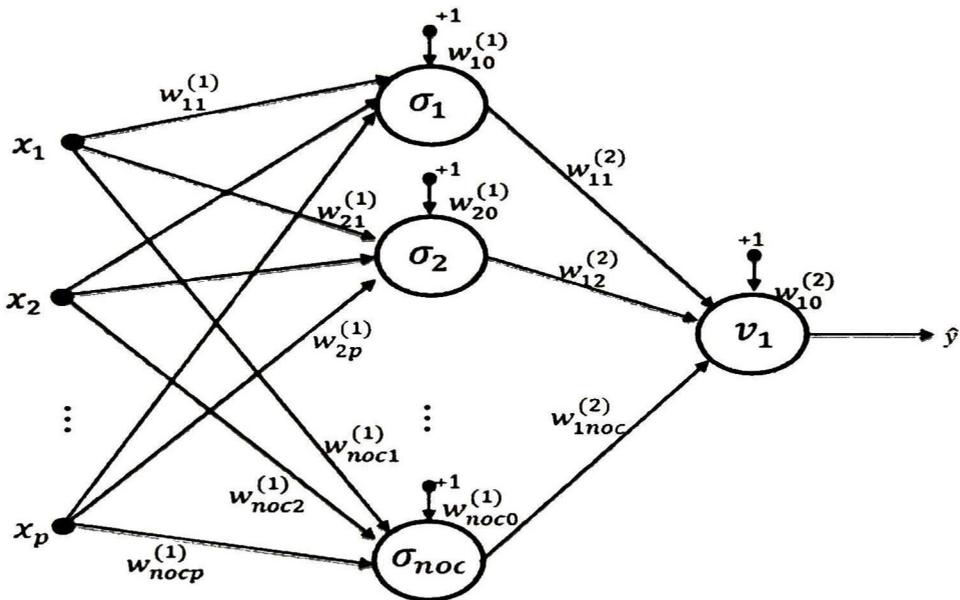


Figura 2.6 Esquema de un perceptrón multicapa

L es el número total de pesos y se define como:

$$L = (p + 1) \times noc + (noc + 1) \times m \quad (2.7)$$

donde m es el número total de salidas de la red.

La salida del perceptrón multicapa mostrado en la Figura 2.6 está definido por:

$$\sigma_i = \frac{1}{1 + e^{-n_i}} \quad i = 1 \dots noc$$

$$n_i = \sum_{j=0}^p w_{ij}^{(1)} x_j \quad x_0 = +1 \quad (2.8)$$

$$v_1 = \sum_{k=0}^{noc} w_{1k}^{(2)} \sigma_k \quad \sigma_0 = +1$$

$$\hat{y} = v_1$$

De aquí que la matriz H tiene la forma:

$$\frac{\partial \hat{y}}{\partial w} = \left[\frac{\partial \hat{y}}{\partial w_{10}^{(1)}} \quad \frac{\partial \hat{y}}{\partial w_{11}^{(1)}} \quad \dots \quad \frac{\partial \hat{y}}{\partial w_{1noc}^{(2)}} \right]$$

donde:

$$\frac{\partial \hat{y}}{\partial w_{10}^{(1)}} = \frac{w_{11}^{(2)} e^{-n_1}}{(1 + e^{-n_1})^2} x_0; \quad \frac{\partial \hat{y}}{\partial w_{11}^{(1)}} = \frac{w_{11}^{(2)} e^{-n_1}}{(1 + e^{-n_1})^2} x_1; \quad \dots \quad \frac{\partial \hat{y}}{\partial w_{1p}^{(1)}} = \frac{w_{11}^{(2)} e^{-n_1}}{(1 + e^{-n_1})^2} x_p;$$

$$\frac{\partial \hat{y}}{\partial w_{20}^{(1)}} = \frac{w_{12}^{(2)} e^{-n_2}}{(1 + e^{-n_2})^2} x_0; \quad \frac{\partial \hat{y}}{\partial w_{21}^{(1)}} = \frac{w_{12}^{(2)} e^{-n_2}}{(1 + e^{-n_2})^2} x_1; \quad \dots \quad \frac{\partial \hat{y}}{\partial w_{2p}^{(1)}} = \frac{w_{12}^{(2)} e^{-n_2}}{(1 + e^{-n_2})^2} x_p;$$

⋮

$$\frac{\partial \hat{y}}{\partial w_{noc0}^{(1)}} = \frac{w_{1noc}^{(2)} e^{-n_{noc}}}{(1 + e^{-n_{noc}})^2} x_0; \quad \frac{\partial \hat{y}}{\partial w_{noc1}^{(1)}} = \frac{w_{1noc}^{(2)} e^{-n_{noc}}}{(1 + e^{-n_{noc}})^2} x_1; \quad \dots \quad \frac{\partial \hat{y}}{\partial w_{nocp}^{(1)}} = \frac{w_{1noc}^{(2)} e^{-n_{noc}}}{(1 + e^{-n_{noc}})^2} x_p;$$

$$\frac{\partial \hat{y}}{\partial w_{10}^{(2)}} = 1; \quad \frac{\partial \hat{y}}{\partial w_{11}^{(2)}} = \frac{1}{1 + e^{-n_1}}; \quad \frac{\partial \hat{y}}{\partial w_{12}^{(2)}} = \frac{1}{1 + e^{-n_2}}; \quad \dots \quad \frac{\partial \hat{y}}{\partial w_{1noc}^{(2)}} = \frac{1}{1 + e^{-n_{noc}}}$$

2.5 Optimización por Enjambre de Partículas

El algoritmo de optimización por enjambre de partículas o PSO, por sus siglas en ingles (Particle Swarm Optimization), fue desarrollado en (Kennedy *et al.*, 1995), como un algoritmo de optimización basado en modelos de simulación social. El algoritmo emplea una población de puntos de búsqueda que se mueven aleatoriamente en el espacio de búsqueda. La población es llamada enjambre y los individuos son llamados partículas. El enjambre se define como el conjunto:

$$S = \{s_1, s_2, \dots, s_N\},$$

de N partículas o soluciones candidato, definidas como:

$$s_j = (s_{j1}, s_{j2}, \dots, s_{jn})^T \in A, \quad j = 1, 2, \dots, N,$$

donde n es el número de componentes de la partícula. Se asume que la función objetivo, $F(s)$, existe para todos los puntos en A . Por ende cada partícula tiene un único valor, $F_j = F(s_j) \in Y$. Se asume que las partículas se mueven en el espacio de búsqueda, A , de un modo iterativo, el cual es posible ajustando su posición mediante un cambio apropiado, llamado velocidad, denotado por:

$$v_j = (v_{j1}, v_{j2}, \dots, v_{jn})^T, \quad j = 1, 2, \dots, N.$$

La velocidad también se adapta para que las partículas sean potencialmente capaces de visitar toda la región A . Sea $s_j(r)$ y $v_j(r)$ la posición y velocidad de la j -ésima partícula en la r -ésima iteración respectivamente. La velocidad se actualiza en base a información obtenida en previos pasos del algoritmo. Esto es implementado en términos de una memoria, donde cada partícula puede almacenar la mejor posición que ha tenido durante toda su búsqueda. Para este propósito el PSO almacena todas las mejores posiciones de todas las partículas:

$$P_s = \{p_1, p_2, \dots, p_N\},$$

donde

$$p_j = (p_{j1}, p_{j2}, \dots, p_{jN})^T \in A, \quad j = 1, 2, \dots, N,$$

Estas posiciones se definen como:

$$p_j = \operatorname{argmin}_r F_j(r).$$

El algoritmo aproxima el mínimo global con la mejor posición visitada por todas las partículas. Sea $*$ el índice de la mejor posición con el valor mínimo de la función en P_s en la r -ésima iteración

$$p^*(r) = \operatorname{argmin}_r F(p_j(r)).$$

De aquí que el PSO se define por las siguientes ecuaciones:

$$v_{jl}(r+1) = v_{jl}(r) + c_1 R_1 (p_{jl}(r) - s_{jl}(r)) + c_2 R_2 (p_j^*(r) - s_{jl}(r)), \quad (2.9)$$

$$s_{jl}(r+1) = s_{jl}(r) + v_{jl}(r+1), \quad j = 1, 2, \dots, N \quad l = 1, 2, \dots, n \quad (2.10)$$

donde; R_1 y R_2 son variables aleatorias uniformemente distribuidas en el intervalo $[0,1]$; y c_1 y c_2 , son factores de peso, también conocidos como parámetros cognitivo y social, respectivamente.

Los dos últimos términos de la Ecuación (2.9), le permiten a la partícula realizar una búsqueda alrededor de su mejor posición p_j y la mejor posición de todo el enjambre p^* . El primer término de la derecha de la Ecuación (2.9) permite a cada partícula realizar una búsqueda global explorando nuevas partes del espacio de búsqueda.

Capítulo 3

3. Preliminares para el entrenamiento de la red

En este capítulo se expone la forma en que se obtuvieron los datos para el entrenamiento de la red utilizada para la predicción del clima del invernadero. Primeramente se presenta el modelo matemático del cual se obtuvieron los datos. Posteriormente se proponen las funciones para emular las perturbaciones y entradas de control que el modelo requiere.

3.1 Modelo matemático del invernadero

Debido a la falta de datos reales para el entrenamiento, estos se obtuvieron de un modelo matemático reportado en (Straten *et al.*, 2011), dicho modelo fue desarrollado para un invernadero donde se cultivan lechugas.

Este modelo considera la temperatura, humedad, la concentración de CO_2 y la producción de materia seca del cultivo de lechuga. Para nuestro análisis solo se considera la temperatura.

La ecuación que describe la temperatura en base a este modelo es:

$$\dot{T}_a^c = \frac{1}{C_{cap}} u_q - (C_{cap}^{vent} u_v + C_{g_o}^{trans})(T_a^c - T_o^c) + C_{o_g}^{rad} I_0 \quad (3.1)$$

donde :

T_a^c : Temperatura dentro del invernadero;

T_o^c : Temperatura fuera del invernadero;

I_0 : Radiación solar fuera del invernadero;

u_v : Tasa de ventilación a través de las ventanas;

C_{cap}^{vent} : Capacidad de calor por unidad de volumen;

$C_{g_o}^{trans}$: Transferencia de calor global a través de la cobertura;

$C_{o_g}^{rad}$: Coeficiente de carga de calor debido a la radiación;

C^{cap} : Volumen del invernadero.

El modelo (3.1), no incluye el enfriamiento por la acción de ventiladores, por ende se modifico para que fuese incluida esta dinámica. Realizando la modificación, el modelo con el que se trabajó fue el siguiente:

$$\dot{T}_a^c = \frac{1}{C^{cap}} u_q - (C_{cap}^{vent}(u_v + u_{venti}) + C_{g_o}^{trans})(T_a^c - T_o^c) + C_{o_g}^{rad} I_0 \quad (3.2)$$

El termino añadido en (3.2) es u_{venti} que denota la entrada y salida de aire causada por los ventiladores que ocasiona un enfriamiento del aire del interior del invernadero.

El valor de los términos $C_{cap}^{vent} = 1290 \text{ J m}^{-3} \text{ }^\circ\text{C}^{-1}$, $C_{g_o}^{trans} = 6.1 \text{ Wm}^{-2} \text{ }^\circ\text{C}^{-1}$, $C_{o_g}^{rad} = .2$ y $C^{cap} = 4.1 \text{ m}^3 \text{ m}^{-2}$, se tomo como la referencia mencionada.

Los comportamientos de T_o^c y I_0 fueron emulados como una primera aproximación con funciones senoidales, como se muestran en las figuras 3.1 y 3.2, esto con el fin de simplificar el análisis.

Dichas funciones son:

$$\begin{aligned} T_o^c &= 7 \text{sen} \left(\pi/17t - \frac{\pi}{2} \right) + 25; \\ I_0 &= 350 \text{sen} \left(\pi/16t - \frac{\pi}{2} \right) + 350. \end{aligned} \quad (3.3)$$

Los coeficientes de las funciones (3.3) fueron sintonizados para obtener información realista.

Debido a que el comportamiento de la temperatura se aproxima a una función senoidal solo en un rango muy específico el cual se muestra en las Figuras 3.1 y 3.2, todo el análisis representa una buena aproximación en dicho rango. Este rango de operación es de aproximadamente 11 horas que varía según la época del año, este periodo abarca desde que sale el sol por la mañana hasta que comienza a meterse.

Los rangos se eligieron en base a las temperaturas máxima y mínima de la región de Jalisco para la época de primavera.

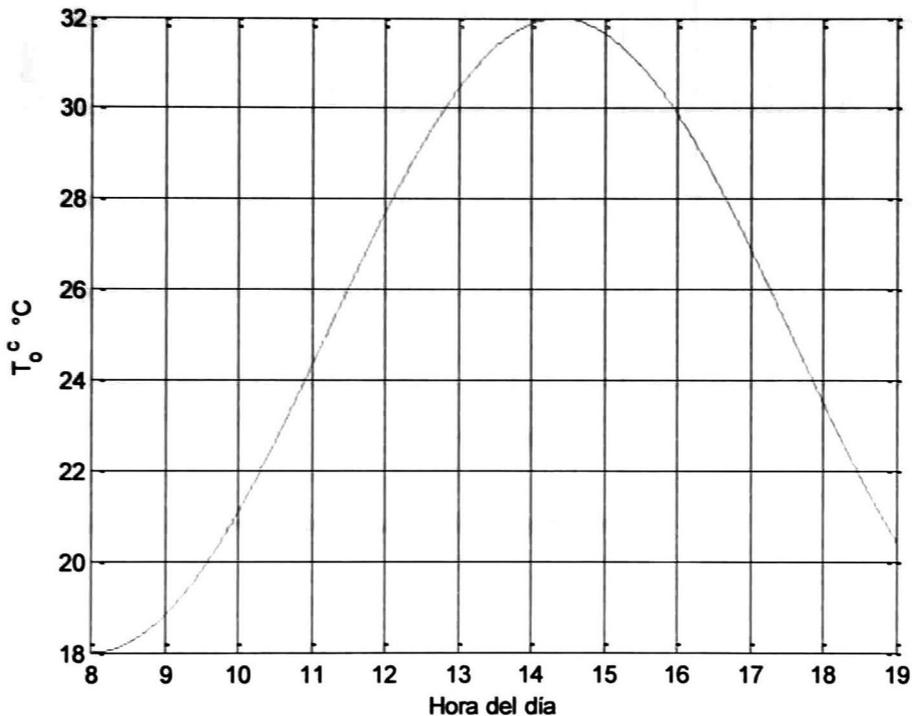


Figura 3.1 Temperatura fuera del invernadero.

Se considero un desfase entre la radiación solar y la temperatura, es decir, se considero que sus máximos se alcanzan a diferentes horas.

Los rangos para la radiación se eligieron en los mismos términos que en la temperatura.

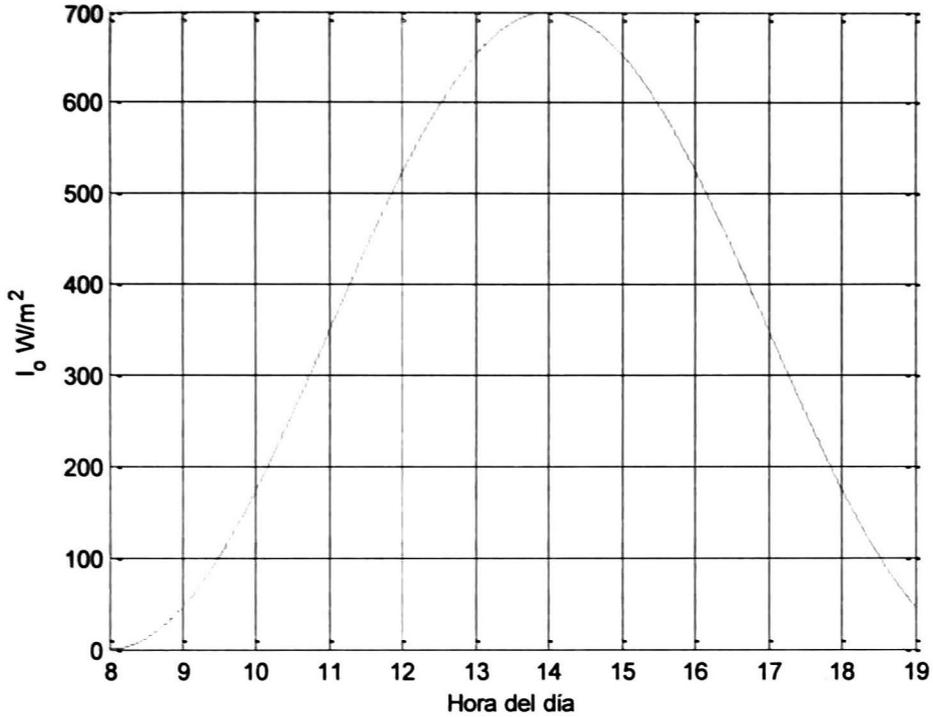


Figura 3.2 Radiación solar fuera del invernadero.

Para u_v se considera un control obtenido en base a como se opera empíricamente en los invernaderos de la región. Este control se muestra en la Figura 3.3, esto una vez más para simplificar el análisis. Este control en realidad se sintetiza considerando las otras variables del invernadero como humedad y la concentración de CO_2 .

La dinámica del modelo 3.1 con T_0^c , I_0 y u_v como se muestran en las Figuras 3.1, 3.2 y 3.3, respectivamente, sin los ventiladores, se puede observar en la Figura 3.4.

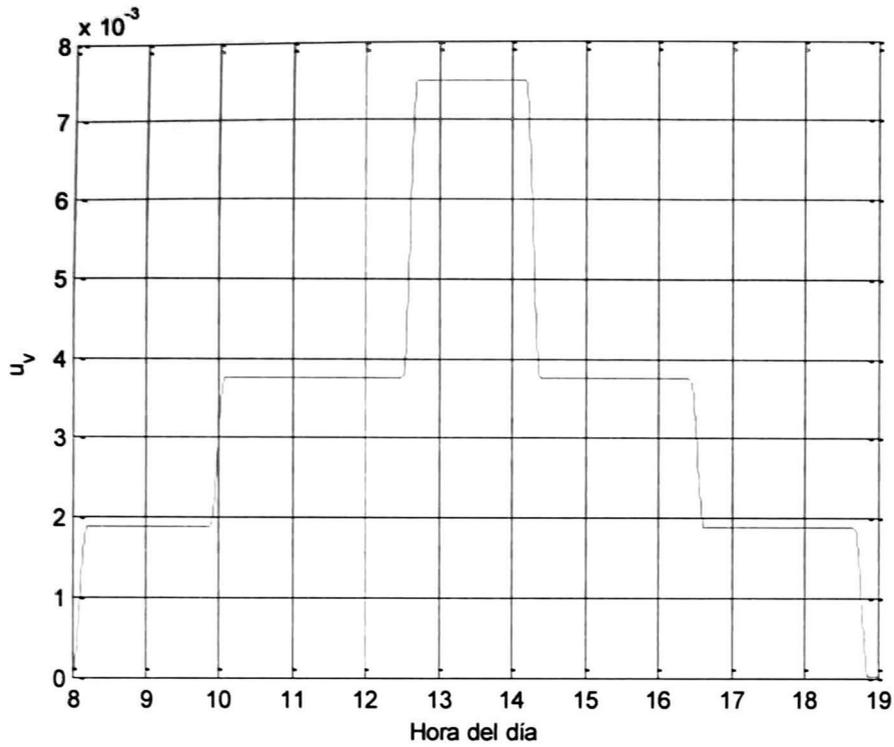


Figura 3.3 Tasa de ventilación a través de las ventanas.

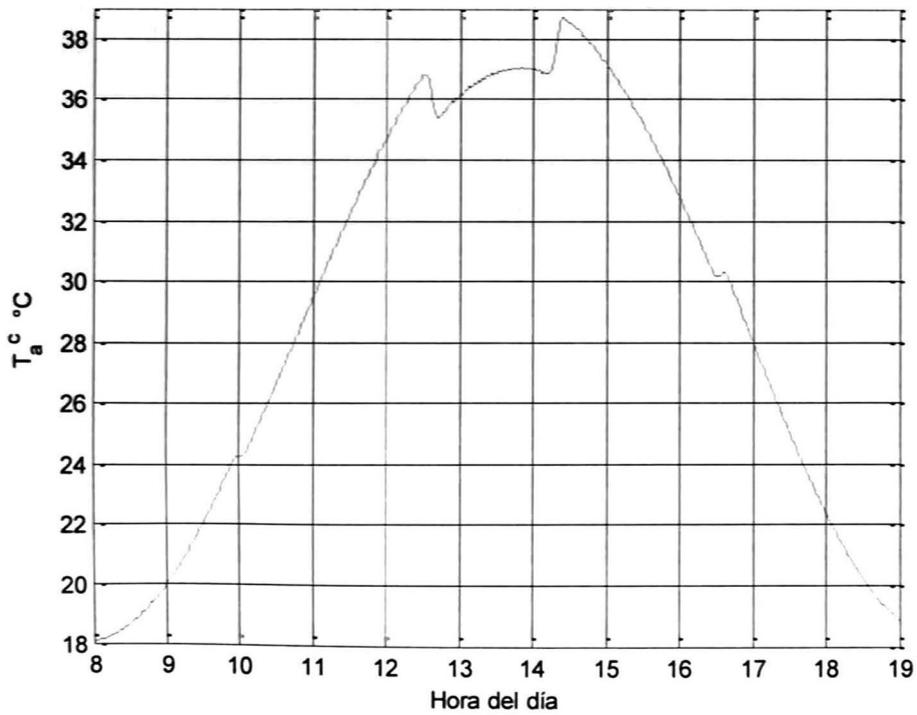


Figura 3.4 Temperatura dentro del invernadero

En lo que respecta a u_{venti} , debido a que no se dispone de un modelo para los ventiladores, los parámetros del control fueron sintonizados de tal modo que el efecto sobre la temperatura fuese realista.

La Figura 3.5 muestra el encendido y apagado de los ventiladores. Es básicamente un escalón suave, es decir, no discontinuo, esto para obtener una dinámica lenta y suave como se esperaría ocurriera en la realidad.

El efecto de los ventiladores sobre el sistema se puede observar en la Figura 3.5, en ella se muestra el comportamiento del sistema con los ventiladores apagados y el comportamiento con los ventiladores encendidos. Dicha simulación se realizó con una condición inicial $T_a^c(0) = T_o^c(0) = 18^\circ C$, es decir, para el análisis se desprecia el calor acumulado que el invernadero conserva del día anterior.

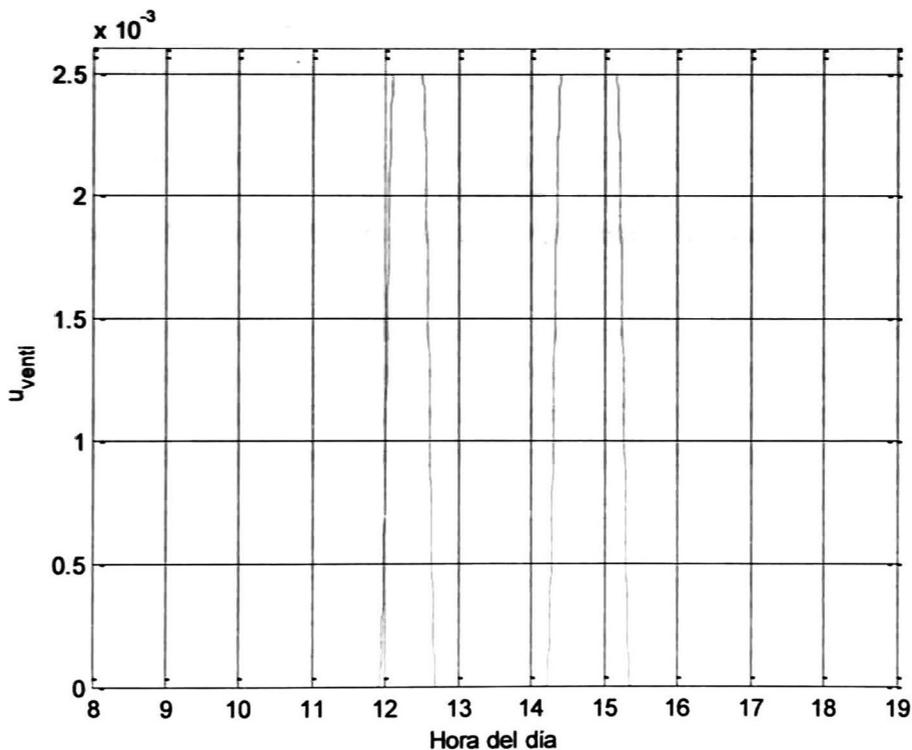


Figura 3.5 Encendido y apagado de los ventiladores.

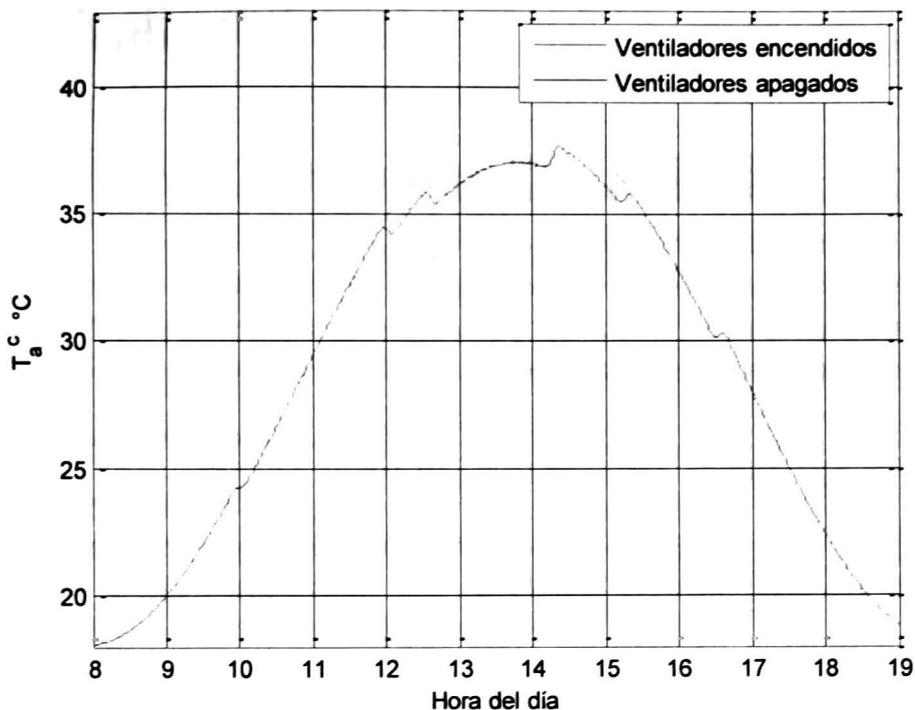


Figura 3.5 Efecto de los ventiladores sobre la temperatura dentro del invernadero.

Un control para los ventiladores se obtiene en esta tesis en los capítulos siguientes. La red neuronal realiza la predicción de la temperatura interna del invernadero, y el algoritmo PSO propone un control suficientemente satisfactorio para los ventiladores. Esto es posible gracias a la dinámica lenta del sistema.

Capítulo 4

4. Entrenamiento de la Red Neuronal

En este capítulo se desarrolla el entrenamiento de la red neuronal para la predicción de la temperatura del invernadero, así como su validación. Primeramente se describe la estructura que tendrá la red neuronal; posteriormente se eligen los parámetros de entrenamiento, y finalmente se evalúa el desempeño de la red neuronal.

4.1 Entrenamiento de la red

4.1.1 Estructura de la red

La estructura de la red neuronal fue sintonizada a prueba y error, partiendo de una estructura simple hasta alcanzar una complejidad suficiente para que los resultados fueran satisfactorios (Sánchez et al., 2004).

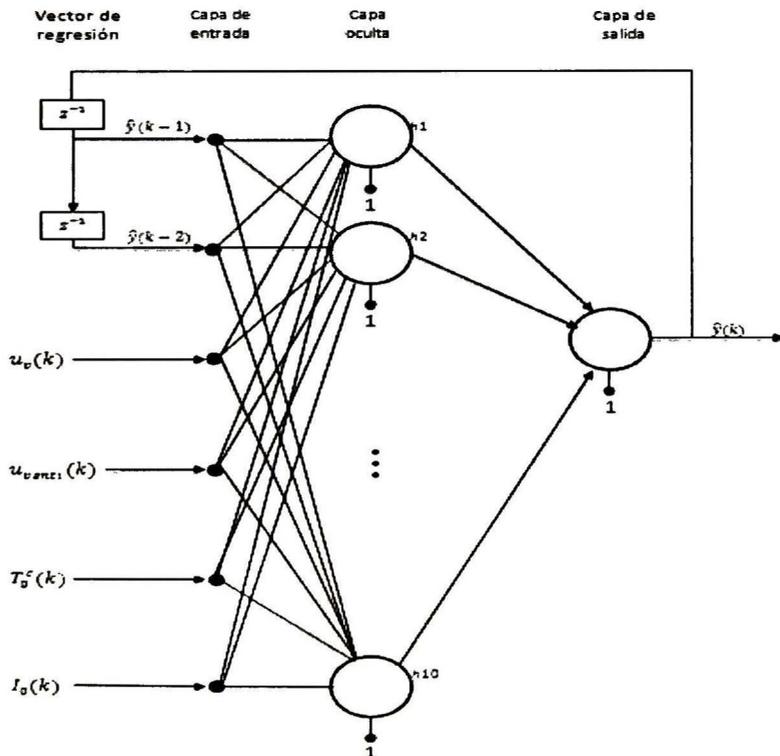


Figura 4.1 Estructura de la red neuronal utilizada.

La estructura consta de una capa oculta con diez neuronas en dicha capa y una función de activación como en (2.8) del Capítulo 2, una neurona en la capa de salida con función de activación lineal y 2 retrasos de la salida hacia la entrada, como se muestra en la Figura 4.1.

4.1.2 Parámetros de entrenamiento

El método de entrenamiento para el perceptrón multicapa fue el filtro de Kalman extendido expuesto en el Capítulo 2. Los parámetros iniciales para el entrenamiento fueron $p_0 = 800$, $q_0 = r_0 = 3 \times 10^{-3}$, los cuales proporcionan una condición suficiente para la convergencia de los pesos según esta demostrado en (Alanis 2004).

El entrenamiento del perceptrón multicapa se realizó fuera de línea, con las funciones 3.3 del Capítulo 3 mostradas en las figuras 3.1 y 3.2 del mismo capítulo, esto para la temperatura externa y la radiación solar, respectivamente. Para la tasa de ventilación de las ventanas y el encendido y apagado de los ventiladores, la red se entrenó con los datos propuestos mostrados en las figuras 4.2 y 4.3, respectivamente.

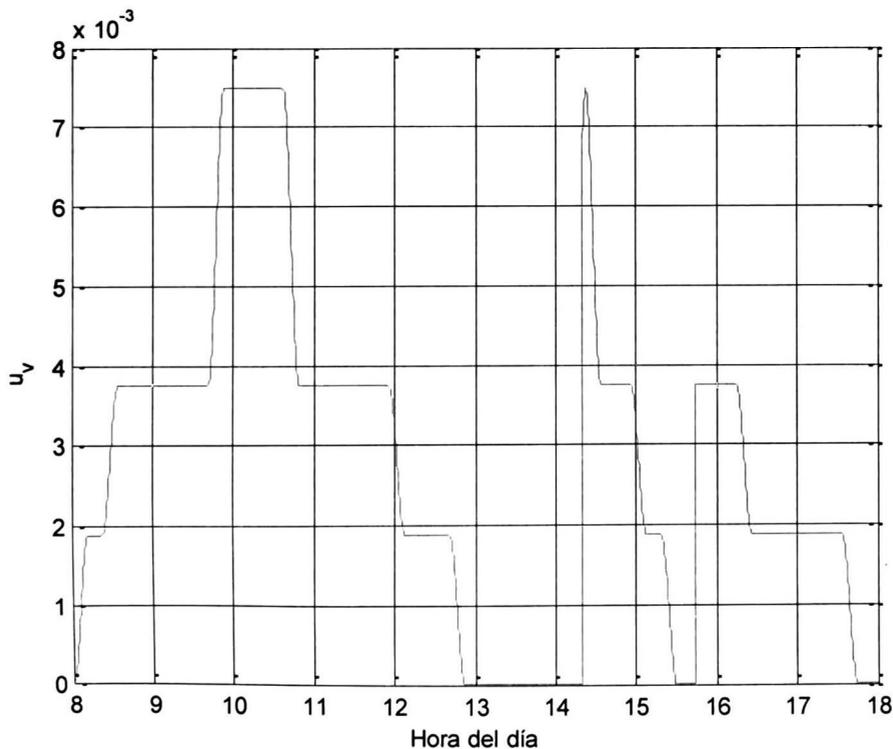


Figura 4.2 Tasa de ventilación a través de las ventanas.

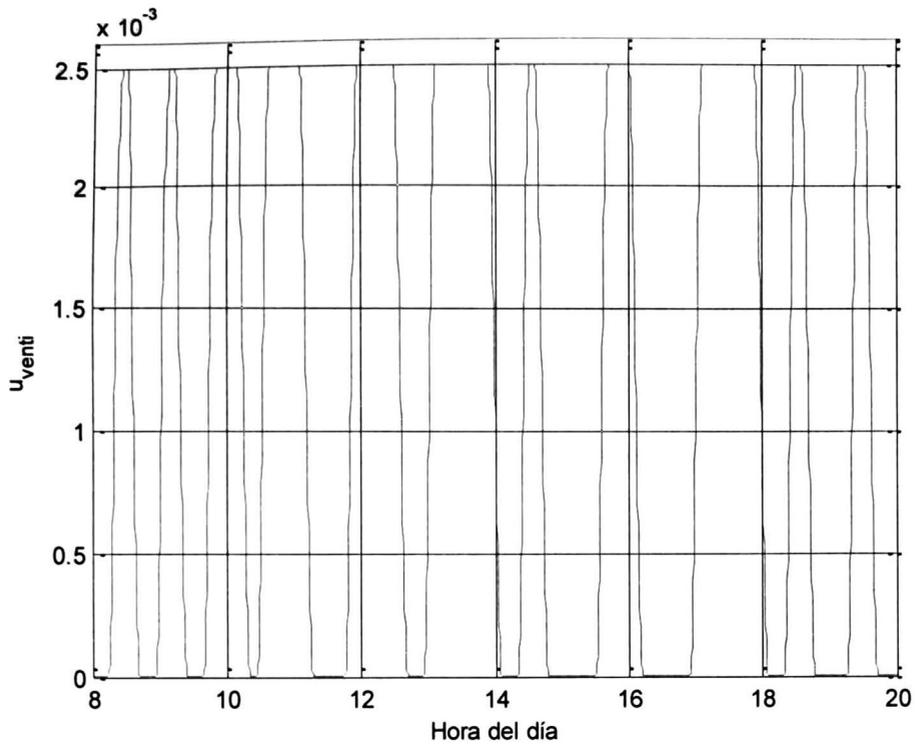


Figura 4.3 Encendido y apagado de los ventiladores.

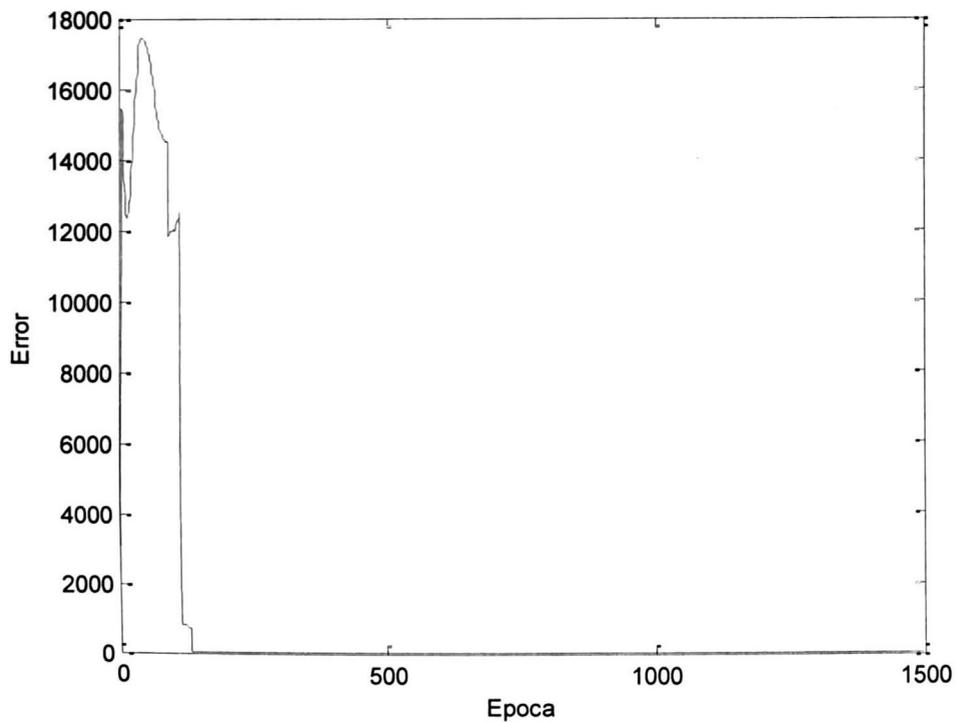


Figura 4.4 Convergencia del error.

La red neuronal fue entrenada con 2561 datos que equivalen a aproximadamente 10 horas de operación, esto con un tamaño de paso de .025 que equivale aproximadamente a 14 segundos. El criterio de paro del entrenamiento fue en base al error cuadrático medio, obteniendo así un desempeño satisfactorio con un error cuadrático medio de 2.3207×10^{-5} en 1137 épocas como se muestra en la Figura 4.4.

4.1.3 Resultados de entrenamiento

De acuerdo a (2.7) de Capítulo 2, el número de pesos sinápticos es 81, cuyos valores son:

$$W^{(1)} = \begin{bmatrix} w_{10}^{(1)} & w_{11}^{(1)} & w_{12}^{(1)} & w_{13}^{(1)} & w_{14}^{(1)} & w_{15}^{(1)} & w_{16}^{(1)} \\ w_{20}^{(1)} & w_{21}^{(1)} & w_{22}^{(1)} & w_{23}^{(1)} & w_{24}^{(1)} & w_{25}^{(1)} & w_{26}^{(1)} \\ w_{30}^{(1)} & w_{31}^{(1)} & w_{32}^{(1)} & w_{33}^{(1)} & w_{34}^{(1)} & w_{35}^{(1)} & w_{36}^{(1)} \\ w_{40}^{(1)} & w_{41}^{(1)} & w_{42}^{(1)} & w_{43}^{(1)} & w_{44}^{(1)} & w_{45}^{(1)} & w_{46}^{(1)} \\ w_{50}^{(1)} & w_{51}^{(1)} & w_{52}^{(1)} & w_{53}^{(1)} & w_{54}^{(1)} & w_{55}^{(1)} & w_{56}^{(1)} \\ w_{60}^{(1)} & w_{61}^{(1)} & w_{62}^{(1)} & w_{63}^{(1)} & w_{64}^{(1)} & w_{65}^{(1)} & w_{66}^{(1)} \\ w_{70}^{(1)} & w_{71}^{(1)} & w_{72}^{(1)} & w_{73}^{(1)} & w_{74}^{(1)} & w_{75}^{(1)} & w_{76}^{(1)} \\ w_{80}^{(1)} & w_{81}^{(1)} & w_{82}^{(1)} & w_{83}^{(1)} & w_{84}^{(1)} & w_{85}^{(1)} & w_{86}^{(1)} \\ w_{90}^{(1)} & w_{91}^{(1)} & w_{92}^{(1)} & w_{93}^{(1)} & w_{94}^{(1)} & w_{95}^{(1)} & w_{96}^{(1)} \\ w_{100}^{(1)} & w_{101}^{(1)} & w_{102}^{(1)} & w_{103}^{(1)} & w_{104}^{(1)} & w_{105}^{(1)} & w_{106}^{(1)} \end{bmatrix}$$

$$= \begin{bmatrix} 13.2207437 & -7.66674479 & 5.9465977 & 1.20418894 & -5.72832566 & -0.45565363 & 0.19110087 \\ 10.6078626 & -8.17765118 & 5.51878294 & 0.57186213 & -3.08976968 & -0.49428829 & 0.11881142 \\ 11.9718958 & -7.27168453 & 2.57188706 & 0.74113195 & 0.44132578 & 0.3874538 & 0.0120171 \\ -0.07733757 & 3.7793194 & -1.87170938 & -0.75422765 & 1.68202934 & -0.02510765 & 0.01047012 \\ -4.4573543 & 4.1066714 & -2.91397301 & 6.00709584 & -6.23944626 & -0.68582096 & 0.0102369 \\ 1.46058536 & 6.38814229 & -3.93423286 & -4.45099476 & 4.88537358 & 0.03911609 & -0.04759235 \\ 3.38115902 & -2.02691846 & -0.38794524 & 1.21483558 & 4.50688426 & 0.64421904 & 0.07360346 \\ -6.94641309 & 3.70262312 & -1.76171549 & 2.01176432 & -1.64538133 & 0.04859926 & 0.00549394 \\ 0.39964752 & 7.48513124 & -4.62922113 & -0.77786336 & 0.5025945 & 0.73569231 & 0.08663365 \\ -4.49932332 & 6.33184119 & -3.13180233 & -2.69739546 & 3.71700904 & -0.25066365 & -0.07842315 \end{bmatrix}$$

$$W^{(2)} = \begin{bmatrix} w_{10}^{(2)} \\ w_{11}^{(2)} \\ w_{12}^{(2)} \\ w_{13}^{(2)} \\ w_{14}^{(2)} \\ w_{15}^{(2)} \\ w_{16}^{(2)} \\ w_{17}^{(2)} \\ w_{18}^{(2)} \\ w_{19}^{(2)} \end{bmatrix} = \begin{bmatrix} -0.9135692 \\ -0.3219393 \\ -0.2289601 \\ -0.4709363 \\ 1.1844932 \\ 1.4695905 \\ 0.2868162 \\ -0.2242242 \\ 1.0178174 \\ 0.8577001 \end{bmatrix}$$

De aquí que:

$$\hat{y}(k) = \hat{T}_a^c(k) = \sum_{i=1}^{10} w_{1i}^{(2)} \varphi \left(\sum_{j=0}^6 w_{ij}^{(1)} x_j(k) \right) + w_{10}^{(2)};$$

donde

$$\chi = \begin{bmatrix} x_0(k) \\ x_1(k) \\ x_2(k) \\ x_3(k) \\ x_4(k) \\ x_5(k) \\ x_6(k) \end{bmatrix} = \begin{bmatrix} 1 \\ \hat{y}(k-1) \\ \hat{y}(k-2) \\ u_v(k) \\ u_{venti}(k) \\ T_o^c(k) \\ I_0(k) \end{bmatrix}.$$

O en su forma matricial

$$\hat{y}(k) = \hat{T}_a^c(k) = ([W^{(1)} \chi \ 1])W^{(2)T}$$

En la Figura 4.5 se muestra la temperatura real y la estimada resultante de la identificación.

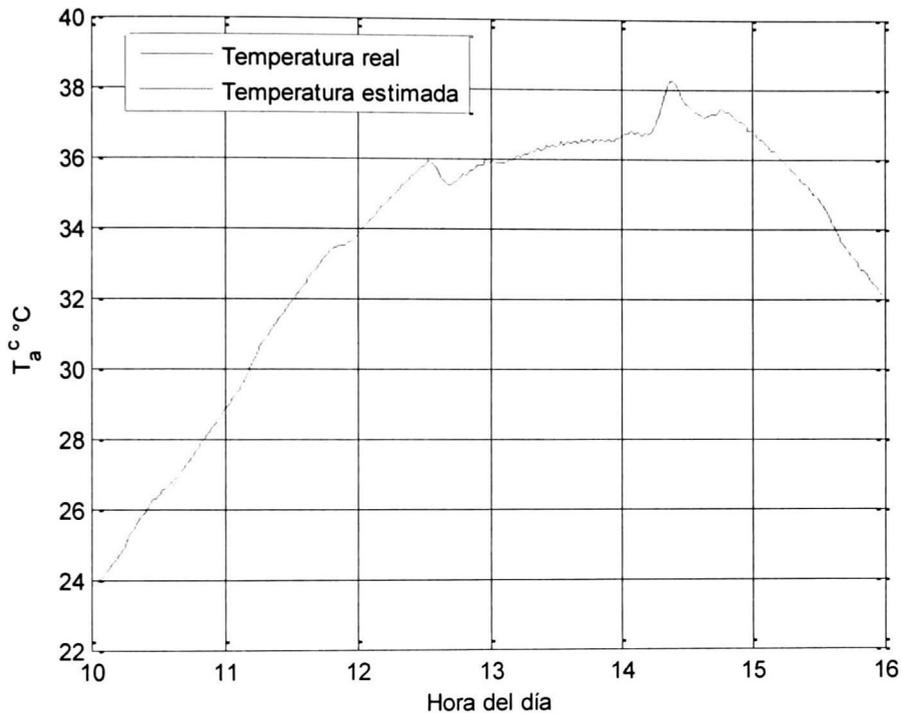


Figura 4.5 Temperatura real y estimada.

4.2 Validación

El siguiente paso es comprobar que la red neuronal entrenada efectivamente fue capaz de abstraer el comportamiento dinámico del sistema en cuestión. Para esto se alimenta a la red neuronal con datos con los que no fue entrenada y se compara su comportamiento con el esperado.

Esta validación, se realiza en dos etapas, la primera consistió en verificar que la red neuronal se comporte adecuadamente en ausencia de ventiladores, es decir, con los ventiladores apagados, y la segunda con los ventiladores prendidos pero simulando un día de operación normal.

. Para la primera validación la red neuronal será alimentada, en lo que respecta a la temperatura externa y la radiación, con los mismos datos con los que fue entrenada la red. Los datos de la tasa de ventilación se muestran en la Figura 4.6. Los ventiladores como se mencionó antes permanecerán apagados.

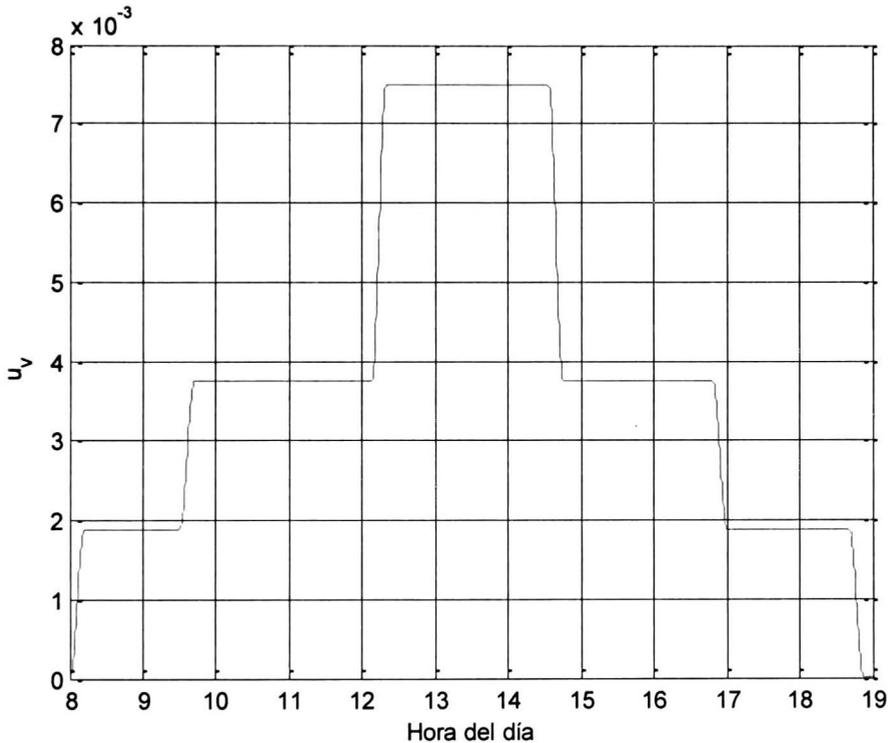


Figura 4.6 Tasa de ventilación a través de las ventanas.

La Figura 4.7 muestra los resultados de la simulación para la validación de la temperatura del interior del invernadero con los datos antes mencionados. Dicha validación obtuvo un error cuadrático medio de 4.9214×10^{-5} , el cual es bastante satisfactorio.

Para la segunda validación, la temperatura externa y la radiación serán las mismas que para la primera validación, solo serán diferentes la tasa de ventilación y el encendido y apagado de los ventiladores, mostradas en las Figuras 4.8 y 4.9, respectivamente.

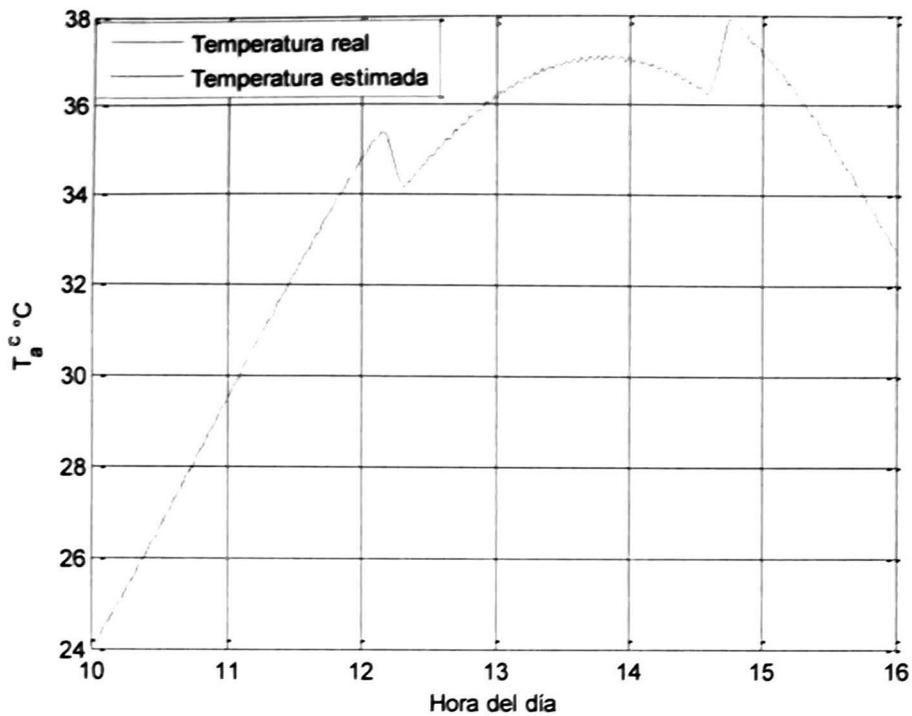


Figura 4.7 Temperatura real y estimada para la validación 1.

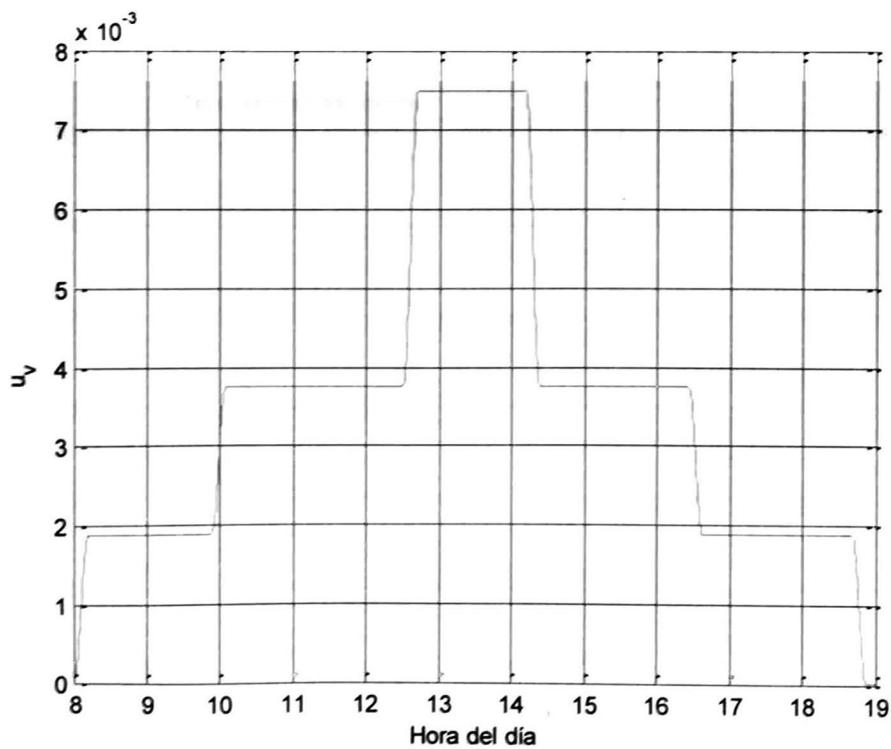


Figura 4.8 Tasa de ventilación a través de las ventanas.

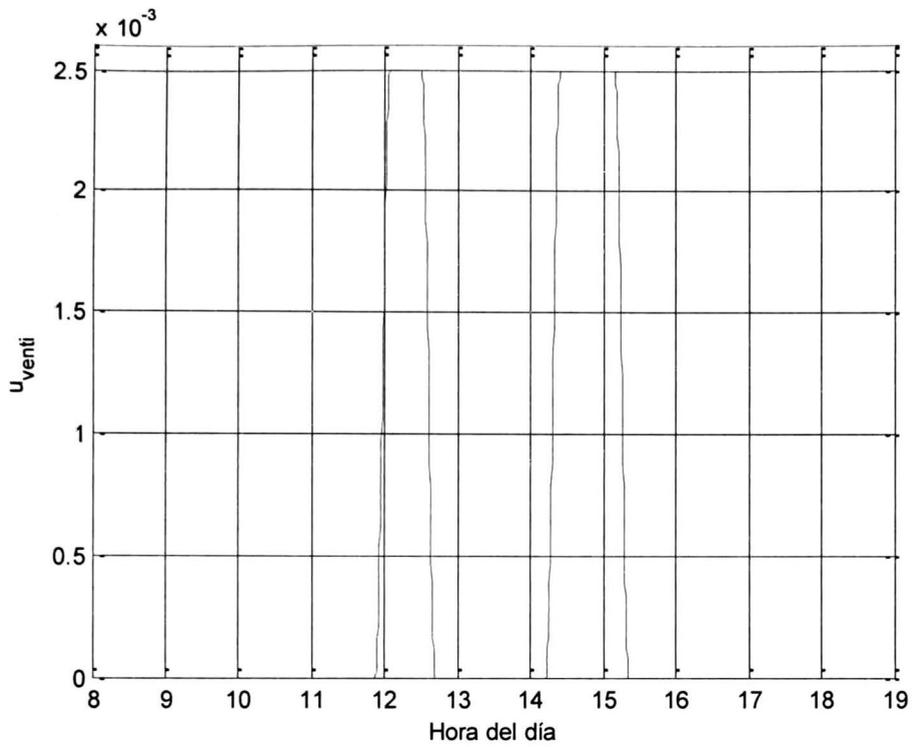


Figura 4.9 Encendido y apagado de los ventiladores

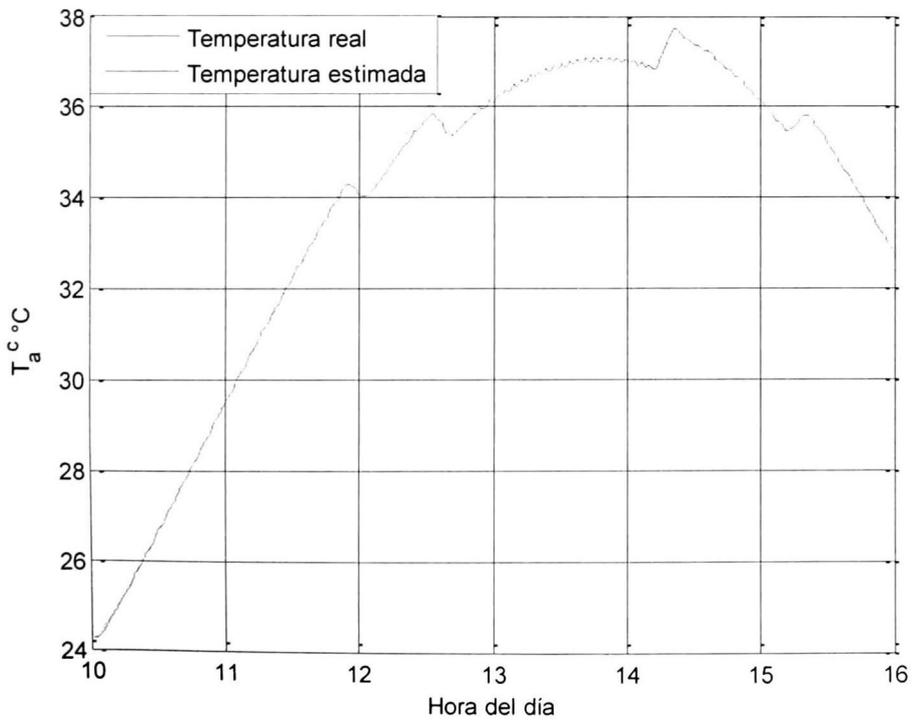


Figura 4.10 Temperatura real y estimada para la validación 2.

La Figura 4.10 muestra los resultados de la simulación para la validación de la temperatura con los datos antes mencionados, obteniendo un error cuadrático medio de 4.1132×10^{-5} , el cual sigue siendo bastante satisfactorio.

En base a todos los resultados anteriores podemos concluir que la red neuronal tiene un buen desempeño, siendo una opción viable para la predicción de la temperatura interior de un invernadero.

Es necesario aclarar que la estructura obtenida en este análisis puede variar cuando pretenda implementarse en un prototipo real. En dicho caso se tendrá que sintonizar la forma de la estructura de la misma manera que se realizó en este análisis, siendo esta similar aunque seguramente más compleja que la expuesta aquí.

Otro aspecto que hay que señalar, es que la predicción de la temperatura interna presupone que se tiene información de la temperatura externa y de la radiación, esto para fines prácticos, pero en una implementación habrá que proporcionar estos datos. Esto se realizara mediante redes neuronales, es decir, se predecirá la temperatura externa y la radiación mediante una estructura neuronal similar a la aquí expuesta.

Capítulo 5

En este capítulo se desarrolla una técnica de control basada en un perceptrón multicapa como predictor y el algoritmo PSO como herramienta para determinar el control para regular la temperatura interna de un invernadero.

5. Control PSO-Neuronal

5.1 Utilidad del algoritmo PSO

Antes de desarrollar esta técnica de control es necesario describir la utilidad del algoritmo PSO en nuestro contexto. Para esto recordemos que nuestro objetivo es regular la temperatura interna de un invernadero, teniendo solamente control sobre el encendido y apagado de los ventiladores, pero no se ha explicado hasta ahora el criterio bajo el cual debe ser regulada dicha temperatura.

La Figura 5.1 muestra la predicción de la temperatura interna, durante un periodo de operación de 11 horas, dada por la red neuronal bajo las siguientes condiciones: la temperatura externa y la radiación como se muestran en las figuras 3.1 y 3.2 del Capítulo 3, los ventiladores apagados y la apertura de las ventanas, u_v , como se muestra en la Figura 4.8 del Capítulo 4. En dicha figura (5.1) puede observarse que existe una diferencia entre la temperatura externa e interna. Ahora, dependiendo del cultivo en desarrollo en el invernadero, esta diferencia puede o no estresar a las plantas, por ejemplo, la genética del jitomate le permite a esta planta tolerar más temperatura, no así con la lechuga, la cual tiene menos tolerancia a altas temperaturas. Por ende, los rangos de temperatura óptimos para un invernadero se deciden en base al cultivo que este desarrollándose dentro de este.

Generalmente se utiliza una cota máxima de temperatura por encima de la cual el cultivo se considera estresado, dicha cota está dada en términos de la diferencia entre temperatura externa y la temperatura interna, por ejemplo, para el jitomate es aceptable una diferencia no mayor a 8 °C, esto dependiendo también de la época del año.

Debido a que nuestro análisis supone que tenemos un cultivo de lechugas, consideraremos $6\text{ }^{\circ}\text{C}$ como cota máxima de la diferencia entre las temperaturas. De aquí que el objetivo de nuestro control es mantener la diferencia entre la temperatura externa e interna por debajo de esta cota.

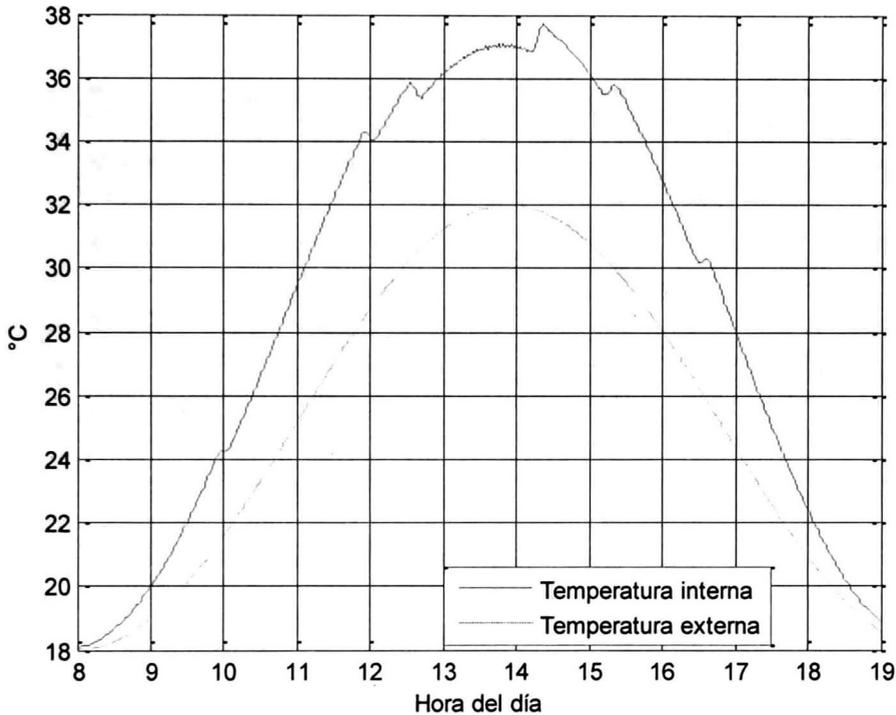


Figura 5.1 Temperatura interna

La Figura 5.2 muestra la diferencia entre la temperatura externa e interna mostradas en la Figura 5.1. Como puede observarse en dicha figura existen dos “picos” de temperatura por encima de 6 grados, que desearíamos bajar, esto mediante el control de los ventiladores.

Ahora, debido a la dinámica lenta del sistema en cuestión, el efecto de los ventiladores sobre la temperatura es también lento. La Figura 5.3 muestra el retardo que tiene el efecto de los ventiladores sobre la temperatura, en esta simulación los ventiladores se encendieron justo en el instante que la diferencia entre las temperaturas rebasó lo 6°C y permanecieron encendidos el tiempo que dicha diferencia excedió los 6°C .

Como puede observarse permanecieron picos de temperatura no deseada, sobre todo para el segundo pico, debido a que no se tomó en cuenta la dinámica lenta de la respuesta. Por ello un control que pretenda eliminar completamente o casi completamente estos picos no deseados de temperatura debe contemplar dicha dinámica.

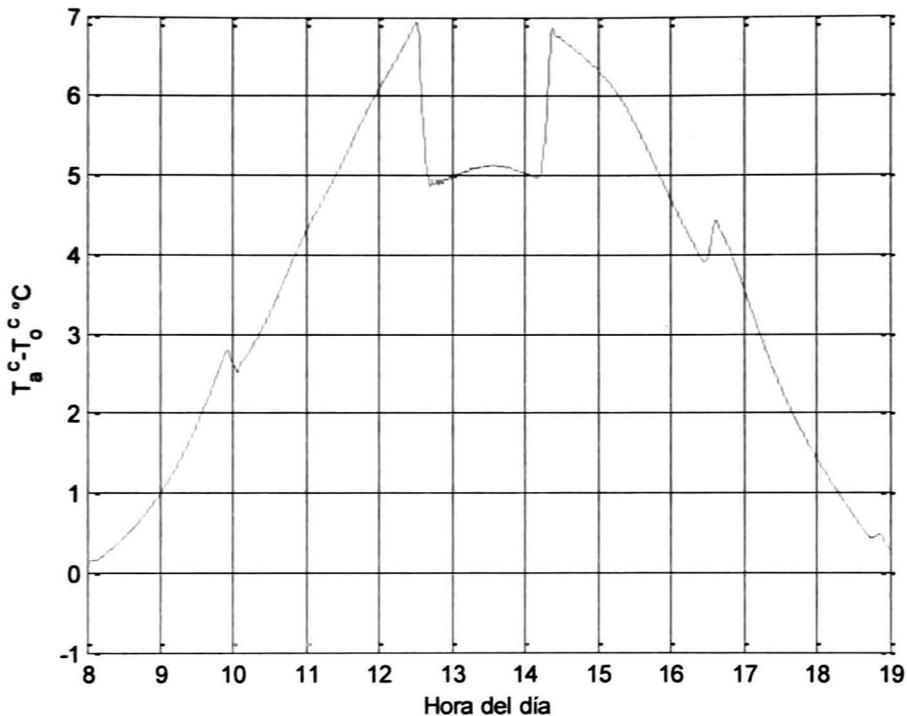


Figura 5.2 Diferencia entre la temperatura interna y externa del invernadero con los ventiladores apagados.

Otro factor importante a considerar para la elección del control, es el tiempo en el cual los ventiladores están encendidos, debido a que más tiempo encendidos implica más costo.

Justo aquí es donde queda evidenciada la utilidad del algoritmo PSO, el cual en resumen, proporciona los tiempos de encendido y apagado de los ventiladores, de modo que los picos de temperatura no deseada desaparezcan por completo o casi por completo, minimizando a su vez el tiempo en el que los ventiladores permanecen encendidos.

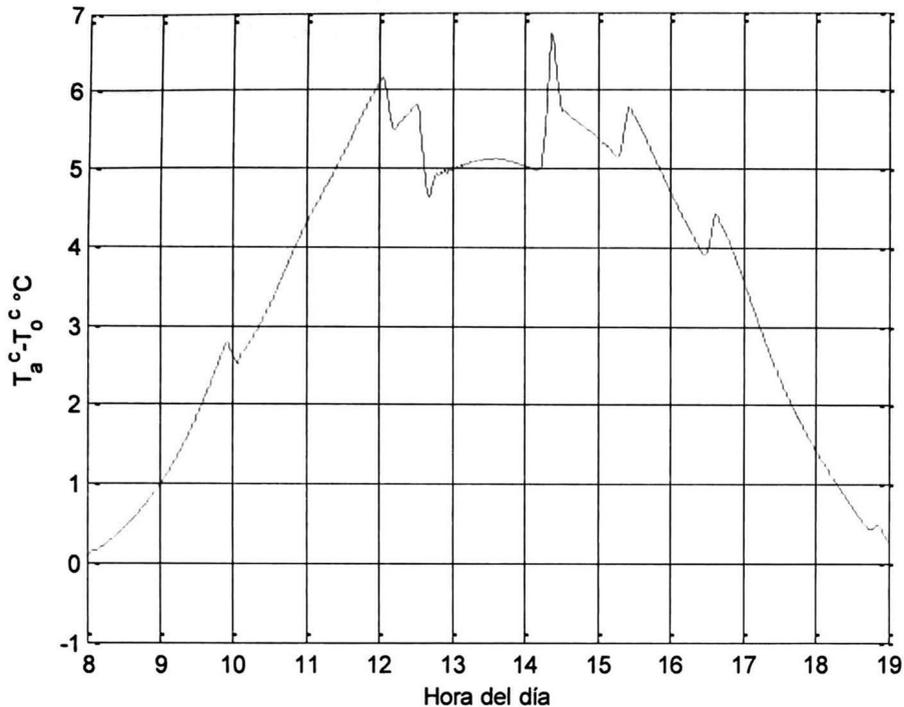


Figura 5.3 Diferencia entre la temperatura interna y externa del invernadero con los ventiladores encendidos.

5.2 Función objetivo

De acuerdo con la Sección 2.5, el algoritmo PSO utiliza una función objetivo como criterio mediante el cual se actualizan las posiciones y velocidades de las partículas del enjambre y se aproxima al punto óptimo. Esta función objetivo es propuesta en el contexto del problema a tratar. Para nuestro caso deseamos minimizar el sobrepaso por encima de 6°C de la diferencia entre las temperaturas, así como el tiempo en que permanecen encendidos los ventiladores. Para entender la idea de cómo se construye la función objetivo supongamos que todas las funciones involucradas son continuas. Así pues, la función objetivo se construye de la siguiente manera:

Sean

$$G(t, t_1, t_2) = \begin{cases} 0 & \text{si } \hat{T}_a^c(t, t_1, t_2) - \hat{T}_o^c(t) \leq 6 \\ \hat{T}_a^c(t, t_1, t_2) - \hat{T}_o^c(t) - 6 & \text{si } \hat{T}_a^c(t, t_1, t_2) - \hat{T}_o^c(t) > 6 \end{cases}$$

donde t_1 es el instante en el que se encienden los ventiladores, y t_2 el instante en el que se apagan. En el caso en que permanezcan apagados $t_1 = t_2 = 0$.

Sean t_1^0 y t_2^0 los tiempos donde el pico de temperatura comienza y termina, respectivamente. Este "pico" inicial, se calcula considerando los ventiladores apagados.

Sea además

$$H(t_1, t_2) = \int_t^{t_2^0 + t^*} G(t, t_1, t_2) dt$$

donde $t < t_1^0$, t^* es un tiempo fijo. De tal manera que:

$$\{t \ni G(t, t_1, t_2) > 6\} \subset [t, t_2^0 + t^*]$$

Es decir, se calcula la integral sobre un intervalo que contenga el pico de temperatura.

Sea $s = (t_1, t_2)$. De aquí que nuestra función objetivo es:

$$F(s) = P_1 H(s) + P_2 \|s\|$$

donde P_1 y P_2 son parámetros de peso, cuyo objetivo es dar prioridad a uno o a otro termino en la optimización. P_1 minimiza el exceso de temperatura, mientras que P_2 minimiza el tiempo en que los ventiladores permanecen encendidos.

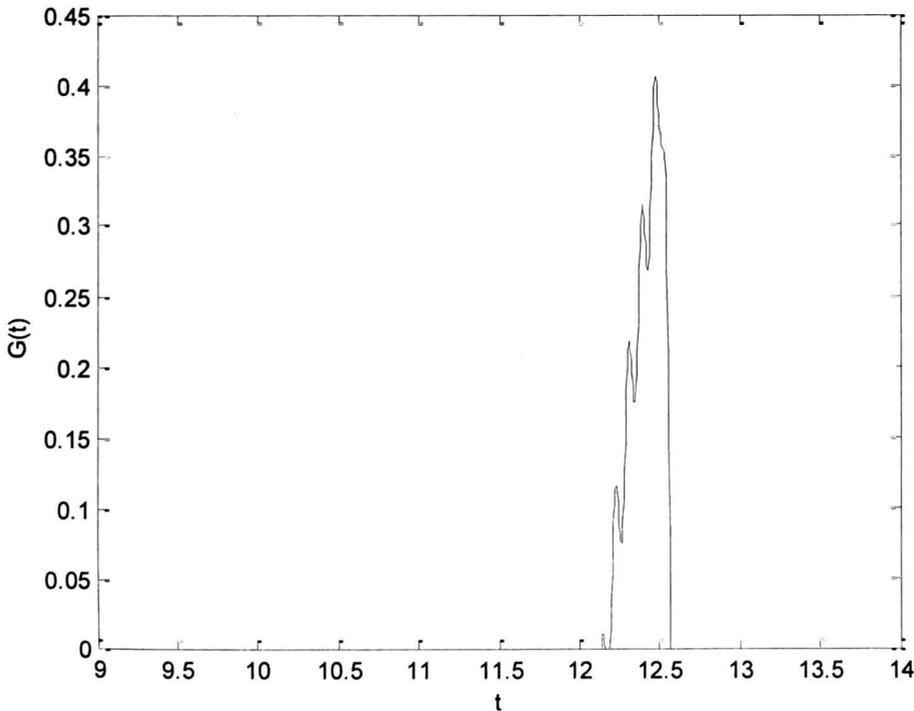


Figura 5.4 Ejemplo de G(t)

La Figura 5.4 muestra una aproximación de $G(t)$ para el primer pico, obtenida a partir de las condiciones de temperatura que muestra la Figura 5.2. El algoritmo PSO decide el control para eliminar los picos de temperatura, pero uno a la vez, como se explica detalladamente más adelante.

Ahora, debido a que todas las funciones involucradas son discretas, es necesario desarrollar el algoritmo considerando este hecho. Por ende redefinimos la función objetivo:

Sean

$$G_k(k, k_1, k_2) = \begin{cases} 0 & \text{si } \hat{T}_a^c(k, k_1, k_2) - \hat{T}_o^c(k) \leq 6 \\ \hat{T}_a^c(k, k_1, k_2) - \hat{T}_o^c(k) - 6 & \text{si } \hat{T}_a^c(k, k_1, k_2) - \hat{T}_o^c(k) > 6 \end{cases}$$

donde $t_1 = Tk_1$, $t_2 = Tk_2$ y T es el tamaño de paso de la predicción.

Sean $t_1^0 = Tk_1^0$, $t_2^0 = Tk_2^0$, $t = Tk$, $t^* = Tk^*$ con $k_2^0 < k^*$, igual que antes

$$H_k(s) = \sum_{i=k}^{k^*} TG_k(k).$$

Como $s = (t_1, t_2) = (Tk_1, Tk_2)$, sin pérdida de generalidad, suponemos que $s = (k_1, k_2)$, donde $k_1, k_2 \in \mathbb{Z}$.

De aquí que la función objetivo es:

$$F_k(s) = P_1 H_k(s) + P_2 \|s\|.$$

Debido a que $s_j \in \mathbb{Z}^2$ es necesario realizar una modificación a la Ecuación (2.10) del Capítulo 2, la cual proporciona la actualización de la posición de las partículas. Dicha modificación es como sigue:

$$s_{jl}(r+1) = \text{round}[s_{jl}(r) + v_{jl}(r+1)], \quad j = 1, 2, \dots, N \quad l = 1, 2, \dots, n$$

Esta modificación nos permite adecuar la búsqueda sobre un subconjunto $A \subset \mathbb{Z}^2$.

5.3 Descripción de la técnica de control PSO-Neuronal

La técnica de control PSO-Neuronal se describe a continuación. Supongamos que conocemos a priori las condiciones de temperatura externa, radiación, y tasa de ventilación a través de las ventanas, y deseamos conocer los momentos en los cuales

debemos prender y apagar los ventiladores de modo que la temperatura interna sea regulada. De aquí que:

- 1) Realizamos al tiempo $t = Tk$ una predicción inicial en el intervalo $[t, t + \#horas]$, para identificar el pico de temperatura más próximo. Esta predicción inicial se realiza considerando los ventiladores apagados.
- 2) Calculamos t_1^0 y t_2^0 , los tiempos donde el pico más próximo de temperatura comienza y termina, respectivamente.
- 3) Inicializamos la posición de las partículas del enjambre alrededor del punto (t_1^0, t_2^0) , esto evidentemente con el fin de proporcionarle información a priori al enjambre.
- 4) Por medio del algoritmo PSO encontramos la aproximación del punto óptimo de operación (T_1, T_2) .
- 5) Ejecutamos el control.

5.4 Simulación del control PSO-Neuronal

Supongamos que tenemos las mismas condiciones de temperatura, radiación, etc., con las cuales se realizó la predicción que se muestra en la Figura 5.1. La función objetivo del algoritmo PSO se propone como:

$$F_k(s) = 20H_k(s) + .01 \| s \|^2$$

La función recursiva que actualiza la velocidad de las partículas del algoritmo PSO se propone como:

$$v_{jl}(r+1) = v_{jl}(r) + R_1 (p_{jl}(r) - s_{jl}(r)) + .1R_2 (p_j^*(r) - s_{jl}(r)),$$

y el número de partículas del enjambre $N = 30$ y 40 épocas como criterio de paro. De acuerdo con el algoritmo descrito en la sección anterior procedemos de la siguiente manera:

- 1) Realizamos una predicción inicial al tiempo $t = 9$, en el intervalo $[9,14]$, con un periodo de muestreo $T = .025$. Calculamos $G_k(k, k_1, k_2)$, e identificamos el pico de temperatura más próximo como se muestra en la Figura 5.4.

2) Después calculamos t_1^0 y t_2^0 , cuyos valores son, 21.4, 24.45, respectivamente.

3) Inicializamos la posición de las partículas del enjambre alrededor del punto (857,979) que equivale en tiempo al punto (12.012,12.584).

4) El algoritmo PSO calcula una aproximación al óptimo en el punto $p^* = (839,945)$ que equivale en tiempo al punto (11.928,12.425), con un valor en la función objetivo de:

$$F_k(p^*) = 20H_k(p^*) + .01 \| p^* \| = 0 + 1.0633.$$

5) Llegado el momento encendemos y apagamos los ventiladores de acuerdo a p^* , como se muestra en la Figura 5.5.

Es importante señalar que $H_k(p^*) = 0$, esto implica que el pico de temperatura fue completamente eliminado como se muestra en la Figura 5.5.

Para demostrar que p^* es una buena aproximación, el cual elimina el pico de temperatura completamente, y optimiza el tiempo que permanecen encendidos los ventiladores, basta con observar que:

$$H_k(p^*) = H_k(839,945) = 0$$

$$H_k(840,945) = .0045$$

$$H_k(839,944) = .004.$$

De aquí p^* proporciona el menor tiempo que permanecen encendidos los ventiladores de tal modo que la diferencia entre las temperaturas se encuentre por debajo de 6°C, para el intervalo de tiempo [9,14].

Repetimos el algoritmo para el segundo pico de temperatura que se muestra en la Figura 5.2.

El algoritmo PSO calcula la aproximación al óptimo en el punto $p^* = (1326,1504)$, que equivale en tiempo al punto (14.21,15.045), con un valor en la función objetivo de:

$$F_k(p^*) = 20H_k(p^*) + .01 \| p^* \| = 0.024 + 1.78 = 1.804.$$

Aunque $H_k(p^*) = 0.024$, el control elimina de forma satisfactoria el pico de temperatura, como muestra la Figura 5.6.

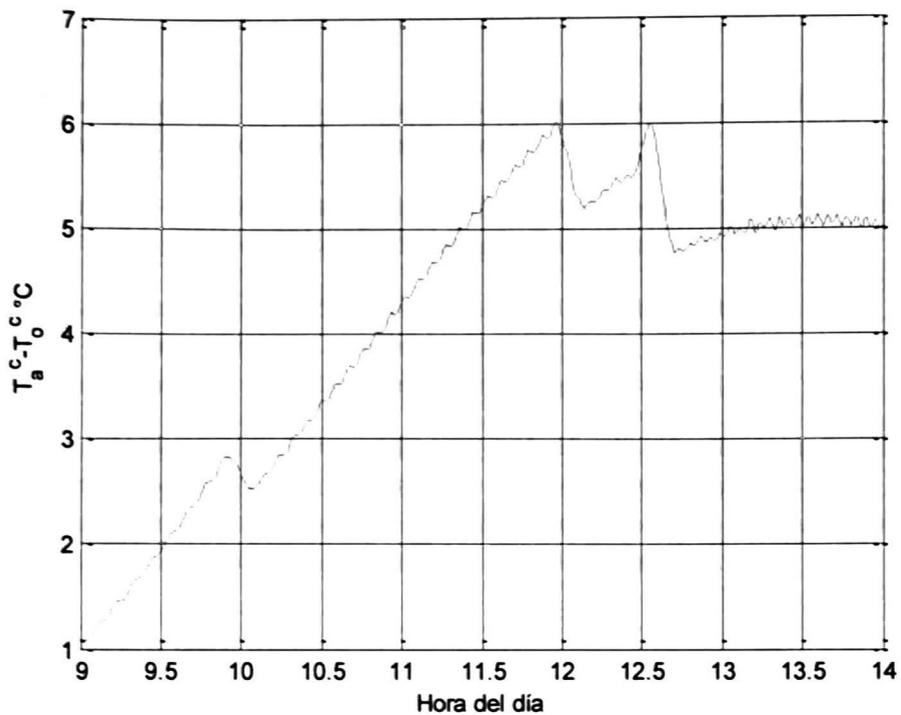


Figura 5.5 Temperatura regulada con el control PSO-Neuronal

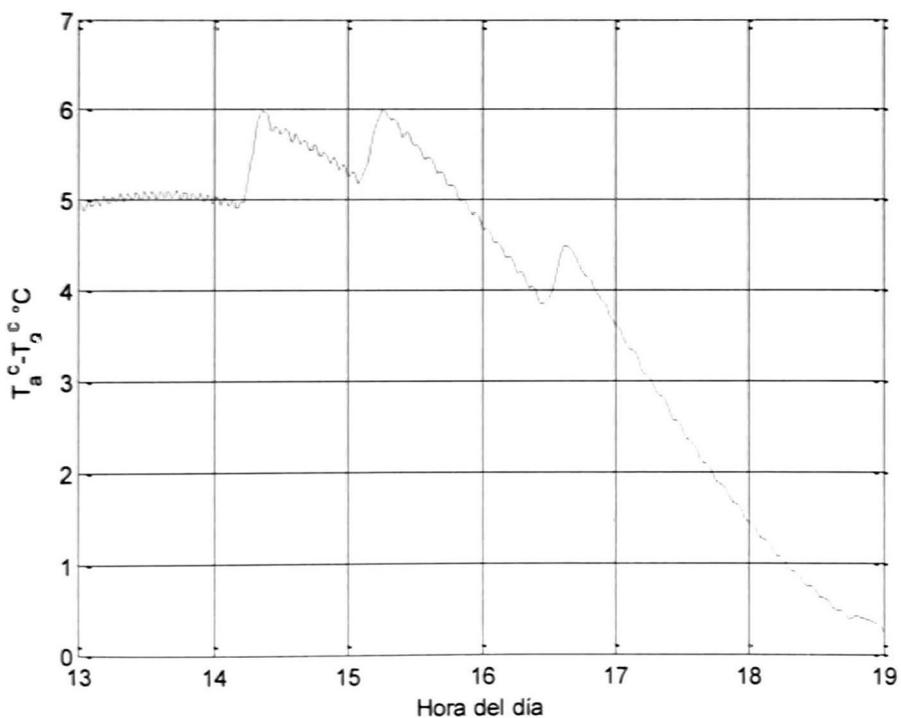


Figura 5.6 Temperatura regulada con el control PSO-Neuronal

El desempeño de la técnica es bastante satisfactorio, no solo por la eficacia del control sino además por la relativa sencilla implementación.

La técnica de control PSO-Neuronal puede ser generalizada a sistemas que compartan las características esenciales de este sistema. Dichas características son:

- 1) Que el modelo matemático del sistema sea muy difícil de obtener.
- 2) La dinámica del sistema en cuestión, sea lenta.
- 3) Las entradas de control del sistema sean escalones o rampas.

Las limitaciones de esta técnica deben ser evaluadas de acuerdo al sistema que se quiera controlar. Es necesario señalar la importancia que tiene el hecho de que la dinámica del sistema sea lenta, sin esta característica el algoritmo heurístico de optimización PSO no tendría el tiempo suficiente para calcular una aproximación al punto óptimo de operación.

Conclusiones y trabajo futuro

Esta técnica puede ser utilizada en sistemas que carezcan de modelos matemáticos precisos, cuya dinámica sea lenta y las entradas de control sean on-off. Dicha técnica de control, es relativamente fácil de implementar en cualquier invernadero con un mínimo de tecnología. La implementación de este control se hará lo antes posible en el prototipo que actualmente se construye en nuestras instalaciones. Ahí se explorarán las limitaciones y alcances de esta técnica de control. Una a una se incluirán los diferentes aspectos que involucran el clima de un invernadero, y se desarrollará un control que integre a las variables involucradas. Se explorarán nuevas estructuras de redes neuronales y se combinarán otros métodos heurísticos de optimización, para generar nuevos controles.

Bibliografía

- Alanis, A.Y.: Neural network training using Kalman Filtering, Tesis de Maestría, Cinvestav, Unidad Guadalajara, Guadalajara Jalisco México, 2004.
- Coelho, J.P., de Moura Oliveira, P.B. y Boaventura Cumba, J.: "Greenhouse air temperatura predictive control using the particle swarm optimization algorithm", Elsevier, 2005.
- Eredics, P. y Dobrowiecki, T.P.: "Neural Models for an Intelligent Greenhouse-The Heating", IEEE, 2010.
- Feldkamp, L. A., Prokhorov, D. V. y Feldkamp, T. M.: "Simple and conditioned adaptive behavior from Kalman filter trained recurrent networks", Neural Networks, vol. 16, pp. 683-689, 2003.
- Haykin, S.: *Neural Networks: A Comprehensive Foundation*, Macmillan College Publishing Company, New York, USA, 1994.

Haykin, S.: *Kalman Filtering And Neural Networks*, John Wiley & Sons, New York, USA, 2001.

Herrero, J.M., Blasco, X., Martinez, M., Ramos, C. y Sanchis, J.: "Robust identification of non-linear greenhouse model using evolutionary algorithms", Elsevier, 2007.

Kennedy, J. and Eberhart, R. : "Particle Swarm Optimization", *Proceedings of the IEEE International Conference on Neural Networks*, vol. 4, Washington, DC, USA, November 1995, pp1942-1948.

Pham, D.T. y Liu, X.: *Neural Networks for Identification, Prediction and Control*, Springer-Verlag, London, Great Britain, 1995.

Salazar, R., Lopez, I. y Rojano, A.: "A Neural Model to Control Greenhouse Environment", *Sixth Mexican International Conference on Artificial Intelligence*, 2008.

Salazar, R., Schmidt, U., Huber, C., Rojano, A. y Lopez, I.: "Neural Networks Models for Temperatures and CO₂Control", *International Journal of Agricultural Research*, 2010.

- Sánchez, E.N. y Alanís, A.Y. y Rico, J.: "Electric Load Demand Prediction using Neural Networks Trained by Kalman Filtering", *Proceedings of the IEEE International Conference on Neural Networks*, 2004, Budapest, Hungría, 2004.
- Sánchez, E.N. y Alanís, A.Y.: *Redes Neuronales*, Pearson-Prentice Hall, Madrid España, 2006.

Straten, G.V., Willigenburg, G.V., Henten, E.V. and Ooteghem, R.V.: *Optimal Control of Greenhouse Cultivation*, Taylor and Francis Group, New York, Usa, 2011.

- Zhao, B. y Wang, K.: "Study on Greenhouse Environment Neural Network Model Based on PSO Algorithm", *International Conference on Intelligent Computing and Cognitive Informatics*, 2010.

Zhao, B. y Wang, K.: "Study on Neural Network Modeling of Greenhouse Environment Based on Partial Least Square Algorithm", *International Conference on Computer and Communication Technologies in Agriculture Engineering*, 2010.



CENTRO DE INVESTIGACIÓN Y DE ESTUDIOS AVANZADOS DEL I.P.N. UNIDAD GUADALAJARA

El Jurado designado por la Unidad Guadalajara del Centro de Investigación y de Estudios Avanzados del Instituto Politécnico Nacional aprobó la tesis

Control PSO-Neuronal para regular la temperatura de un invernadero

del (la) C.

Raúl AVILA MIRANDA

el día 10 de Agosto de 2012.

Dr. José Luis Alejandro Naredo
Villagrán
Investigador CINVESTAV 3C
CINVESTAV Unidad Guadalajara

Dra. Ofelia Begovich Mendoza
Investigador CINVESTAV 3B
CINVESTAV Unidad Guadalajara

Dr. José Javier Ruíz León
Investigador CINVESTAV 3B
CINVESTAV Unidad Guadalajara

Dr. Antonio Ramírez Treviño
Investigador CINVESTAV 3A
CINVESTAV Unidad Guadalajara

Dr. Mario Angel Siller González
Pico
Investigador CINVESTAV 2C
CINVESTAV Unidad Guadalajara



CINVESTAV - IPN
Biblioteca Central



SSIT0011086