Ingeniería. Revista de la Universidad de Costa Rica
ISSN: 1409-2441
marcela.quiros@ucr.ac.cr
Universidad de Costa Rica
Costa Rica

Gólcher Barguil, Luis Alejandro
CONTROL ADAPTIVO UTILIZANDO PROGRAMACIÓN DINÁMICA HEURÍSTICA
Ingeniería. Revista de la Universidad de Costa Rica, vol. 17, núm. 2, agosto-diciembre, 2007, pp. 11-26
Universidad de Costa Rica
Ciudad Universitaria Rodrigo Facio, Costa Rica

Ingeniería 17 (2): 00-00, ISSN: 1409-2441; 2007. San José, Costa Rica

CONTROL ADAPTIVO UTILIZANDO PROGRAMACIÓN DINÁMICA HEURÍSTICA

Luis Alejandro Gólcher Barguil

Resumen

Se compara el desempeño de dos métodos diferentes para controlar los estados de un sistema simulado de un tanque, utilizando los conceptos de Programación Dinámica Heurística. El desempeño es medido en términos de su capacidad de aprendizaje, tiempo de entrenamiento y manejo del ruido. El objetivo de los algoritmos es hacer que la temperatura del tanque siga una referencia dada. Para esta tarea, el Enfoque Estocástico aprende a controlar el sistema más ágilmente; sin embargo, el Enfoque Determinístico maneja mejor el ruido en la salida del sistema. Más aún, si la señal de referencia está constantemente variando, el Enfoque Determinístico controla mejor el sistema.

Palabras clave: control, adaptivo, neuronal.

Abstract

The performance of two methods is compare on controlling the states of a simulated tank system using Heuristic Dynamic Programming concepts. The performance is measure in terms of learnability, training time and noise handling. The goal of the algorithms is to make the tank's temperature track a given reference. For this task, the Stochastic Aproach method learned to control the system faster; however, the Deterministic Approach handled the system's output noise better. Nevertheless, if the reference signal is constantly changing, the Deterministic Approach would prove to control better the system.

Key words: control, adaptive, neural.

Recibido: xxxx • **Aprobado:** xxxxx

En las décadas pasadas, se ha desarrollado una gran variedad de métodos para controlar los procesos industriales [1]. Entre estos se encuentran los que se basan en la teoría de Programación Dinámica [5]. Este trabajo compara dos enfoques diferentes para implementar un algoritmo de programación dinámica en el control de la temperatura de un tanque. El estudio está motivado por la falta de comparaciones entre estos distintos métodos.

Figura 1. Diagrama de bloques del sistema.
Fuente: (El autor)

1. INTRODUCCIÓN

Suponga que se tiene un sistema de una entrada y una salida, como el que se muestra en la Figura 1.

El sistema transforma la señal de entrada, u, en una señal, y, tal que

$$y = f(u,x) \tag{1}$$

en donde f es una función lineal o no-lineal y x son los estados del sistema. Ahora, presuma que la señal de salida, y, del sistema necesita ser igual a un valor constante, r. Esto conlleva al problema de determinar cuál entrada, u, produce la salida deseada, r, llamada la referencia.

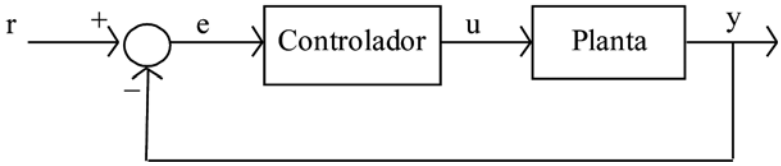

Figura 2. Sistema de control con retroalimentación.
Fuente: (El autor)

La solución que la teoría de control clásica señala, está basada en el concepto del lazo de retroalimentación, como se presenta en la Figura 2.

Un controlador, C, es construido con la capacidad de alterar, u, la señal de entrada al sistema, tal que el error, $e = r - y$, se mantenga cerca de cero. La desventaja de este método es que, para diseñar el controlador, se requiere la función de la planta f en la ecuación (1).

Una red neuronal hace uso de aprendizaje no lineal, procesamiento en paralelo y de generalización, haciendo esta técnica la opción inteligente cuando la función de la planta no se conoce. Muchos intentos se han realizado para aplicar redes neuronales multicapa al campo del control automático. Estos controladores pueden ser clasificados en tres grandes áreas:

- **Control Supervisado:** Una red neuronal aprende el mapeo desde las entradas, y traslada las entradas hasta las acciones deseadas, por medio del entrenamiento de lo que debe hacer la red neuronal [1].

- **Control Inverso:** Una red neuronal aprende el comportamiento inverso del sistema [2]. Este tipo de control puede presentar serios problemas si la función f no tiene inversa.

- **Control Adaptivo Neuronal:** Una red neuronal es utilizada para identificar el comportamiento del sistema y predecir futuras salidas del proceso utilizando una red neuronal como emulador [3], [4]. El controlador está basado en la minimización del error predicho.

Las ventajas de estas estrategias residen en que no requieren conocer la función de la planta explícitamente, como se describe en la ecuación (1).

La idea detrás de estos algoritmos está basada en el hecho de que existe un sistema con ciertos estados iniciales. Para llevar la salida del sistema a una cierta referencia, los estados del sistema deben ser trasladados a otros valores. La Figura 6 muestra esta situación para un sistema de dos estados.

Aplicando al sistema una señal de entrada determinada, $u(t)$ - una tarea específica del controlador-, los estados $X = (X_0, X_1)$ se trasladan a los valores óptimos, pero únicamente si el sistema es controlable. Estos valores son óptimos porque en ese punto en el espacio, producen la salida deseada. Sin embargo, hay cientos de caminos que el sistema puede tomar. Por lo tanto, el diseñador debe seleccionar uno de estos caminos tal que se minimice un determinado indicador.

Utilizando la teoría de programación dinámica (PD), muchas técnicas distintas se han desarrollado para resolver este problema [5]. El punto radica en mover los estados del sistema para que produzcan la salida deseada. Sin embargo, la entrada de control que se aplica al sistema se selecciona de tal forma que minimice un criterio de desempeño, $J(x, u)$, que el diseñador preliminarmente ha establecido. PD alcanza el menor índice de desempeño tomando un paso a la vez en el camino óptimo desde (t) hasta $(t+1)$, y adicionando el mínimo costo de estar en $(t+1)$ hasta el estado final (t_f). La Figura 7 describe la situación.

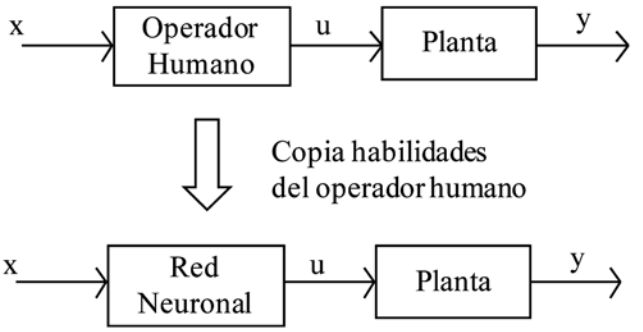

Figura 3. Control Supervisado
Fuente: (El autor)

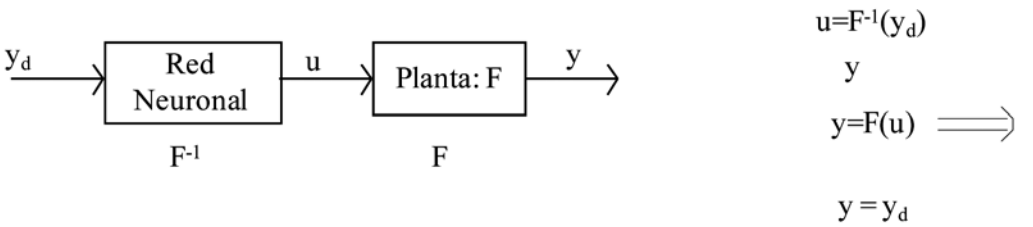

Figura 4. Control Inverso.
Fuente: (El autor)

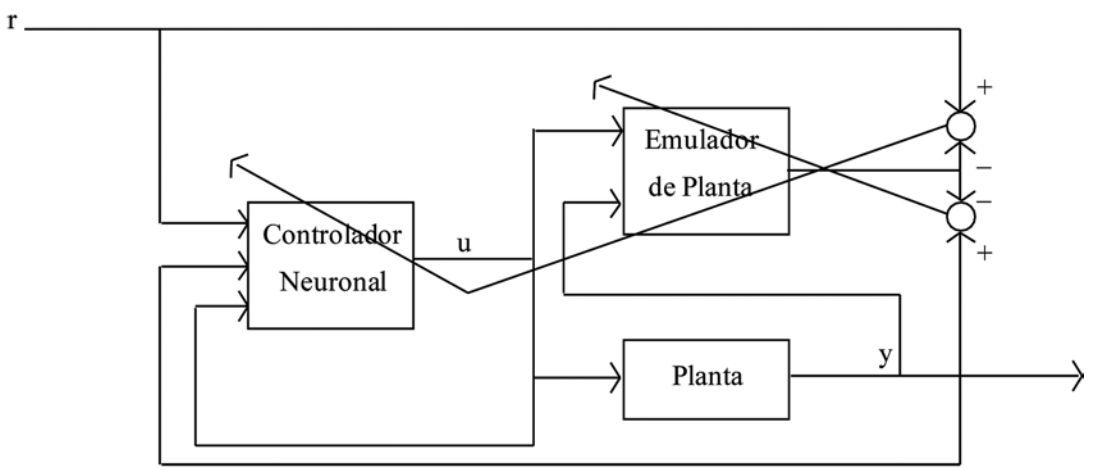

Figura 5. Control Adaptivo.
Fuente: (El autor)

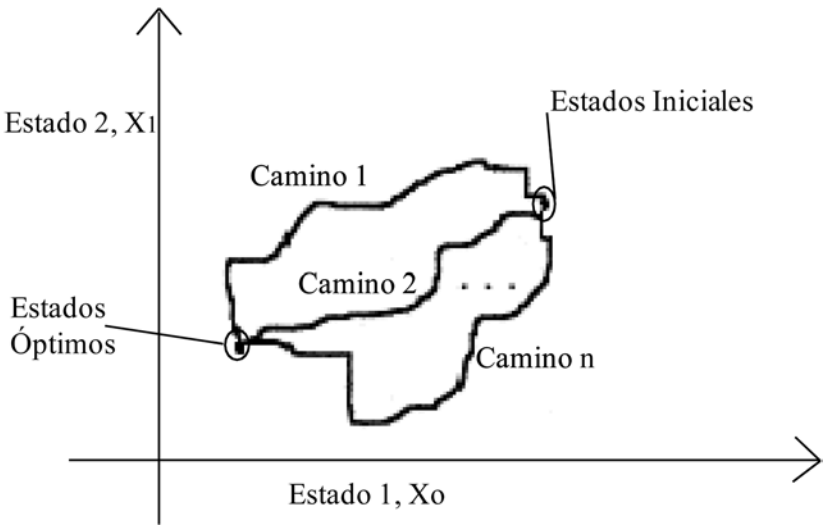

Figura 6. Espacio de estados de un sistema de segundo orden.
Fuente: (El autor)

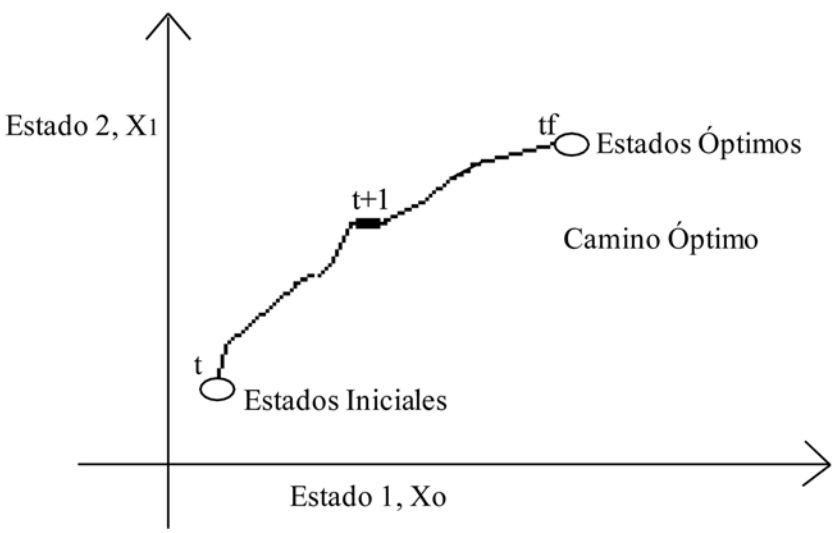

Figura 7. Enfoque de la Programación Dinámica.
Fuente: (El autor)

Si se representa el mínimo índice de desempeño o el mínimo costo con J^*_{t,t_f}, que se puede alcanzar desde el paso (t) hasta el paso (t_f), entonces PD establece que

$$J^*_{t,t_f} = J_{t,t+1} + J^*_{t+1,t_f} \tag{2}$$

en donde $J_{t,t+1}$ es el costo de tomar un paso en el camino óptimo y $J^* t_{t+1}, t_f$ es el mínimo costo alcanzable desde el paso $t+1$hasta t_f.

Ahora el problema de seleccionar el camino óptimo se ha descompuesto en pasos simples. Básicamente, un indicador de desempeño es minimizado con la restricción correspondiente al modelo de la planta.

Un índice de desempeño típico $J(x,u)$ es

$$J(x,u) = \sum_{k=t}^{t_f} [(y-r)^2 + \rho u^2] \tag{3}$$

en donde y es la salida del sistema, r es la señal de referencia, ρ es el factor costo y u es la entrada al sistema.

Si ρ es cero, la forma de minimizar $J(x,u)$ es seleccionando una entrada u tal que $y=r$. Si ρ es mayor que cero, entonces un factor de energía de entrada es incorporado al índice de desempeño. Esto implica que un valor mínimo de entrada, u, se debe seleccionar tal que $y=r$.

Es necesario predecir el término del lado izquierdo de la ecuación (2) y el segundo término del lado derecho de la misma ecuación. Después de haber predicho estos términos, la mejor acción, u, que minimiza estos dos términos, debe ser encontrada. Una forma de determinar este valor es predecir J^*_{t+1,t_f}para todos los posibles valores de u en el paso $t+1$; y luego evaluar cuál acción causa el menor J^*_{t+1,t_f}. El problema con este enfoque es que requiere de una gran cantidad de cálculos computacionales. En la próxima sección, otro enfoque es presentado.

2. PROGRAMACIÓN DINÁMICA HEURÍSTICA

Un enfoque de PD es llamado Programación Dinámica Heurística (PDH) [5,6]; éste comparte la idea de una tercera red neuronal del Control Adaptivo Neuronal, como se describió anteriormente. Una entrada de excitación, $u(t)$, es seleccionada tal que minimice los valores futuros de una función de costo J^*_{t+1,t_f}. La minimización se obtiene calculando una función $\hat{J}_{t,t_f}$ para predecir la suma descontada del costo futuro.

Para implementar PDH se deben construir dos redes neuronales: una que predice J^*_{t,t_f}, llamada la Red Crítica (del verbo criticar), y otra para generar la acción de control $u(t)$, denominada la Red de Acción. La Figura 8 muestra sus conexiones.

Aunque existen pequeñas diferencias, la mayoría de implementaciones PDH usualmente siguen los pasos básicos, que se describen a continuación.

Para un entrenamiento en línea, los pasos básicos de PDH, en el paso t, son:

1. Obtenga y almacene los estados del sistema $x(t)$.
2. Con la Red Crítica genere $\hat{J}_{t,t_f}$.
3. Con la Red de Acción calcule $u(t)=Ax(t)$.
4. Obtenga $x(t+1)$, ya sea esperando hasta $t+1$ o prediciendo $x(t+1)=f(x(t),u(t))$.
5. Calcule

$$J_{t,t_f\ deseado} = J_{t,t+1} + \lambda \hat{J}_{t+1,t_f} \tag{4}$$

donde el primer término en el lado derecho de la ecuación es el costo de tomar un paso en el camino óptimo; el último término en el lado derecho de la ecuación es el indicador de desempeño para $x(t+1)$, y la constante λ (entre 0 y 1) es un factor de

descuento para futuras predicciones. El factor de descuento controla la longitud del horizonte finito, sobre el cual ocurre la planificación.

6. Actualice la Red Crítica en *t* con

$$error_{crítico} = J_{t,t_f\ deseado} - \hat{J}_{t,t_f} \quad (5)$$

7. Actualice la Red de Acción en *t* .

8. Repita los pasos del 1 al 7 hasta que el error del estimado de la Red Crítica y de la Red de Acción se encuentre dentro de un rango pre-especificado.

El paso 7 se puede desarrollar en dos formas distintas, las cuales conllevan a diferentes arquitecturas para implementar PDH.

La primera se denomina Enfoque Determinístico y está basada en la arquitectura mostrada en la Figura 9 [7].

Se compone de dos redes neuronales. El papel de la Red Crítica es calcular $J_{t,t_f}(x)$, el indicador de desempeño, estimando $\hat{J}_{t,t_f}(x)$. La Red Crítica aprende por la Ecuación (5) con un indicador estimado $\hat{J}_{t+1,t_f}(x+1)$. Por simplicidad, se asume que J es explícitamente independiente de $u(t)$ o que ρ es cero. Si ρ es mayor que cero, entonces se requiere adicionar otra entrada a la Red Crítica, $u(t)$. El modelo de la planta es una red neuronal que ha aprendido la función f en ecuación (1). Esta red neuronal debe aprender el modelo antes de implementar el aprendizaje en línea PDH. Una vez que tiene la habilidad de generalizar correctamente el proceso real, puede seguir aprendiendo durante el entrenamiento en línea.

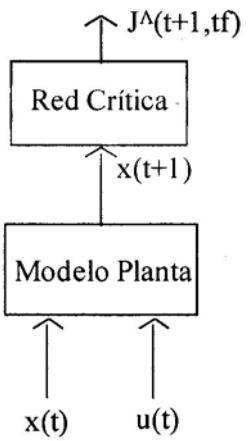

Figura 9. Enfoque Determinístico
Fuente: (El autor)

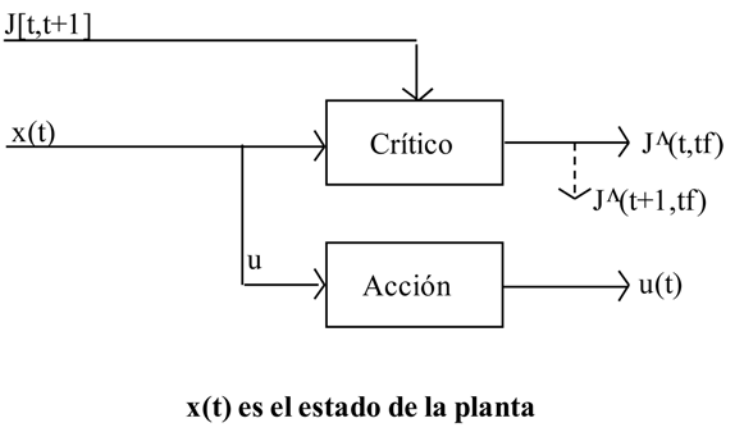

Figura 8. Arquitectura General de PDH
Fuente: (El autor)

Con un modelo de la planta, los estados futuros pueden ser estimados si una acción *u* es tomada. Esto permite calcular si el futuro indicador de desempeño es menor que el costo presente.

Básicamente *u* debe ser seleccionada tal que $\hat{J}_{t+1,t_f}(x+1)$ sea mínimo. Esto puede ser ejecutado calculando las derivadas parciales del indicador de desempeño futuro con respecto a *u*, tomándolos de la Red Crítica y el modelo de la planta:

$$\frac{d\hat{J}_{t+1,t_f}(x+1)}{du(t)}$$

Con esta ecuación, la acción *u* se selecciona tal que su derivada sea siempre negativa. Si *u* produce una derivada negativa, entonces esta acción está minimizando $\hat{J}_{t+1,t_f}$. Si *u* produce una derivada positiva, entonces la acción está maximizando $\hat{J}_{t+1,t_f}$ y la dirección de la acción debe ser cambiada. La Figura 10 muestra esta situación.

Esta idea puede ser implementada con un simple algoritmo, el cual perturba *u* para calcular la derivada. Otra opción es construir una red neuronal y enseñarle con:

$$error_{acción} = \mathrm{B}\frac{d\hat{J}_{t+1,t_f}}{du(t)} \tag{6}$$

donde B es un factor de escalamiento, el cual provee usualmente una convergencia más rápida y un aumento de estabilidad.

El segundo punto de vista es el Enfoque Estocástico, y está basado en la arquitectura de la Figura 11.

Se compone de dos redes neuronales: la Red Crítica y la Red SRV (por sus siglas en inglés, Stochastic Real Value Network). La Red Crítica tiene la misma función que su homólogo en el Enfoque Determinístico, y si le enseña con la

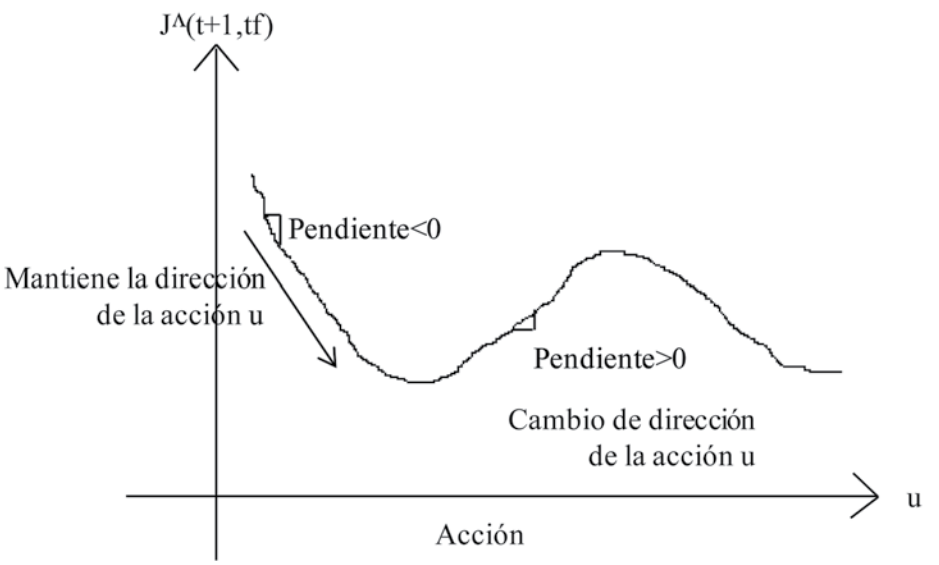

Figura 10. Seleccionando la acción, *u*.
Fuente: (El autor)

misma función de error en ecuación(5). La Red SRV [8] explora el espacio de entradas para maximizar una señal de refuerzo, *r*,

$$r = e^{-[J_{t,t+1}+\hat{J}_{t+1,tf}]} \quad (7)$$

De esta forma, para maximizar el refuerzo que la Red SRV recibe de una acción determinada, la salida *u* tiene que minimizar el costo presente. Dado que un modelo de la planta no se construye, el sistema tiene que esperar hasta el siguiente paso, $t+1$, para obtener $x(t+1)$ en el algoritmo PDH, (paso 4).

3. ESTUDIO DE SIMULACIÓN

Un sistema térmico es controlado utilizando el algoritmo PDH, tal que permita establecer una comparación entre los dos enfoques. La Figura 12 describe el proceso. Dado que es un sistema de primer orden, hay un único estado, el cual corresponde a la salida del sistema, $y(k)$.

El problema de control se establece como un problema de seguimiento, en el intento de alcanzar la temperatura del tanque a una determinada referencia. Por lo tanto, la función de costo en tomar un paso en el camino óptimo, es:

$$J_{t,t+1} = (y - r)^2 \quad (8)$$

Consecuentemente, la única forma de minimizar la ecuación(8) es haciendo $y=r$.

En ambos enfoques, la Red Crítica es la misma. Esta consiste de una unidad de entrada, una capa intermedia de tres unidades, y una unidad de salida. Las capas están completamente interconectadas. La función de activación de las unidades está determinada por:

$$f(x) = \frac{1}{1+e^{-x}} \quad (9)$$

La Red Crítica aprende utilizando el método Diferencias Temporales, TD(0) [9].

Enfoque Determinístico

La red neuronal que modela la planta consiste de dos unidades de entrada, una para el estado $x(k)$, y la otra para entrada de control, $u(t)$; cinco unidades en la capa intermedia, y una unidad en la salida $x(k+1)$. Las capas están completamente interconectadas. El conjunto de entrenamiento consiste de ochenta patrones de entrada, los

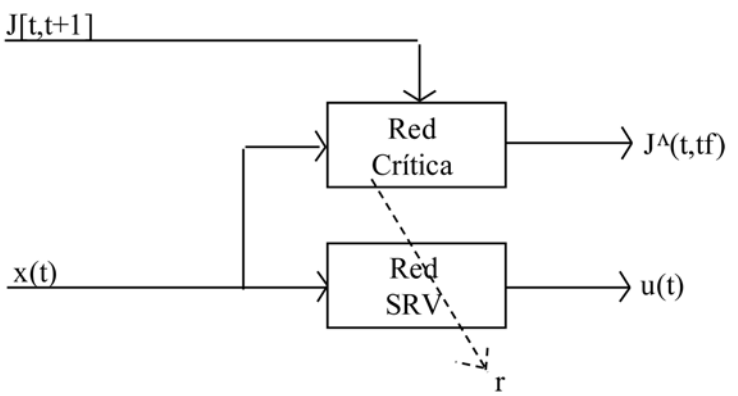

Figura 11. Enfoque Estocástico.
Fuente: (El autor)

cuales se obtienen haciendo un barrido a través de la entrada desde el estado de 1 °C a 80 °C, y seleccionando valores aleatorios para la entrada de control. Las salidas meta son las respuestas que el proceso exhibe para esos patrones. La red neuronal es enseñada utilizando el método de retropropagación por lote.

Este barrido por el espacio de estados es posible en el marco de la simulación, pero en el caso de un sistema físico, puede no ser posible, dado que, no se conoce cuál entrada de control genera un estado determinado. Por lo tanto, el barrido debe realizarse explorando el espacio de estados, tan sabiamente como sea posible con la entrada de control.

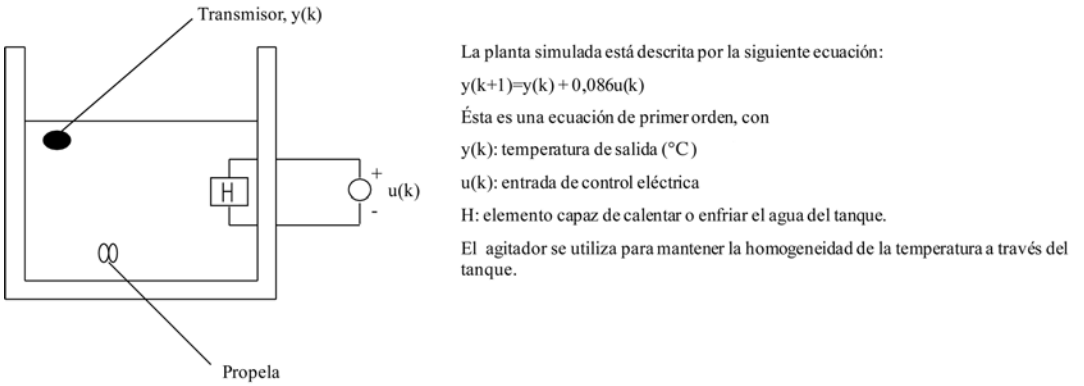

Figura 12. Sistema térmico simulado.
Fuente: (El autor)

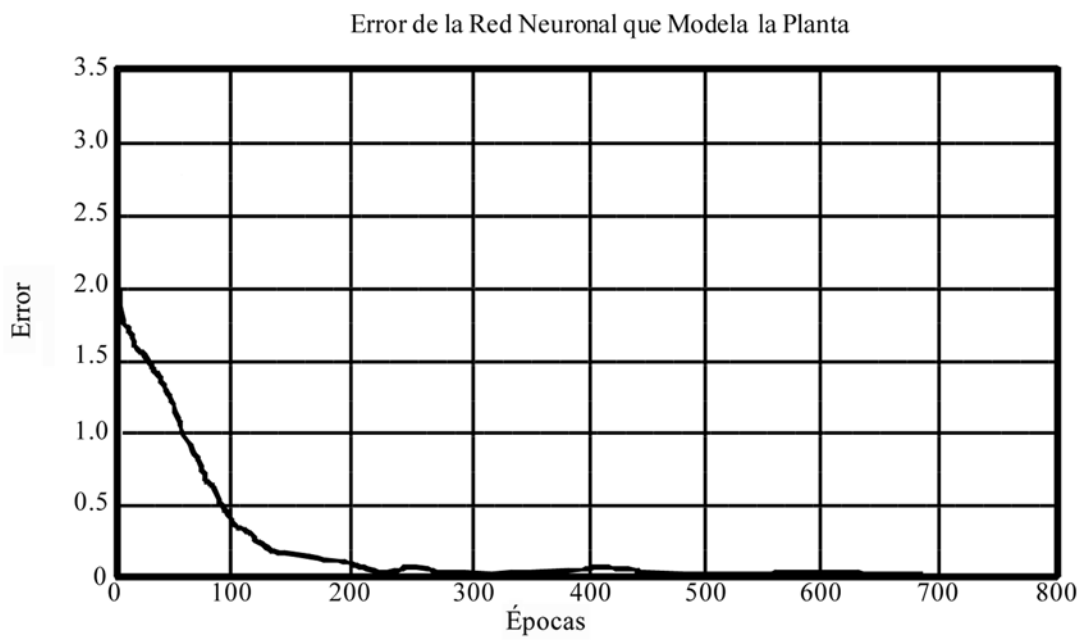

Figura 13. Enfoque Determinístico: Modelo de la Planta.
Fuente: (El autor)

En la Figura 13, el error es graficado contra las épocas de aprendizaje. Con setecientas épocas, el error es aproximadamente cero; esto significa que la red neuronal aprende muy bien los patrones de entrenamiento, pero no tiene una buena generalización en el resto del espacio de entradas. Muchos puntos de parada se probaron y en quinientas épocas se encuentra que la red generaliza bien. El aprendizaje del modelo de la planta es laborioso, aunque la planta es lineal, de una entrada y una salida. Es considerablemente difícil conocer el mejor punto para detener el entrenamiento, dado que el conjunto de patrones de validación es el resto del espacio positivo bidimensional del conjunto de entrada.

Una vez que la red neuronal generaliza bien el modelo de la planta, el sistema de control se simula con un estado inicial de 25 °C y una referencia a alcanzar de 40 °C. La Figura 14 muestra que la Red Crítica ha aprendido a predecir el futuro costo descontado. Asimismo, el estado, después de un hundimiento en las primeras épocas, comienza a seguir la referencia. En cuatrocientos cincuenta épocas, la salida está prácticamente cerca de los 40 °C. La pendiente de la curva decrece conforme el número de épocas aumenta, dado que la derivada del futuro costo descontado tiende a cero. En la Figura 15 se observa como la acción de control, *u*, está minimizando el costo futuro descontado, haciendo la pendiente siempre negativa.

Después de que la salida ha alcanzado 40 °C, la referencia se cambia a 60 °C. De nuevo, de la Figura 16, le toma al sistema alrededor de cuatrocientas cincuenta épocas para lograr la meta. Las épocas se pueden traducir en pasos en el tiempo, con el conocimiento de un periodo de muestreo:

$$\text{Paso en el tienpo} = (\text{época}) \bullet (\text{periodo de muestreo}) \quad (10)$$

Adicionalmente, el sistema se prueba con ruido en la salida del proceso térmico. El ruido adicional corresponde a una distribución normal con medio 2 y desviación estándar 0,5 °C (Figura 17). Le toma al sistema aproximadamente mil épocas en alcanzar la referencia, en donde tiene una mínima oscilación.

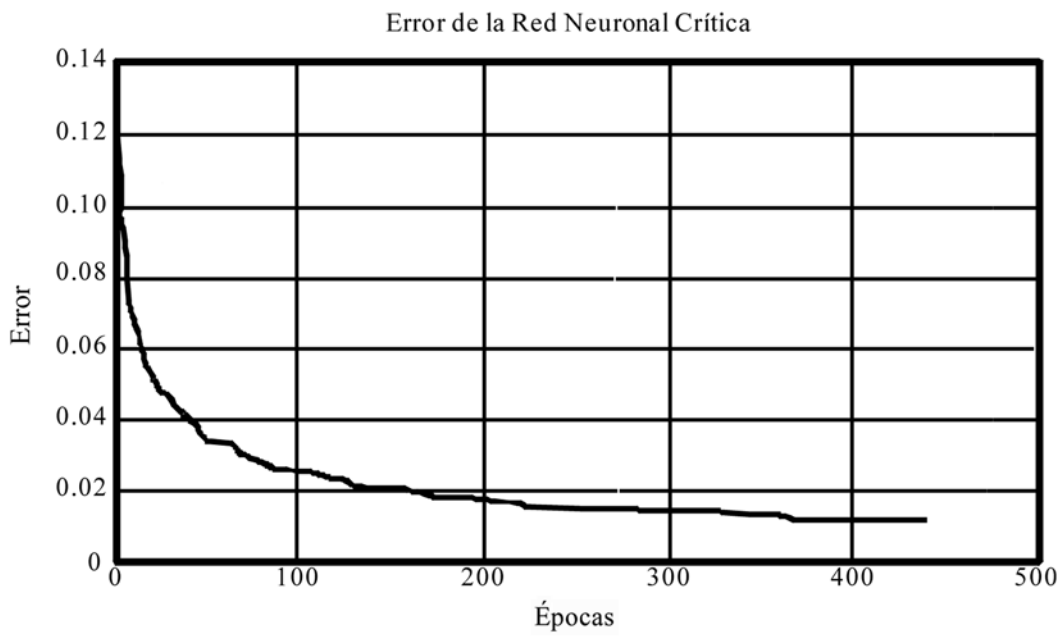

Figura 14. Enfoque Determinístico: Red Crítica.
Fuente: (El autor)

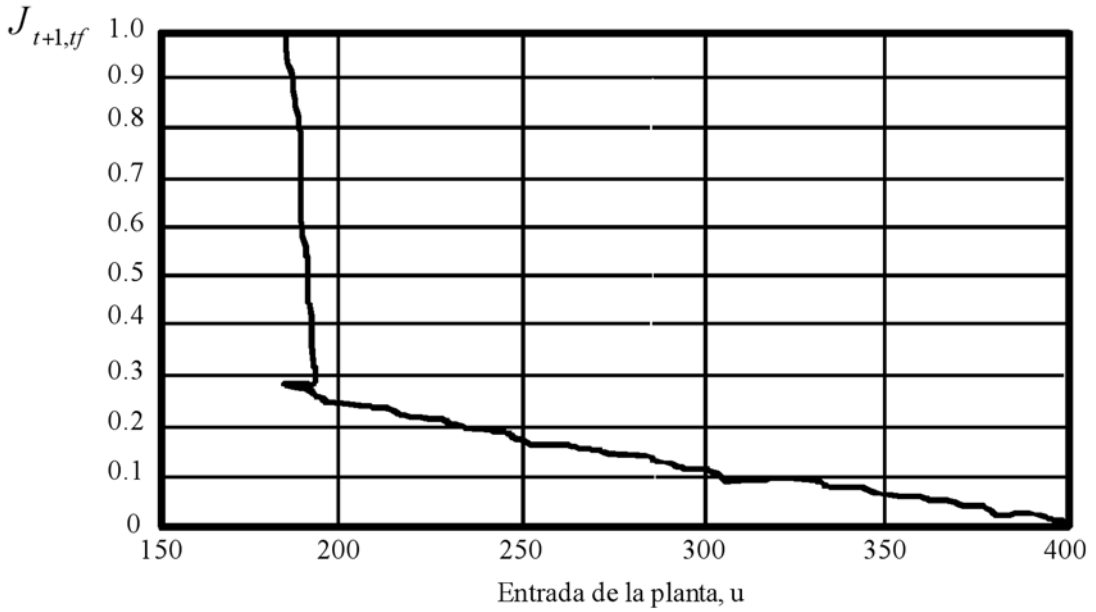

Figura 15. Enfoque Determinístico: Seguimiento de temperatura.
Fuente: (El autor)

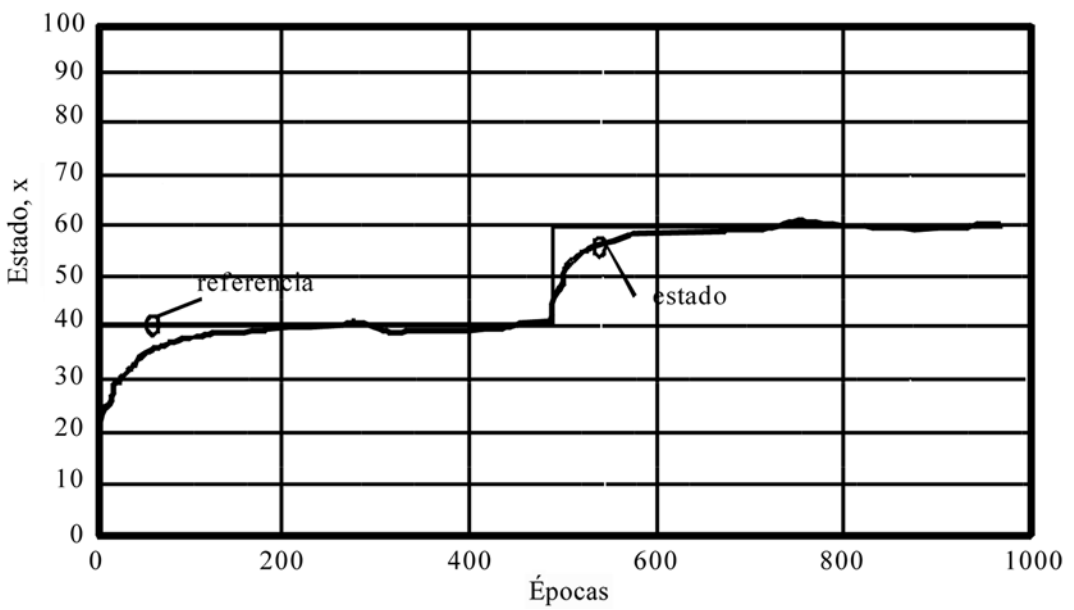

Figura 16. Enfoque Determinístico: Seguimiento de temperatura
Fuente: (El autor)

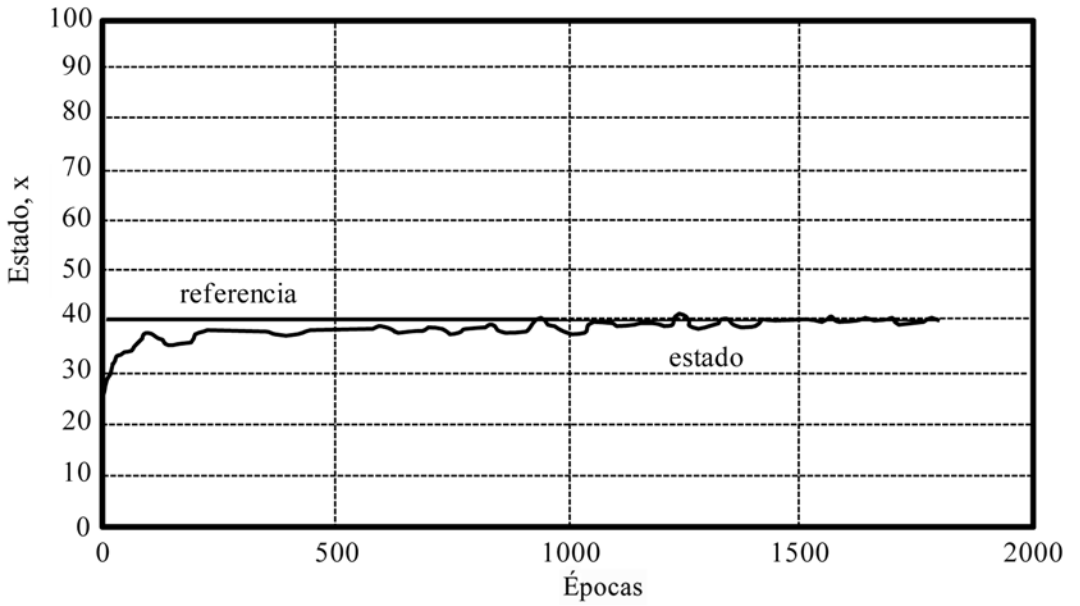

Figura 17. Enfoque Determinístico: Salida de temperatura con referencia incluyendo ruido
Fuente: (El autor)

Enfoque Estocástico

La red neuronal SRV consiste de una unidad, la cual tiene una entrada y una salida. Su entrada corresponde al estado actual del proceso, y su salida corresponde a la entrada de control del proceso, *u*.

El estado inicial se establece en 25 °C y la referencia en 40 °C. La Red Crítica aprende nuevamente a predecir el futuro costo descontado, como se muestra en la Figura 18.

La salida en las primeras épocas da un gran salto hasta 110 °C. Luego, el sistema comienza a seguir la referencia conforme la señal de refuerzo se aproxima a 1, su valor máximo, (Ver Figura 19). Entre quinientas y ochocientas épocas, la salida oscila alrededor de 40 °C con menores valores, conforme las épocas aumentan, (Figura 20). En novecientas épocas, las salida sólidamente alcanza la referencia.

En la Figura 21, la señal de refuerzo se grafica contra la entrada de la planta, con el fin de simbolizar el carácter exploratorio de la Red SRV. En las primeras épocas, la Red SRV encuentra que necesita aumentar la temperatura, entonces, la entrada de control aumenta su valor. Después, aprende que la temperatura es muy alta y la entrada de control disminuye. Conforme el estado se acerca a la referencia, la entrada de la planta tiende a cero.

Después de que la salida ha alcanzado 40°C, la referencia se cambia a 60 °C. De nuevo, se observa que se toma unas novecientas épocas para alcanzar la nueva referencia, (Figura 22).

Asimismo, el sistema se prueba con ruido a la salida del proceso, (Figura 23). Le toma al sistema aproximadamente cuatro mil épocas para mantener la salida alrededor de 40 °C.

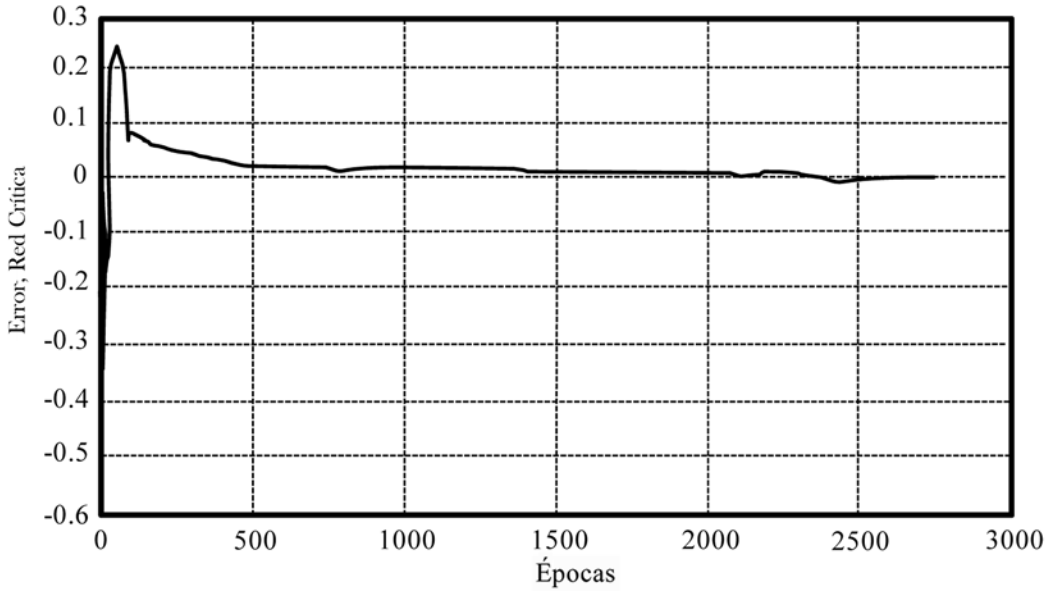

Figura 18. Enfoque Estocástico: Red Crítica.
Fuente: (El autor)

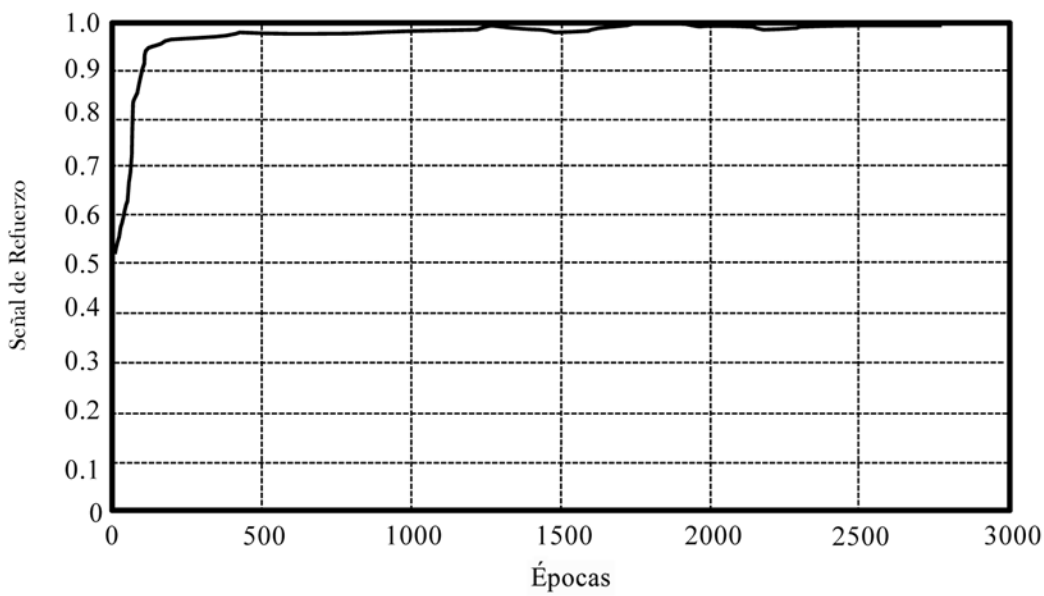

Figura 19. Enfoque Estocástico: Señal de Refuerzo.
Fuente: (El autor)

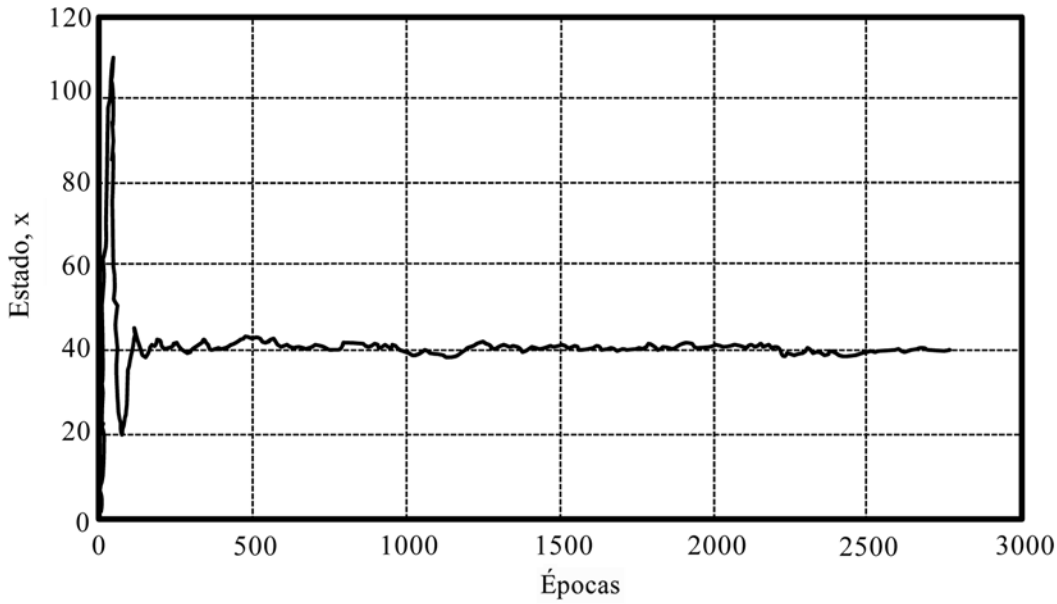

Figura 20. Enfoque Estocástico: Seguimiento de temperatura.
Fuente: (El autor)

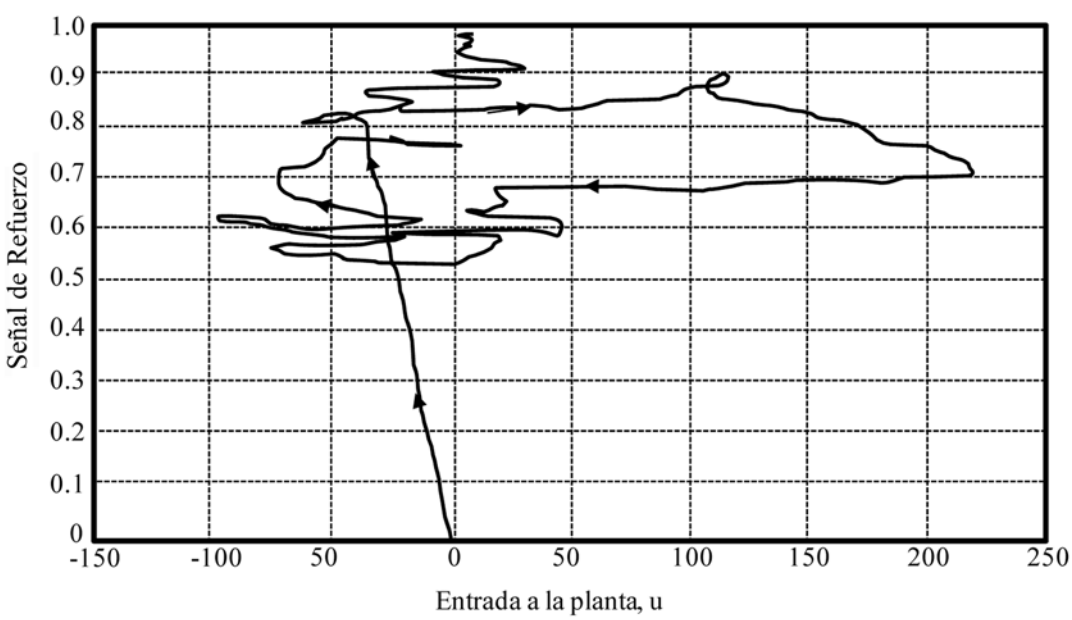

Figura 21. Enfoque Estocástico: Señal de refuerzo.
Fuente: (El autor)

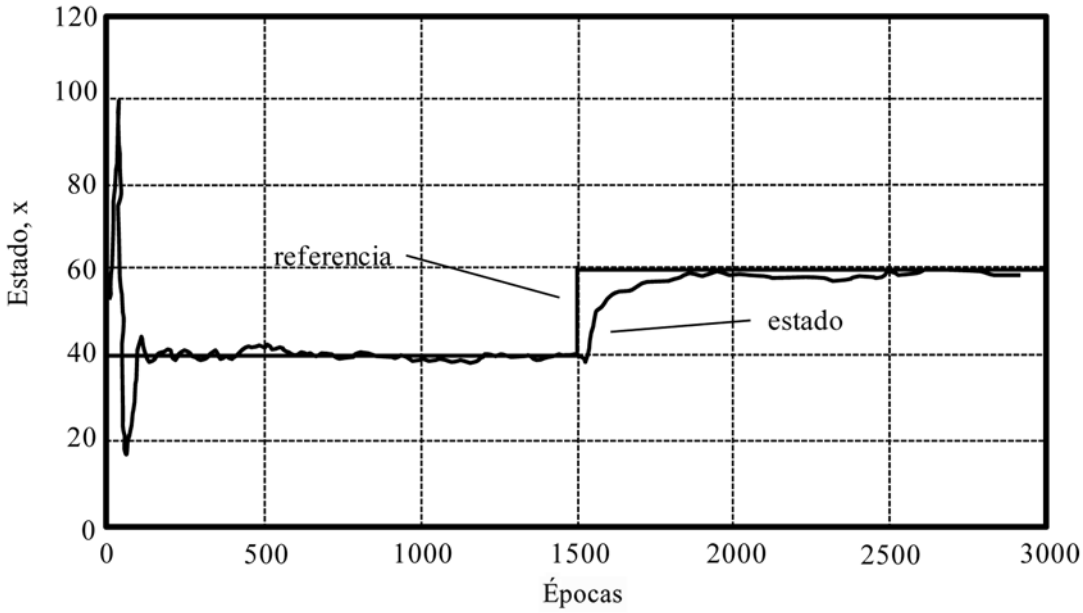

Figura 22. Enfoque Estocástico: Seguimiento de temperatura.
Fuente: (El autor)

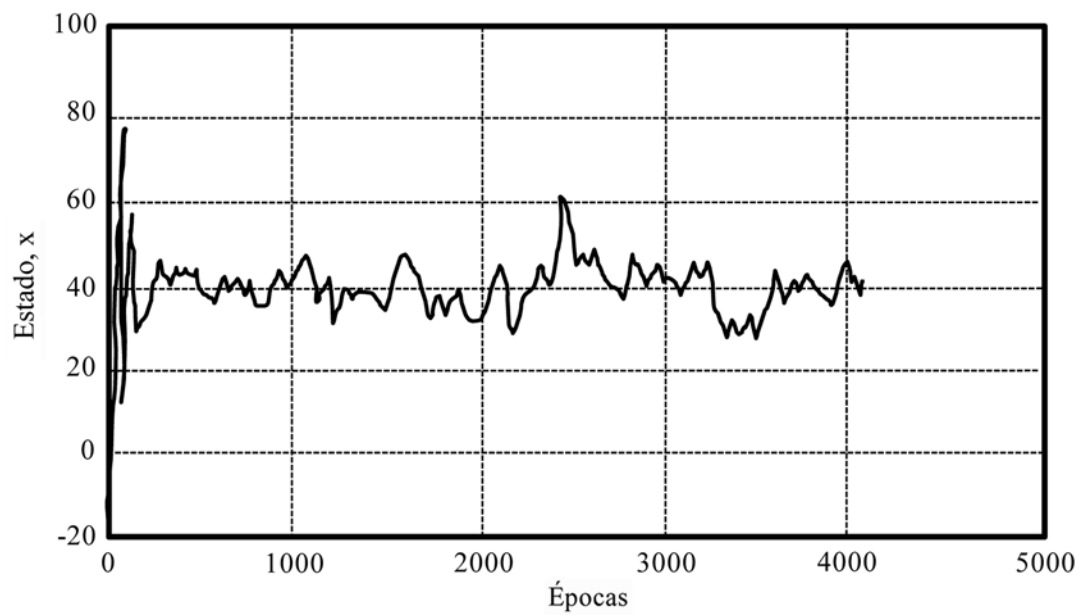

Figura 23. Enfoque Estocástico: Salida con referencia incluyendo ruido.
Fuente: (El autor)

4. DISCUSIÓN

Ambas arquitecturas de PDH aprenden a controlar el sistema térmico. No obstante, el Enfoque Estocástico es más rápido que el enfoque Determinístico. Por otro lado, si la referencia tiene constantes cambios, el Enfoque Determinístico trabaja mejor, ya que no tiene que aprender la función del proceso de nuevo, y porque la red neuronal del modelo de la planta continúa aprendiendo en línea.

El entrenamiento del modelo de la planta es una tarea laboriosa. El proceso simulado en este artículo es muy sencillo. Se eligió de esta manera para poder llevar el control de los procesos internos de las redes neuronales. En la práctica, no obstante, las redes neuronales son útiles en el área del control automático cuando se tienen plantas no lineales con múltiples entradas y salidas. El tiempo de aprendizaje de estas funciones incrementará significativamente, haciendo el Enfoque Estocástico más conveniente.

La selección de la función de refuerzo puede ser también una tarea laboriosa y si la selección correcta no se realiza, la Red SRV podría requerir más épocas para controlar la planta.

No obstante, para sistemas con señales ruidosas, es mejor utilizar el Enfoque Determinístico por su probada superioridad ante el ruido.

5. TRABAJOS FUTUROS

Trabajos futuros deben estudiar el desempeño de estos enfoques en sistemas no lineales – en donde las redes neuronales son más útiles dado que no se requiere linealizar la planta – con múltiples entradas y salidas, para determinar en dónde se desempeñan mejor.

REFERENCIAS BIBLIOGRÁFICAS

H. Asada y S. Liu, “Transfer of human skills to neural net robot controllers”, in Proc. R&A, pp.2442-2448.

D. Psaltis, A. Sideris, y A.A. Yamamura, “A multilayered neural network controller”, IEEE Control Systems Mag., pp. 17-20, April 1988.

K. Narendra y K. Parthasarathy, “Identification and control of dynamical systems using neural networks”, IEEE Trans. Neural Networks, vol. 1, no. 1, pp. 4-27, 1990.

K.S. Narendra, “Adaptive control using neural networks”, en W.T. Miller, R.S. Sutton, y P.J. Werbos, eds., Neural Networks for Control. Cambridge, MA: The MIT Press, pp. 115-142, 1990.

D. A. White y M.I. Jordan, “Optimal Control: A Foundation for Intelligent Control”, en Handbook for Intelligent Control.

P.J. Werbos, “Approximate Dynamic Programming for Real-Time Control and Neural Modelling”, en Handbook for Intelligent Control.

Miller, Sutton y Werbos, “Neural networks for control”, Cambridge, MA: The MIT Press, pp. 67-160,1990.

V. Gullapalli, “A stochastic learning algorithm for real-value functions”, Neural Networks, 3, 671-692,1990.

SOBRE EL AUTOR