

# Dynamic systems control and identification using VSC-based learning algorithms for perceptron networks

*Francklin Rivas-Echeverría y Eliezer Colina-Morles*

*Departamento de Sistemas de Control, Escuela de Ingeniería de Sistemas,  
Facultad de Ingeniería, Universidad de Los Andes, Mérida 5101, Venezuela.  
Tlf: (58) 74 402847, Fax: (58) 74 402846*

## Abstract

In this paper a set of Variable Structure Control (VSC)-based on-line learning algorithms for continuous time two layer and three layer perceptron networks with non-linear and linear activation functions are presented. The proposed algorithms result in a temporal learning capabilities of a neural network with dynamically adjusted weights, and zero convergence of the learning error in a finite time. These learning algorithms are used with identification and control schemes for linear and non linear dynamic systems.

**Key words:** Neural networks, variable structure control, learning algorithms, identification, control.

## Control e identificación de sistemas dinámicos utilizando redes neuronales entrenadas con algoritmos basados en control de estructura variable

### Resumen

En este artículo se proponen algoritmos de entrenamiento en línea basados en control de estructura variable para redes neuronales perceptrónicas de dos y tres capas con funciones de activación lineal y no lineal. Dichos algoritmos proveen un entrenamiento con capacidades de aprendizaje temporal, con ajuste dinámico de los pesos y convergencia a cero del error de entrenamiento en tiempo finito. Estos algoritmos de entrenamiento se utilizan en esquemas de identificación y control de sistemas dinámicos lineales y no lineales.

**Palabras clave:** Redes neuronales, control de estructura variable, algoritmos de entrenamiento, identificación, control.

### 1. Introducción

Un algoritmo adaptativo basado en control de estructura variable para entrenamiento de "Adalines" (neuronas) continuas fue propuesto originalmente por Sira-Ramírez y Colina-Morles en [1, 2]. Este algoritmo resulta en una ley de actualización dinámica de los pesos de interconexión de la neurona de manera tal que se genere

un régimen deslizante [3] sobre la superficie de nivel cero del error de entrenamiento. Las propiedades de convergencia son probadas utilizando los criterios de Lyapunov [1, 2, 4, 5, 6, 7, 8].

El presente trabajo constituye una generalización de los resultados presentados en [1] y [2], ya que en él se proponen algoritmos de entrenamiento basados en control de estructura variable

para redes neuronales de dos y tres capas. Igualmente se presenta una forma de utilizar estos algoritmos de entrenamiento para obtener aproximaciones en línea de las ganancias de los modelos directos e inversos de sistemas dinámicos, con la potencial aplicación en el campo del control usando una variedad de esquemas.

El artículo está organizado de la siguiente manera: Las secciones 2.1 y 2.2 presentan la derivación de las expresiones dinámicas para el ajuste de los pesos de redes neuronales de dos y tres capas con funciones de activación no lineal, respectivamente. Las propiedades de convergencia a cero del error de entrenamiento así como el acotamiento de los pesos de interconexión son probados utilizando argumentos simples de análisis lineal. Particularizaciones de los algoritmos para redes lineales de dos y tres capas son presentadas en las secciones 2.3. y 2.4, respectivamente. La sección 3 contiene un ejemplo de aplicación en el cual se ilustran las bondades de los algoritmos propuestos para realizar seguimiento de señales, así como su robustez ante ruido. En la sección 4 se presenta una aplicación de los algoritmos de entrenamiento basados en control de estructura variable para la identificación temporal de los modelos directos e inversos de sistemas dinámicos. La sección 5 presenta los resultados obtenidos para el control de un sistema no lineal utilizando un esquema basado en el modelo inverso. Finalmente, la sección 6 contiene las conclusiones pertinentes.

## 2. Algoritmos de entrenamiento basados en control de estructura variable

El proceso de entrenamiento supervisado de una red neuronal puede ser interpretado como el problema de encontrar los valores adecuados de sus parámetros (pesos) en búsqueda de aproximar, de la mejor manera, una función dada. Matemáticamente hablando, este problema puede ser formulado como sigue:

Dada una función  $f(x)$ , definida sobre un conjunto  $X$  y una función de aproximación  $\hat{f}(x, W)$ , que depende de  $W \in P$  y del vector real  $x$ , encontrar los parámetros  $W^*$  tal que  $\|\hat{f}(x, W^*) - f(x)\| \leq \|\hat{f}(x, W) - f(x)\| < \varepsilon, \forall W \in P$ , y  $\varepsilon > 0$ .

Las redes neuronales perceptrónicas son funciones de aproximación típicas  $\hat{f}: \mathfrak{R}^n \rightarrow \mathfrak{R}^p$ , y algunas de sus topologías clásicas comprenden el uso de una o más capas de neuronas.

En el caso de redes neuronales de dos capas, la función de aproximación es lineal respecto a la base de funciones  $\{\Gamma_i\}_{i=1}^m$  de las entradas originales "x" de la red. Es decir,  $\hat{f}(x, W) = W_o^T \Gamma(W_i^T x)$ . En el caso de tres capas, la función de aproximación es de la forma  $\hat{f}(x, W) = W_o^T \Gamma(W_h^T \Gamma(W_i^T x))$ , donde  $W_i$ ,  $W_h$  y  $W_o$  son los pesos de las capas de entrada, intermedia y de salida respectivamente.

La siguiente sección contiene el desarrollo de algoritmos de entrenamiento basados en control de estructura variable para redes neuronales de dos capas y se prueba las propiedades de convergencia a cero del error de entrenamiento y el acotamiento de los pesos de interconexión.

### 2.1. Red neuronal no lineal de dos capas

La Figura 1 ilustra una red neuronal completamente conectada de dos capas. De allí, las siguientes ecuaciones se pueden verificar con facilidad:

$$Y_o(t) = (W_o(t))^T Z_1(t) \quad (2.1)$$

$$Z_1(t) = \Gamma(Y_1(t)) \quad (2.2)$$

$$Y_1(t) = (W_i(t))^T X_a(t) \quad (2.3)$$

donde:

$$W_o(t) \in \mathfrak{R}^{n \times p}; Z_1(t) \in \mathfrak{R}^n; W_i(t) \in \mathfrak{R}^{(n+1) \times n}$$

El vector  $(n+1)$ -dimensional  $X_a(t)$  está compuesto de "n" señales de entrada a la red y un elemento constante llamado umbral o desviación. La función de activación no lineal " $\Gamma$ " se asume diferenciable. El vector "Yo" es la salida de la red, y las matrices  $W_o$  y  $W_i$  representan los pesos de interconexión de la capa de salida y de la capa de entrada respectivamente.

El error de entrenamiento está definido por:

$$E(t) = (e_1(t), \dots, e_i(t), \dots, e_p(t))^T = Y_d(t) - Y_o(t) \quad (2.4)$$

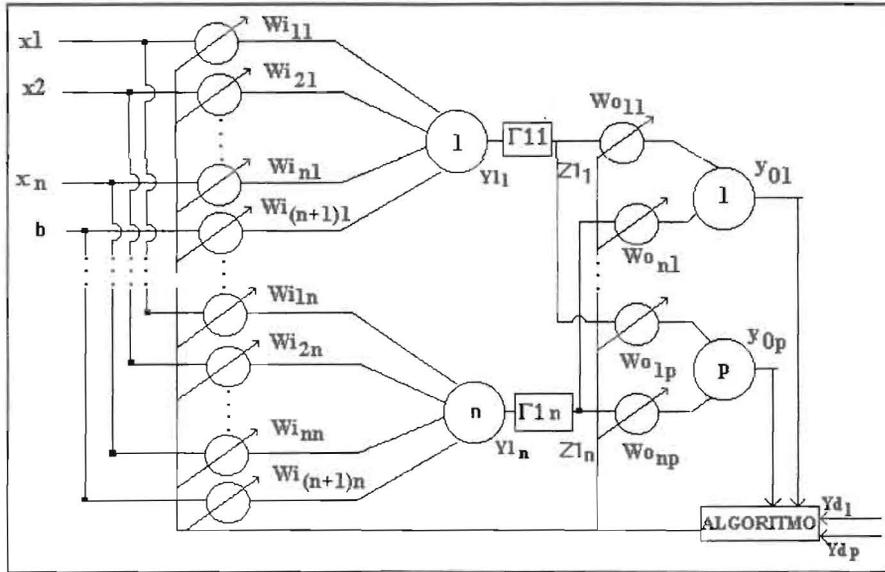


Figura 1. Red Neuronal de dos capas.

donde “ $Yd(t)$ ” constituye el vector de entrenamiento.

El siguiente teorema es formulado considerando que el vector de entrada “ $Xa(t)$ ”, el vector de entrenamiento “ $Yd(t)$ ” y sus derivadas están acotadas por:

$$\|Xa(t)\| < B_x, \quad \dot{X}a(t) < B_x \quad (2.5)$$

$$\|Yd(t)\| < B_y, \quad \dot{Y}d(t) < B_y \quad (2.6)$$

También se asume que los elementos de las matrices  $Wo$  y  $Wi$  están acotados por  $\|Wo\| \leq B_{wo}$  y  $\|Wi\| \leq B_{wi}$ , respectivamente. Este hecho será probado posteriormente.

**2.1.1. Teorema**

Si las leyes de actualización para los pesos de interconexión de la red neuronal de dos capas, ilustrada en la Figura 1, son seleccionadas según:

$$Wo(t) = -Wo(t) + \frac{Z1(t)}{Z1(t)^T Z1(t)} [k \text{sign}(E(t))] \quad (2.7)$$

$$Wi(t) = -Wi(t) + \frac{Xa(t)}{Xa(t)^T Xa(t)} Z1(t)^T \left( \frac{\partial \Gamma(y1(t))}{\partial y1(t)} \right)^{-1} \quad (2.8)$$

donde “ $k$ ” es una ganancia escalar que satisface la condición:

$$k > \eta + B_y - B_{wo} B_{wi} (B_x - B_x) \text{ con } \eta > 0 \quad (2.9)$$

entonces, para cualquier vector de error de entrenamiento inicial “ $E(0)$ ”, las dinámicas del error de aprendizaje convergerán a cero en tiempo finito “ $tr$ ” estimado por:

$$tr_i \leq \frac{|e_i(0)|}{\eta} \quad (2.10)$$

y un régimen deslizante las mantendrá sobre la superficie  $e_i(t)=0, i=1,2,\dots,p$  para todo  $t > tr$ .

**Prueba**

Obsérvese que el error de entrenamiento está definido por:

$$E(t) = Yd(t) - Yo(t) = Yd(t) - Wo(t)^T Z1(t) \quad (2.11)$$

y su derivada está dada por:

$$\begin{aligned} \dot{E}(t) = \dot{Y}d(t) - \dot{W}o(t)^T Z1(t) - Wo(t)^T \left( \frac{\partial \Gamma(Y1(t))}{\partial Y1(t)} \right) \dot{W}i^T Xa \\ - Wo(t)^T \left( \frac{\partial \Gamma(Y1(t))}{\partial Y1(t)} \right) Wi^T \dot{X}a \end{aligned} \quad (2.12)$$

Sustituyendo las ecuaciones (2.7) y (2.8) en la ecuación (2.12) se obtiene:

$$\dot{E}(t) = \dot{Y}d(t) - Wo(t)^T \left( \frac{\partial \Gamma(Y1(t))}{\partial Y1(t)} \right) Wi^T (\dot{X}a - Xa) - k \text{sign}(E(t)) \quad (2.13)$$

Considérese una función candidata de Lyapunov dada por:

$$v(E(t)) = \frac{1}{2} \|E(t)\|^2 \quad (2.14)$$

Es fácil demostrar que la derivada de (2.14) es:

$$\dot{v}(E(t)) = E(t) \left\{ \dot{Y}d(t) - W_o(t)^T \left( \frac{\partial \Gamma(Y_1(t))}{\partial Y_1(t)} \right) W_i^T (\dot{X}a - Xa) - k \operatorname{sign}(E(t)) \right\} \quad (2.15)$$

Considerando las ecuaciones (2.5), (2.6) y la desigualdad de Schwartz, se obtiene la siguiente expresión:

$$\dot{v}(E(t)) \leq |E(t)| \left[ B_y - B_{wo} B_{wi} (B_x - B_x) \right] - k |E(t)| \leq [B_y - B_{wo} B_{wi} (B_x - B_x) - k] |E(t)| \quad (2.16)$$

la cual es semi-definida negativa si y sólo si  $k > \eta + B_y - B_{wo} B_{wi} (B_x - B_x)$  con  $\eta > 0$ , y entonces las trayectorias del error de entrenamiento convergerán a cero.

Por otro lado, nótese que la condición para que se genere un régimen deslizante [3] sobre las superficies  $e_i(t)=0$  puede ser expresada según:

$$\dot{e}_i(t) \leq \eta \operatorname{sign}(e_i(t)) \quad (2.17)$$

Entonces:

$$\int_0^\tau e_i(t) dt \leq \eta \int_0^\tau \operatorname{sign}(e_i(t)) dt \quad (2.18)$$

y para  $\tau < tr$

$$e_i(\tau) - e_i(0) \leq -\eta \tau \operatorname{sign}(e_i(0)) \quad (2.19)$$

Cuando  $t=tr$ , el valor de " $e_i(t)$ " será igual a cero y:

$$e_i(0) \geq \eta \tau \operatorname{sign}(e_i(0)) \quad (2.20)$$

multiplicando ambos lados de la ecuación (2.20) por " $\operatorname{sign}(e_i(0))$ " se tiene:

$$|e_i(0)| \geq \eta tr \quad (2.21)$$

y entonces

$$tr_i \leq \frac{|e_i(0)|}{\eta} \quad (2.22)$$

### 2.1.2. Acotamiento de la solución para los pesos de interconexión

La solución de las ecuaciones diferenciales (2.7) y (2.8) puede ser obtenida usando el método de la variación de las constantes [6, 7]. Dado que los autovalores de las ecuaciones (2.7) y (2.8) son reales estrictamente negativos (ya que son iguales a -1), y considerando las condiciones de acotamiento de las señales de entrada, entrenamiento y sus respectivas derivadas, dadas por las expresiones (2.5) y (2.6), es fácil probar el acotamiento de la solución para los pesos de interconexión.

### 2.2. Red neuronal no lineal de tres capas

La Figura 2 muestra una red neuronal de tres capas. De allí, se puede verificar las siguientes expresiones:

$$Y_o(t) = (W_o(t))^T Z_2(t) \quad (2.23)$$

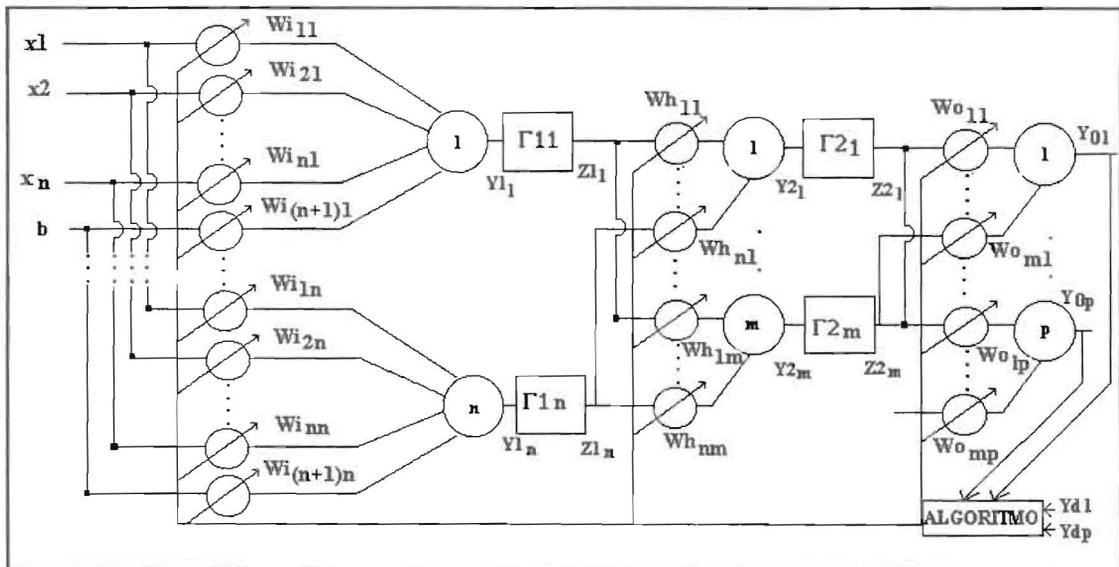


Figura 2. Red neuronal de tres capas.

$$Z2(t) = \Gamma(Y2(t)) \quad (2.24)$$

$$Y2(t) = (Wh(t))^T Z1(t) \quad (2.25)$$

$$Z1(t) = \Gamma(Y1(t)) \quad (2.26)$$

$$Y1(t) = (Wi(t))^T Xa(t) \quad (2.27)$$

donde:  $Wo(t) \in \mathbb{R}^{m \times p}$ ;  $Z2(t) \in \mathbb{R}^m$ ;  $Wh(t) \in \mathbb{R}^{n \times m}$ ;  $Z1(t) \in \mathbb{R}^n$  y  $Wi(t) \in \mathbb{R}^{(n+1) \times n}$

Se asumirá que las condiciones de acotamiento dadas por las expresiones (2.5) y (2.6) siguen siendo válidas.

### 2.2.1. Teorema

Si las leyes de actualización de los pesos de interconexión de la red neuronal mostrada en la Figura 2 son seleccionadas de acuerdo a la solución de las ecuaciones:

$$\dot{Wo}(t) = -Wo(t) + \frac{Z2(t)}{Z2(t)^T Z2(t)} [k \text{sign}(E(t))] \quad (2.28)$$

$$\dot{Wh}(t) = -Wh(t) + \frac{Z1(t)}{Z1(t)^T Z1(t)} Z2(t)^T \left( \frac{\partial \Gamma(Y2(t))}{\partial Y2(t)} \right)^{-1} \quad (2.29)$$

$$\dot{Wi}(t) = -Wi(t) + \frac{Xa(t)}{Xa(t)^T Xa(t)} Z1(t)^T \left( \frac{\partial \Gamma(Y1(t))}{\partial Y1(t)} \right)^{-1} \quad (2.30)$$

donde "k" es una ganancia escalar que satisface la condición:

$$k > \eta + B_y - B_{wo} B_{wh} (B_x - B_x), \text{ con } \eta > 0 \quad (2.31)$$

entonces, dada cualquier condición inicial arbitraria "E(0)", el error de entrenamiento "E(t)" convergerá a cero en tiempo finito "tr" estimado por:

$$tr_i \leq \frac{|e_i(0)|}{\eta} \quad (2.32)$$

y un régimen deslizante se mantendrá sobre la superficie  $e_i(t)=0$ ,  $i=1,2,\dots,p$ ; para todo  $t > tr$ .

### Prueba:

La prueba de este teorema sigue los mismos lineamientos de la prueba del Teorema 2.1.1.

### 2.2.2. Acotamiento de la solución para los pesos de interconexión

Para probar el acotamiento de las soluciones de las ecuaciones diferenciales (2.28), (2.29)

y (2.30), se hace uso de los mismos argumentos planteados en la sección 2.1.2.

### 2.3. Red neuronal lineal de dos capas

En esta sección se presenta una particularización del algoritmo de entrenamiento propuesto en la sección 2.1. Esto es, se asumirá que cada neurona de la topología ilustrada en la Figura 1 es una neurona lineal (es decir, posee una función de activación lineal). Tal consideración resulta en una relación lineal entre las entradas de la red y sus salidas. Bajo estas circunstancias, las ecuaciones dinámicas que describen el comportamiento de los pesos de interconexión son:

$$\dot{Wo}(t) = -Wo(t) + \frac{Y1(t)}{Y1(t)^T Y1(t)} [k \text{sign}(E(t))] \quad (2.33)$$

$$\dot{Wi}(t) = -Wi(t) + \frac{Xa(t)}{Xa(t)^T Xa(t)} Y1(t)^T \quad (2.34)$$

### 2.4. Red neuronal lineal de tres capas

Las ecuaciones presentadas a continuación, describen la ley de actualización de los pesos de interconexión de redes de tres capas con neuronas que poseen funciones de activación lineal.

$$\dot{Wo}(t) = -Wo(t) + \frac{Y2(t)}{Y2(t)^T Y2(t)} [k \text{sign}(E(t))] \quad (2.35)$$

$$\dot{Wh}(t) = -Wh(t) + \frac{Y1(t)}{Y1(t)^T Y1(t)} Y2(t)^T \quad (2.36)$$

$$\dot{Wi}(t) = -Wi(t) + \frac{Xa(t)}{Xa(t)^T Xa(t)} Y1(t)^T \quad (2.37)$$

### 3. Ejemplo de Aplicación

En esta sección se presentan los resultados obtenidos mediante simulaciones computacionales del seguimiento de la señal analógica  $2\cos(5t)\sin(3t)$  (mostrada en la Figura 3) utilizando redes neuronales de dos y tres capas con funciones de activación lineal y no lineal. Las Figuras 4 y 5 ilustran la convergencia del error de entrenamiento para redes de dos y tres capas con diferentes valores de la ganancia escalar "k" ( $k=10, 40, 200$ ). Nótese que mientras más grande es el valor de k, menor es el tiempo de convergen-

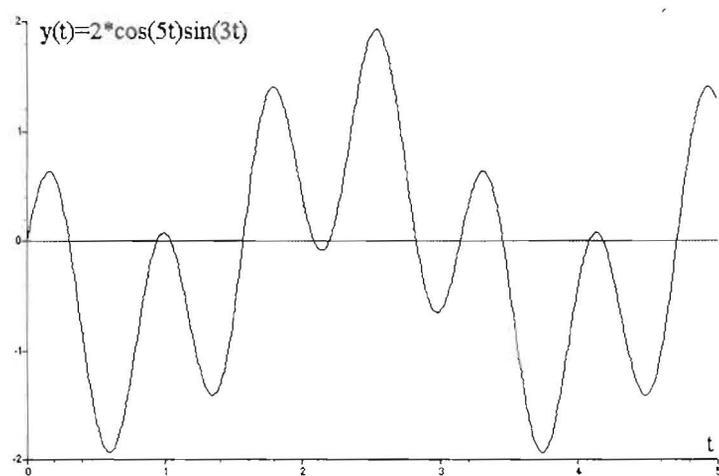


Figura 3. Señal analógica a seguir.

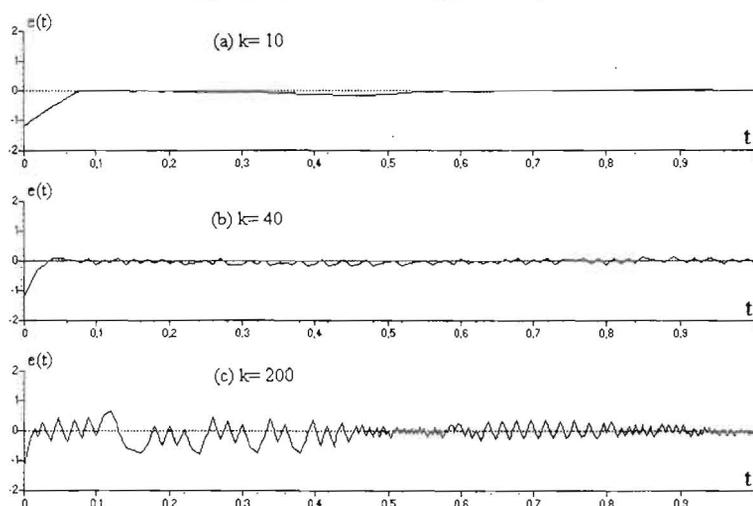


Figura 4. Error de entrenamiento para R.N. de 2 capas con varios valores de  $k$ .

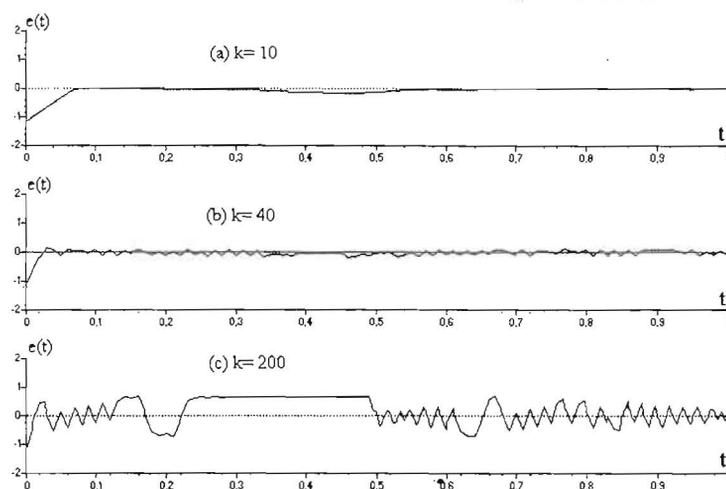


Figura 5. Error de entrenamiento para R.N. de 3 capas con varios valores de  $k$ .

cia a cero del error en ambas topologías pero se generan mayores oscilaciones (chattering) sobre la superficie de deslizamiento. Esto ocurre debido a que  $k$  es el parámetro que indica la velocidad de convergencia a la superficie de deslizamiento, así que al ser este parámetro mayor el error convergerá a cero más rápidamente pero también

tendrá mayor tendencia a alejarse de la superficie. Las Figuras 6 y 7 muestran el comportamiento dinámico de los pesos de interconexión para redes neuronales lineales de dos y tres capas.

Por otro lado, la robustez de los algoritmos propuestos ante perturbaciones a la entrada de la red, como la ilustrada en la Figura 8 (ruido

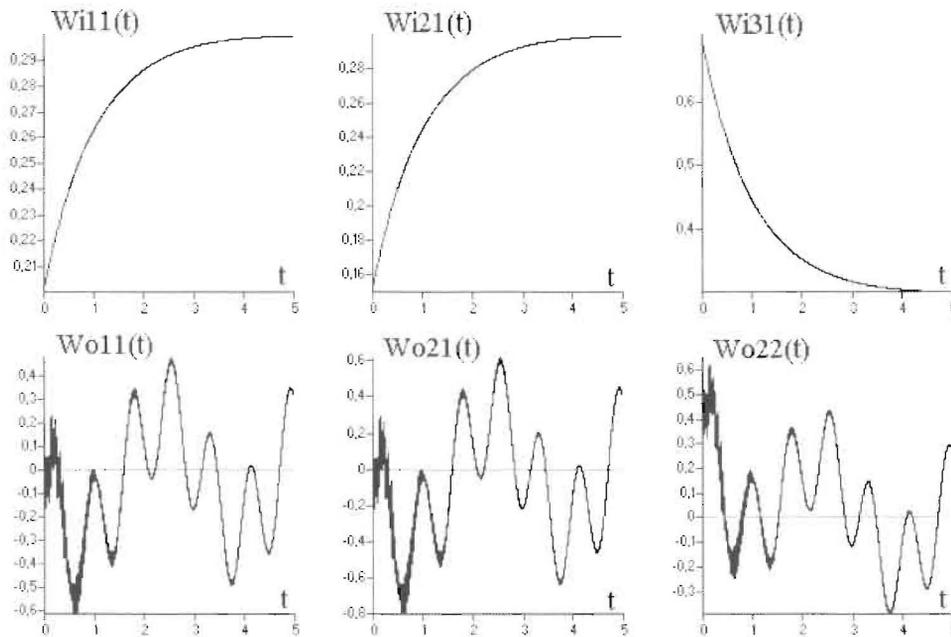


Figura 6. Comportamiento dinámico de los pesos de interconexión de una R.N. de dos capas.

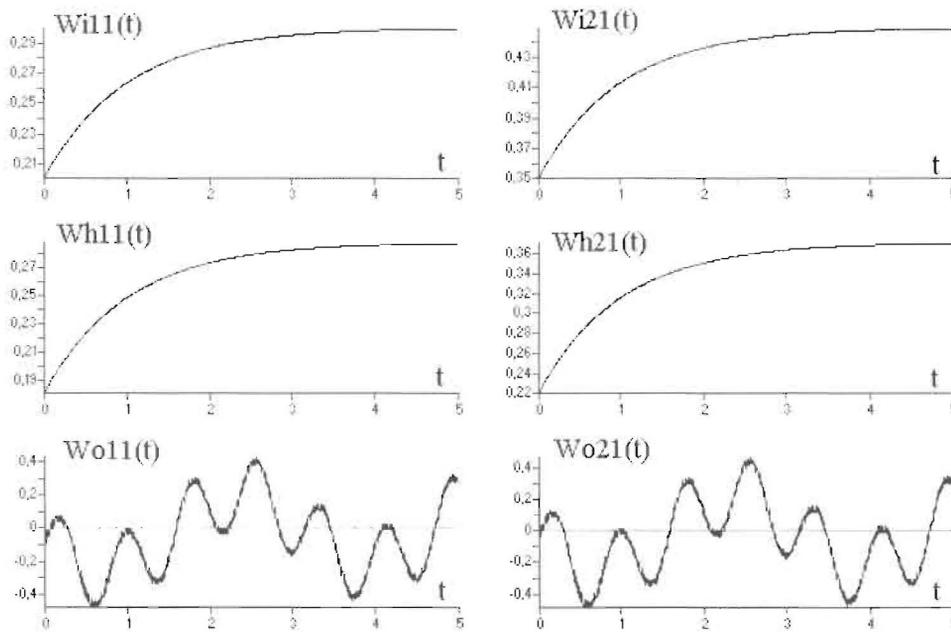


Figura 7. Comportamiento dinámico de los pesos de interconexión de una R.N. de tres capas.

gaussiano con media -1 y desviación estandar 0.5), puede ser apreciada en la Figura 9. En dicha figura, se presenta el error de entrenamiento para redes de dos y tres capas en presencia de la señal de perturbación para diferentes valores de la ganancia "k". En la Figura 10 se puede comparar el error de seguimiento obtenido utilizando una red lineal y una no lineal de dos capas. Nótese que no hay diferencia significativa en el comportamiento de ambas topologías; lo cual hace recomendar el uso de la versión lineal de los algoritmos de entrenamiento, ya que se consigue re-

laciones mucho más sencillas entre la entrada y la salida de la red neuronal.

#### 4. Identificación de los operadores directos e inversos de transferencia de sistemas dinámicos

El problema de identificación de sistemas dinámicos via redes neuronales puede ser planteado de la siguiente manera:

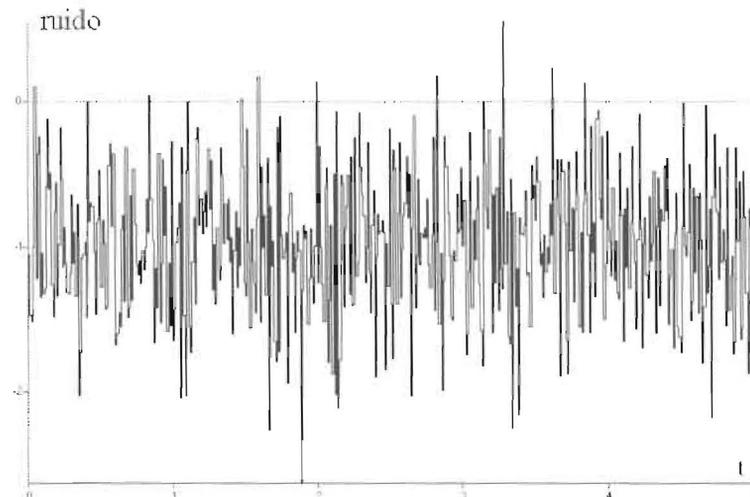


Figura 8. Señal de ruido a la entrada de la R.N.

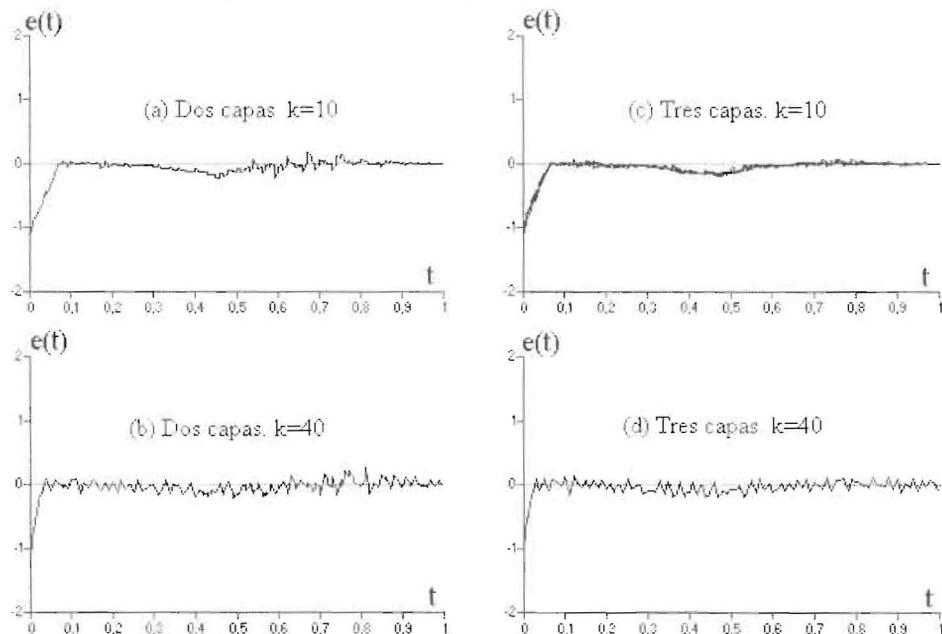


Figura 9. Error de entrenamiento para R.N. de dos y tres capas con  $k=40$  y señal de perturbación.

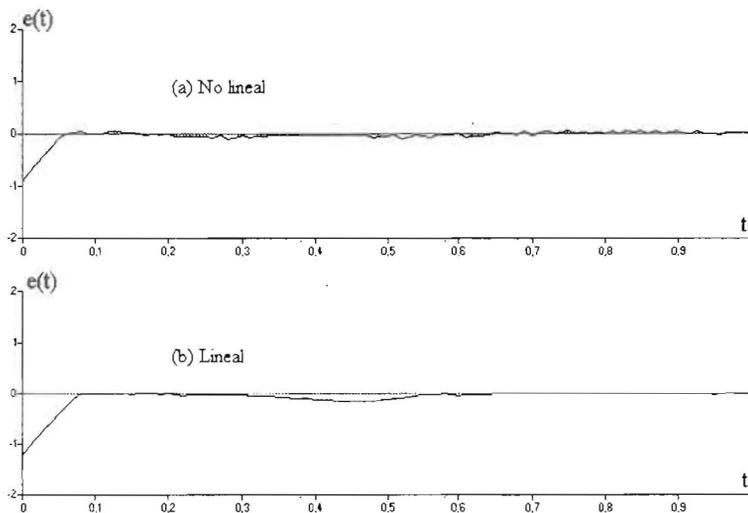


Figura 10. Error de entrenamiento para R.N. de dos capas con  $k=200$  usando neuronas no lineales y lineales.

Considérese un sistema causal dado por:

$$y(t)=F(u(t)) \tag{4.1}$$

donde  $u(t) \in U$  y  $y(t) \in Y$  son las entradas y salidas del sistema respectivamente, y  $F(\cdot)$  es un operador que aplica los elementos de  $U$  en  $Y$ .

Ahora, considérese un modelo neuronal dado por:

$$\hat{y}(t) = \hat{F}(u(t), W(t)) \tag{4.2}$$

entonces, es necesario encontrar los pesos ( $W(t)$ ) de manera tal que:

$$\|y(t) - \hat{y}(t)\| = \|F(u(t)) - \hat{F}(u(t), W(t))\| < \varepsilon \tag{4.3}$$

con  $\varepsilon > 0$ .

Los pesos serán actualizados utilizando los algoritmos de entrenamiento basados en control de estructura variable presentado en las secciones anteriores.

**4.1. Identificación del operador directo de transferencia**

La Figura 11 ilustra un esquema para la identificación del operador de transferencia directo, donde el error de entrenamiento de la red neuronal es representado por la diferencia entre la salida del sistema original y la salida neuronal. Este esquema de identificación no estima los parámetros del sistema, sólo provee una identifica-

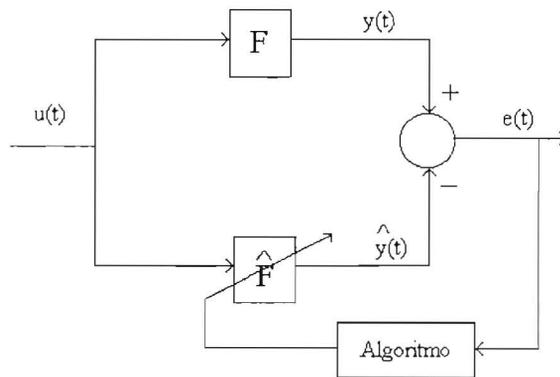


Figura 11. Identificación del modelo directo.

ción temporal de la ganancia del operador de transferencia del sistema.

Nótese que si se considera topologías donde se utilicen solamente neuronas lineales, la salida de la red neuronal puede ser expresada como:

$$Y_o(t) = Wd^T Xa(t) \tag{4.4}$$

donde " $Wd^T(t)$ " es una matriz equivalente de pesos, la cual depende de la topología seleccionada (por ejemplo, si se considera el caso de una red de tres capas con neuronas lineales,  $Y_o(t) = (W_o(t))^T (W_h(t))^T (W_i(t))^T Xa(t)$ , donde  $(W_o(t))^T$ ,  $(W_h(t))^T$  y  $(W_i(t))^T$  son las matrices de pesos de la capa de salida, intermedia y de entrada respectivamente), y  $Xa(t)$  es el vector de entradas a la red neuronal que contiene la entrada al sistema y algunos de sus valores retrasados (es decir,  $Xa(t) = (u(t), u(t-\tau), u(t-2\tau), \dots, u(t-r\tau))^T$ ).

Para efectos de simplicidad, se asume que para un intervalo de tiempo de longitud  $nT = T$ , el valor de  $u(t)$  no cambia significativamente. Bajo estas circunstancias, el modelo neuronal para un sistema de una entrada y una salida, puede ser escrito como:

$$Y_o(t) = \sum_{i=1}^{n+1} Wd_i u(t - (i - 1)T) \approx \sum_{i=1}^{n+1} Wd_i u(t) \quad (4.5)$$

Usando las leyes de actualización para los pesos dadas por las ecuaciones (2.7) y (2.8) o (2.16), (2.17) y (2.18), la diferencia entre la salida del sistema y la salida de la red neuronal será cero en tiempo finito ( $tr$ ) y entonces  $Y_o(t) = Y(t)$  para  $t \geq tr$ , lo cual implica que:

$$Y(t) \approx \left( \sum_{i=1}^{n+1} Wd_i \right) u(t) \quad \text{para } t \geq tr \text{ y } nT \leq t \leq (n+1)T \quad (4.6)$$

Nótese de (4.6) que el factor  $\left( \sum_{i=1}^{n+1} Wd_i \right)$  representa la ganancia del operador directo de transferencia del sistema dinámico.

#### 4.2. Identificación del operador inverso de transferencia

La Figura 12 muestra un esquema de identificación para el operador inverso de transferencia basado en redes neuronales. Dicho esquema es conceptualmente similar al esquema para identificación del operador directo de transferencia. Utilizando una formulación análoga a la desarrollada en la sección anterior, la ganancia del operador inverso de transferencia puede ser expresado de la siguiente manera:

$$\frac{u(t)}{Y(t)} \approx \sum_{i=1}^{n+1} Wd_i \approx \hat{P}^{-1} \quad \text{para } t \geq tr \text{ y } nT \leq t \leq (n+1)T \quad (4.7)$$

Es importante tomar en cuenta algunas consideraciones antes de utilizar este esquema: Primero, la invertibilidad y unicidad del modelo del proceso y segundo, la estabilidad de tal modelo.

### 5. Control basado en modelo inverso utilizando redes neuronales

La Figura 13 muestra un esquema de control basado en el modelo inverso. Dicho esquema utiliza una red neuronal para la identificación de la ganancia instantánea del operador inverso de transferencia.

De la Figura 13 se obtiene la siguiente expresión:

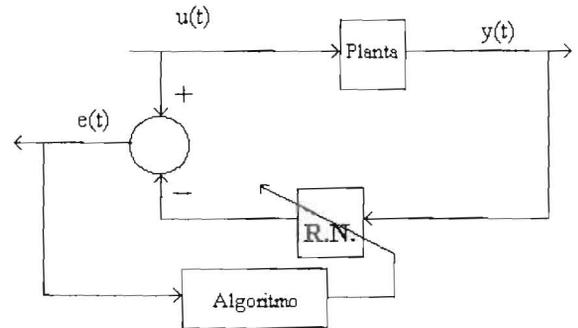


Figura 12. Identificación del modelo inverso.

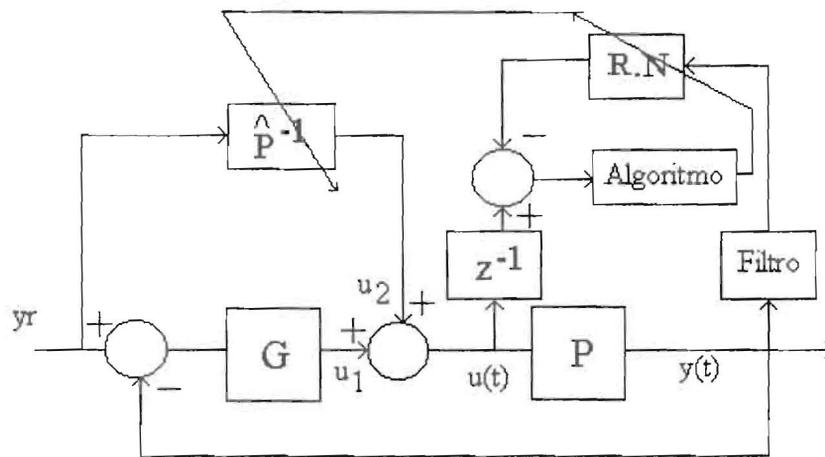


Figura 13. Esquema de control basado en modelo inverso.

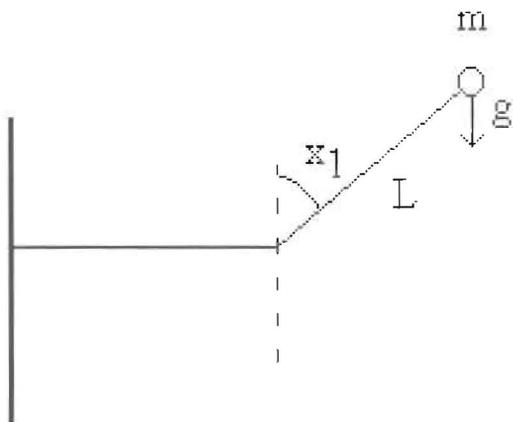


Figura 14. Péndulo de Kapitza.

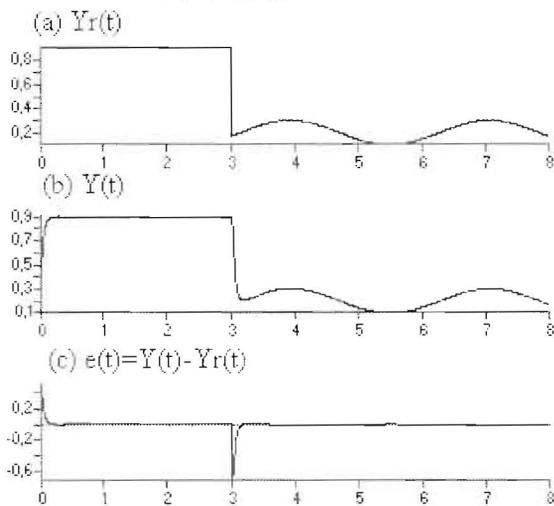


Figura 15. Respuesta del Sistema con k=200.

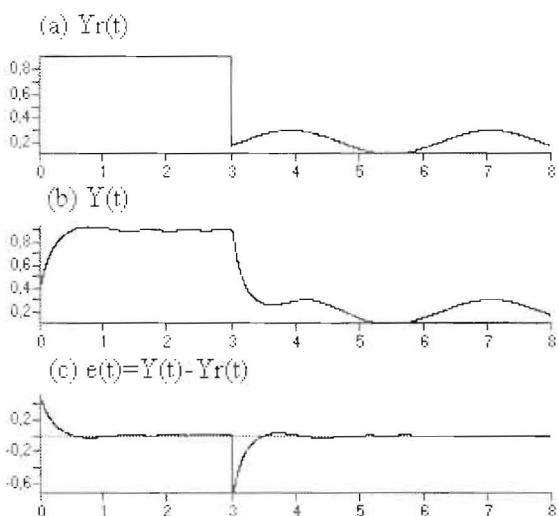


Figura 16. Respuesta del Sistema con k=50.

$$y(t) = \frac{P\hat{P}^{-1} + PG}{1 + PG} yr(t) \tag{5.1}$$

Una vez que la identificación de la ganancia asociada al operador inverso de transferencia basada en redes neuronales ha sido alcanzada (es decir, después de un tiempo  $t_r$ ), la salida del sistema  $y(t)$  seguirá a la señal de referencia  $yr(t)$ . Esto es, ya que  $P\hat{P}^{-1} \approx 1$  entonces  $y(t) \approx yr(t)$ .

El filtro "G" presentado en el esquema propuesto, influencia la respuesta transitoria del sistema y en caso de que el sistema sea inestable podría ser usado para compensar el efecto de polos inestables del sistema. El diseño del filtro "G" debe cuidar que el producto  $P^{-1}G$  sea estable y propio y requiere del conocimiento sobre el orden dinámico del sistema.

### 5.1. Ejemplo de un sistema no lineal

Para mostrar el comportamiento del esquema de control basado en el modelo inverso, se presenta a continuación el problema de controlar la posición angular del Péndulo de Kapitza [1, 7].

La Figura 14 muestra el Péndulo de Kapitza, y la representación del modelo no lineal en el espacio de estado está dada por:

$$\begin{bmatrix} \dot{x}_1(t) \\ \dot{x}_2(t) \\ \dot{x}_3(t) \end{bmatrix} = \begin{bmatrix} x_2(t) + \frac{u(t)}{L} \sin x_1(t) \\ \left( g - \frac{u^2(t)}{L^2} \cos x_1(t) \right) \sin x_1(t) - \frac{u(t)}{L} x_2(t) \cos x_1(t) \\ u(t) \end{bmatrix} \tag{5.2}$$

$$y(t) = x_1(t) \tag{5.3}$$

donde:  $x_1(t)$  es el ángulo de la barra respecto al eje vertical;  $x_2(t)$  es proporcional al impulso generalizado;  $x_3(t)$  es la posición vertical del punto de suspensión;  $u(t)$  es la velocidad del punto de suspensión;  $G$  es la aceleración de Gravedad; y  $L$  es la longitud de la barra.

Nótese en las Figuras 15 y 16 que mientras más grande sea el valor de "k", menor será el tiempo en que la salida del sistema alcance la señal de referencia. Pero mientras más pequeño sea el tiempo de convergencia, el sistema requerirá de señales de control más grandes (Figura 17).

## 6. Conclusiones

En este trabajo se proponen algoritmos de entrenamiento basados en control de estructura variable para redes neuronales de dos y tres capas con funciones de activación lineal y no lineal. Se ha probado, usando el criterio de Lyapunov, que las dinámicas del error de entrenamiento convergen a cero en tiempo finito, el cual depende de la condición inicial del error y de la selección de un parámetro de ganancia  $k$ , el cual mientras más grande sea menor será el tiempo de convergencia a cero del error, pero se generarán mayores oscilaciones (chattering) sobre la superficie de deslizamiento.

La actualización de los pesos de interconexión se obtiene por medio de la solución de ecuaciones diferenciales lineales, y resulta en valores acotados para cada uno de los pesos.

También se propone el uso de los algoritmos de entrenamiento en línea basados en control de estructura variable para redes neuronales de dos y tres capas, para la identificación temporal de la ganancia de los operadores directos e inversos de transferencia de sistemas dinámicos desconocidos usados en algún esquema de control.

## Referencias Bibliográficas

1. H. Sira-Ramírez and E. Colina-Morles, Adaptive Learning in Adalines: A Sliding mode Control Approach. *IEEE Transactions on Circuits and Systems*, Vol. 42 N°12, December, 1995.
2. E. Colina-Morles, Identification and Control of Dynamical Systems using Neural Networks. Ph.D. Thesis, University of Sheffield, England, December 1993.
3. H. Sira-Ramírez, On the Sliding Mode Control of Nonlinear Systems. *Systems and Control Letters*, Vol. 19, pp. 302-312, 1992
4. F. Rivas-Echeverría and E. Colina-Morles, Variable structure Control based algorithm for perceptron networks. *Proceeding of IA-STD International Conference on Modelling and Simulation*, April, 1996
5. F. Rivas-Echeverría and E. Colina-Morles, Variable structure control based on-line learning design for continuous time multi-layer network. *Proceedings of the 1996 IEEE International Symposium on Intelligent Control*, Sept. 1996
6. J. Kuchewsky; S. Hui and S. Zak, Application of Feedforward Neural Networks to Dynamical System Identification and Control. *IEEE Transactions on Control Systems Technology*, Vol. 1, N°1, March 1993.
7. E. Colina-Morles and N. Mort, Inverse Model Neural Network based control of Dynamic Systems. *Journal of Systems Engineering*, 1993
8. W.J. Rugh, *Linear Systems Theory*. (New Jersey: Prentice Hall, 1992).

Recibido el 16 de Marzo de 1998

En forma revisada el 2 de Febrero de 2000

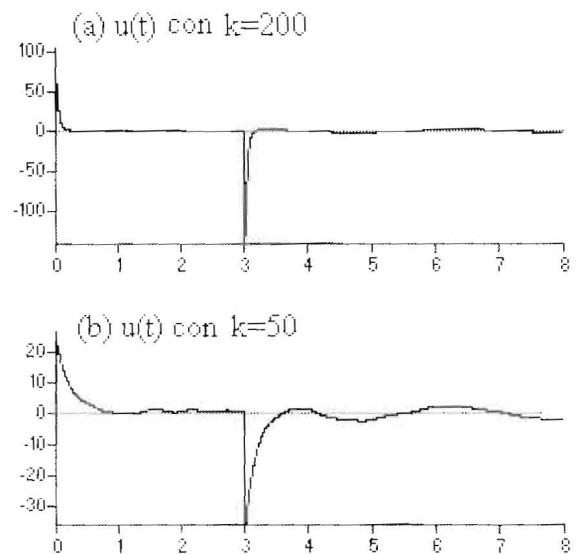


Figura 17. Velocidad del punto de suspensión.