

# IMPLEMENTACION DE UN CONTROLADOR DE SEGUIMIENTO CON REDES NEURONALES PARA UN BRAZO FLEXIBLE SENCILLO: COMPARACION CON CONTROLADORES PD Y PID

Luis B. Gutiérrez\* and Frank L. Lewis\*\*

\* Facultad de Ingeniería Eléctrica y Electrónica  
Universidad Pontificia Bolivariana  
Medellín, Colombia

\*\* Automation and Robotics Research Institute  
The University of Texas at Arlington  
7300 Jack Newell Blvd. South  
Fort Worth, TX 76118-7115, USA

**Resumen:** El objetivo de este trabajo es mostrar los resultados de la implementación práctica de un controlador de seguimiento con redes neuronales para un brazo flexible sencillo y comparar su desempeño con el de controladores estándar PD y PID. El controlador se compone de un lazo exterior de seguimiento PD, un lazo interior de perturbación singular para estabilizar la dinámica de los modos flexibles y un lazo interior con una red neuronal para linearizar la dinámica lenta. La red neuronal no requiere entrenamiento fuera de línea. Se muestra que el desempeño del controlador propuesto es mucho mejor que el de los controladores PD o PID. En las pruebas se agregó un término de fricción adicional para demostrar la habilidad de la red neuronal para aprender la dinámica no lineal.

**Abstract:** The objective of this paper is to show the results of the practical implementation of a neural network tracking controller on a single flexible link and compare its performance to that of PD and PID standard controllers. The NN controller is composed of an outer PD tracking loop, a singular perturbation inner loop for stabilization of the fast flexible mode dynamics, and a neural network inner loop used to feedback linearize the slow pointing dynamics. No off-line training or learning is needed for the NN. It is shown that the tracking performance of the NN controller is far better than that of the PD or PID standard controllers. An extra friction term was added in the tests to demonstrate the ability of the NN to learn unmodeled nonlinear dynamics.

**Palabras clave/Keywords:** Neural Net Tracking Controller, Single Flexible Link

## 1. INTRODUCCION

En la literatura reciente se han propuesto muchos controladores con redes neuronales para brazos

robóticos y otros sistemas no lineales (Chen and Liu, 1994; Narendra and Annaswamy, 1987; Narendra, 1991; Polycarpou and Ioannou, 1991; Rovithakis and Christodoulou, 1994; Sadegh, 1993; Sanner and

Slotine, 1991). El desempeño de estos controladores con redes neuronales ha sido cuestionado, a pesar de que varias de estas referencias proveen pruebas de estabilidad. En este trabajo se presenta la implementación del controlador con redes neuronales derivado por Yesildirek, *et al.* (1994) sobre un brazo robótico flexible sencillo. Se encuentra que el controlador con redes neuronales se comporta mucho mejor que los controladores estándar PD y PID, incluso para el caso del brazo de un elemento que es lineal excepto por los efectos de fricción no lineales.

El control de brazos robóticos flexibles pertenece a una clase de problemas caracterizados por la reducida efectividad de control y la presencia de una dinámica adicional inestable. Algunos otros problemas en esta categoría son las estructuras espaciales de gran escala, los puentes grúa y otros procesos industriales. El requerimiento de controladores más rápidos y con mayor precisión introduce un reto que los investigadores han enfrentado de diferentes maneras.

Algunos investigadores (Madhavan and Singh, 1991; Wang and Vidyagar, 1991) han observado que la dinámica aproximada de los brazos robóticos flexibles puede linearizarse usando linearización por retroalimentación de entrada-salida pero que la **dinámica natural no es asintóticamente estable** cuando la posición del extremo es tomada como la salida. Para controlar el brazo se definió una salida modificada que daba lugar a una dinámica natural estable. Sin embargo, esta salida no corresponde a objetivos prácticos de seguimiento excepto en casos de comando de referencia constante. En (Siciliano and Book, 1988; Vandegrift, *et al.*, 1994) se usa linearización por retroalimentación de entrada-salida y un término de perturbación singular (Kokotovic, 1984) para estabilizar la dinámica interna en un brazo flexible de varios elementos. Finalmente, en (Lin, *et al.*, 1990) se usa la técnica de Lyapunov para estabilizar una componente del error de seguimiento, pero no el error de seguimiento en su totalidad.

Hay diferentes técnicas de control para brazos robóticos rígidos disponibles en la literatura. Estas técnicas requieren de un conocimiento exacto de los términos no lineales (torque computado), conocimiento de cotas en las incertidumbres de los parámetros (control robusto), o conocimiento de la matriz de regresión de las funciones del robot (control adaptivo) (Lewis, *et al.*, 1993a). En la práctica es muy difícil tener tal conocimiento a priori

de la dinámica del brazo, especialmente en presencia de términos de fricción que pueden no tener una forma dinámica conocida.

Para sobrellevar estas limitaciones se ha propuesto un controlador con redes neuronales en (Lewis, *et al.*, 1993b; Lewis, *et al.*, 1995). En este esquema hay un lazo de seguimiento externo PD con la red neuronal usada en un lazo interno de linearización por retroalimentación. Las reglas de entrenamiento de los pesos incluyen un término modificado (*e-modification*) (Narendra and Annaswamy, 1987) y un término de segundo orden adicional. Usando la técnica de Lyapunov se muestra que las reglas de entrenamiento garantizan el buen desempeño de seguimiento y pesos acotados incluso si no existen pesos ideales de modo que la red neuronal reconstruya exactamente la función no lineal del robot.

Yesildirek, *et al.* (1994) diseñó un controlador de seguimiento para un brazo flexible usando el método de perturbación singular además de un lazo interno de linearización por retroalimentación usando redes neuronales. Allí se define una salida modificada que corresponde a los requerimientos prácticos de seguimiento. La estructura del controlador incluye un lazo exterior de seguimiento, un lazo interior de perturbación singular para la estabilización de la dinámica rápida y un lazo interior con una red neuronal para linearizar la dinámica rígida. Aplicando la teoría de perturbación singular se muestra que después de estabilizar la dinámica rápida, la dinámica lenta puede ser controlada aplicando el mismo método usado en (Lewis, *et al.*, 1993b y Lewis, *et al.*, 1995). Esta técnica evita el requerimiento del conocimiento de los términos de fricción, gravedad y fuerzas de Coriolis/centrípetas, o cualquier matriz de regresión. En contraste con otros controladores con redes neuronales en la literatura, no hay fase de entrenamiento fuera de línea, los pesos de la red neuronal son fáciles de inicializar sin requerir el conocimiento de pesos iniciales que garanticen la estabilidad (los pesos son inicializados en cero) y el controlador garantiza que el error de seguimiento y la señal de control son acotadas.

En este trabajo se presentan algunos resultados de la implementación práctica para un brazo flexible sencillo con el controlador diseñado por Yesildirek, *et al.*, (1994). A pesar de que la dinámica del brazo flexible sencillo es lineal, un término de fricción adicional se agregó en la implementación para

mostrar la capacidad del controlador con redes neuronales de compensar las no linealidades en el modelo. Se realizó una comparación con controladores estándar PD y PID para demostrar el desempeño superior del controlador propuesto.

## 2. DINAMICA DE UN BRAZO ROBOTICO FLEXIBLE

En (Centikunt, *et al.*, 1986; De Luca and Siciliano, 1991; Hastings and Book, 1986; Hastings and Book, 1987; Lin and Lewis, 1993) se muestra que la dinámica de un robot con varios elementos flexibles puede ser representada por

$$M(q)\ddot{q} + D(q, \dot{q})\dot{q} + Kq + F(q, \dot{q}) + G(q) = B(q)u, \quad (1)$$

con

$$q = \begin{bmatrix} q_r \\ q_f \end{bmatrix}$$

donde  $q_r$  es el vector de modos rígidos (coordenadas generalizadas de unión) y  $q_f$  es el vector de modos flexibles (las amplitudes de los modos flexibles).  $M(q)$  representa la matriz de inercia,  $D(q, \dot{q})$  es la matriz de fuerzas de Coriolis y centrífugas,  $K$  es la matriz de dureza,  $F(q, \dot{q})$  es la matriz de fricción,  $G(q)$  es la matriz de gravedad,  $B(q)$  es una matriz de entrada dependiente de las condiciones de frontera seleccionadas en el método de modos asumidos, y  $u$  incluye los torques de control aplicados en cada unión.

El modelo (1) tiene las mismas propiedades de un robot de elementos rígidos estándar (Lin and Lewis, 1993). Esto es,  $M(q)$  es positiva definida y acotada por encima y por debajo,  $D(q, \dot{q})$  está acotada por  $d_b(q)\|\dot{q}\|$ , y  $D(q, \dot{q})$  puede escogerse tal que  $\dot{M}(q) - 2D(q, \dot{q})$  es antisimétrica (Lin and Lewis, 1993).

## 3. CONTROL CON REDES NEURONALES DE BRAZOS ROBOTICOS DE ELEMENTOS FLEXIBLES

### 3.1. Método de Perturbación Singular

El método de Perturbación Singular básicamente consiste en partir la dinámica de un sistema en dos partes, cada una de ellas en un escala de tiempo

separada (Kokotovic, *et al.*, 1976; Kokotovic, *et al.*, 1986; Kokotovic, 1984). En este caso la dinámica lenta corresponde a los modos rígidos  $q_r$  y la dinámica rápida corresponde a los modos flexibles  $q_f$ . Para aplicar el método de perturbación singular, (1) puede ser dividida como en (Siciliano and Book, 1988; Vandegrift, *et al.*, 1994)

$$\begin{aligned} \ddot{q}_r &= -D_r^1 \dot{q}_r - D_{rf}^1 \dot{q}_f - K_r^1 q_r - F_r^1 - G_r^1 + B_r^1 u \\ \ddot{q}_f &= -D_f^1 \dot{q}_r - D_{ff}^1 \dot{q}_f - K_f^1 q_f - F_f^1 - G_f^1 + B_f^1 u \end{aligned} \quad (2)$$

Ahora se introduce el factor de escala  $\varepsilon$  y se define

$$\varepsilon^2 \xi = q_f, \quad (3)$$

donde  $1/\varepsilon^2$  es la más pequeña dureza en  $K_{ff}^1$ .

Se define

$$\tilde{K}_{ff} \equiv \varepsilon^2 K_{ff}^1. \quad (4)$$

Entonces

$$\begin{aligned} \ddot{q}_r &= -D_r^1(\theta)\dot{q}_r - D_{rf}^1(\theta)\varepsilon^2\dot{\xi} - H_{rf}(1/\varepsilon^2)\tilde{K}_{ff}(q_r, \varepsilon^2\xi)\varepsilon^2\xi \\ &\quad - F_r^1(q_r, \varepsilon^2\xi) - G_r^1(q_r, \varepsilon^2\xi) + B_r^1(q_r, \varepsilon^2\xi)u \\ \varepsilon^2\ddot{\xi} &= -D_f^1(\theta)\dot{q}_r - D_{ff}^1(\theta)\varepsilon^2\dot{\xi} - (1/\varepsilon^2)\tilde{K}_{ff}(q_r, \varepsilon^2\xi)\varepsilon^2\xi \\ &\quad - F_f^1(q_r, \varepsilon^2\xi) - G_f^1(q_r, \varepsilon^2\xi) + B_f^1(q_r, \varepsilon^2\xi)u \end{aligned} \quad (5)$$

donde  $\theta = (q_r, \dot{q}_r, \varepsilon^2\xi, \varepsilon^2\dot{\xi})$ .

Aquí es considerado el caso en que la dureza de los elementos es suficientemente grande de modo que  $\varepsilon$  es suficientemente pequeño. El objetivo del control es que  $q_r(t)$  siga a  $q_d(t)$ , una trayectoria predeterminada. Para este propósito definase el control

$$u = \bar{u} + u_F \quad (6)$$

donde  $\bar{u}$  es la componente lenta y  $u_F$  es la componente rápida.

Para obtener las ecuaciones de la dinámica lenta, sea  $\varepsilon = 0$  en (5) para obtener

$$\ddot{q}_r = -\bar{D}_r^1 \dot{q}_r - \bar{H}_{rf} \tilde{K}_{ff} \bar{\xi} - \bar{F}_r^1 - \bar{G}_r^1 + \bar{B}_r^1 \bar{u} \quad (7)$$

y la ecuación algebraica de la parte lenta

$$0 = -\bar{D}_f^1 \dot{q}_r - \bar{H}_{ff} \tilde{K}_{ff} \bar{\xi} - \bar{F}_f^1 - \bar{G}_f^1 + \bar{B}_f^1 \bar{u}, \quad (8)$$

la cual es resuelta para las variables lentas

$$\bar{\xi} = \bar{K}_{ff}^{-1} \bar{H}_{ff}^{-1} (-\bar{D}_{ff}^1 \dot{\bar{q}}_r - \bar{F}_f^1 - \bar{G}_f^1 + \bar{B}_f^1 \bar{u}). \quad (9)$$

Sustituyendo (9) en (7), se obtiene

$$\ddot{\bar{q}}_r = \bar{M}_r^{-1} (-\bar{D}_r \dot{\bar{q}}_r - \bar{F}_r - \bar{G}_r + \bar{u}). \quad (10)$$

Para el subsistema rápido se definen los estados

$$\begin{aligned} \zeta_1 &\equiv \xi - \bar{\xi} \\ \zeta_2 &\equiv \varepsilon \dot{\xi} \end{aligned} \quad (11)$$

con una escala de tiempo  $\tau = t / \varepsilon$  resultando en

$$\begin{aligned} \frac{d\zeta_1}{d\tau} &= \zeta_2 \\ \frac{d\zeta_2}{d\tau} &= -\bar{D}_{ff}^1 \dot{\bar{q}}_r - D_{ff}^1 \varepsilon \zeta_2 - \bar{H}_{ff} \bar{K}_{ff} (\zeta_1 + \bar{\xi}) - \bar{F}_f^1 - \bar{G}_f^1 + \bar{B}_f^1 (\bar{u} + u_r) \end{aligned} \quad (12)$$

puesto que  $d\varepsilon/d\tau \approx 0$ . Haciendo  $\varepsilon = 0$  y sustituyendo, de (9) se encuentra la dinámica rápida que es

$$\frac{d}{d\tau} \begin{bmatrix} \zeta_1 \\ \zeta_2 \end{bmatrix} = \begin{bmatrix} 0 & I \\ -\bar{H}_{ff} \bar{K}_{ff} & 0 \end{bmatrix} \begin{bmatrix} \zeta_1 \\ \zeta_2 \end{bmatrix} + \begin{bmatrix} 0 \\ \bar{B}_f^1 \end{bmatrix} u_r, \quad (13)$$

$$\frac{d\zeta}{d\tau} = A_r \zeta + B_r u_r, \quad (14)$$

$$\text{con } \zeta = \begin{bmatrix} \zeta_1^T & \zeta_2^T \end{bmatrix}^T.$$

De acuerdo al teorema de Tikhonov (Kokotovic, 1984; Kokotovic, et al., 1986) el sistema original (2) puede describirse dentro de un orden  $\varepsilon$  usando (10) y (14) con

$$\begin{aligned} q_r &= \bar{q}_r + O(\varepsilon) \\ q_f &= \varepsilon^2 (\bar{\xi} + \zeta_1) + O(\varepsilon) \end{aligned} \quad (15)$$

con  $O(\varepsilon)$  denotando términos de orden  $\varepsilon$ .

Ahora se define la salida

$$y = \begin{bmatrix} \bar{q}_r \\ \dot{\bar{q}}_r \end{bmatrix}, \quad (16)$$

la cual corresponde a la parte lenta de las variables de los modos rígidos. Se asume que  $(A_r, B_r)$  es estabilizable, los parámetros tienen incertidumbres y perturbaciones acotadas (variables del subsistema lento), y las variables lentas cambian suavemente en

el tiempo. La asunción de estabilizabilidad de  $(A_r, B_r)$  se satisface en sistemas prácticos y es mucho más suave que el requerimiento de una dinámica natural estable. Además, la definición en (16) corresponde a objetivos prácticos de seguimiento en contraste con las salidas "reflejadas" definidas en (Madhavan and Singh, 1991; Wang and Vidyagar, 1991). Bajo estas asunciones, un control que estabiliza la parte rápida  $u_r(t)$  puede ser diseñado fácilmente usando técnicas lineales (e.g. método de  $H_\infty$ ) de modo que

$$u_r = - \begin{bmatrix} K_{pF} & K_{dF} \end{bmatrix} \begin{bmatrix} \zeta_1 \\ \zeta_2 \end{bmatrix} = - \frac{K_{pF}}{\varepsilon^2} q_f - \frac{K_{dF}}{\varepsilon} \dot{q}_f + K_{pF} \bar{\xi} \quad (17)$$

estabiliza a (14), con  $\bar{\xi}$  dado por (9).

### 3.2. Control con Redes Neuronales de la Dinámica Rígida

La dinámica lenta dada por (10) puede reescribirse como

$$\bar{M}_r \ddot{\bar{q}}_r + \bar{D}_r \dot{\bar{q}}_r + \bar{F}_r + \bar{G}_r = \bar{u} \quad (18)$$

que es exactamente la forma de Lagrange de un brazo robótico de  $n$  elementos, satisfaciendo las propiedades estándar de un robot. Para esta parte puede diseñarse un controlador con redes neuronales (Lewis, et al., 1993b; Lewis, et al., 1995). Nótese que  $\bar{M}_r - 2\bar{D}_r$  es antisimétrica.

Dada una trayectoria deseada  $q_d(t)$  para  $\bar{q}_r$ , el error de seguimiento es

$$e = q_d - \bar{q}_r \quad (19)$$

El error de seguimiento filtrado se define como

$$r = \dot{e} + \Lambda e, \quad (20)$$

donde  $\Lambda = \Lambda^T > 0$ . Usando (20), la dinámica del brazo puede reescribirse en términos del error de seguimiento filtrado como

$$\bar{M}_r \ddot{r} = -\bar{D}_r r - \bar{u} + h(x) \quad (21)$$

donde la función no lineal del robot es

$$\begin{aligned} h(x) &= \bar{M}_r(\bar{q})(\ddot{q}_d + \Lambda \dot{e}) + \bar{D}_r(\bar{q}, \dot{\bar{q}})(\dot{q}_d + \Lambda e) \\ &\quad + \bar{F}_r(\bar{q}) + \bar{G}_r(\bar{q}) \end{aligned} \quad (22)$$

con  $x = [e^T \quad \dot{e}^T \quad \ddot{e}^T \quad \ddot{q}_d^T \quad \ddot{q}_d^T]^T$ . Se asume que  $h(x)$  es desconocida.

Una red neuronal puede usarse para estimar  $h(x)$  basados en la propiedad de aproximación universal de las redes neuronales la cual se establece en el siguiente teorema.

**Teorema: Propiedad de Aproximación Universal de las Redes Neuronales**

Sea  $f(x): \mathcal{R}^n \rightarrow \mathcal{R}^m$  una función suave. Entonces; dado un conjunto compacto  $S \in \mathcal{R}^n$  y un número positivo  $\varepsilon_N$ , existe una red neuronal de dos capas tal que

$$f(x) = W^T \sigma(V^T x) + \varepsilon, \quad (23)$$

con  $\|\varepsilon\| < \varepsilon_N$  para todo  $x \in S$ , para algún número (suficientemente grande)  $L$  de neuronas en la capa oculta.  $\varepsilon$  es generalmente una función de  $x$  y es llamado error de reconstrucción de la red neuronal.  $\varepsilon$  decrece cuando  $L$  se incrementa.  $\square$

El valor estimado de  $h(x)$  es dado por

$$\hat{h}(x) = \hat{W}^T \sigma(\hat{V}^T x), \quad (24)$$

Sea

$$Z \equiv \begin{bmatrix} V & 0 \\ 0 & W \end{bmatrix} \quad (25)$$

la matriz de pesos ideales, la cual es desconocida.

El error de aproximación funcional de la red neuronal es

$$\tilde{h}(x) = h(x) - \hat{h}(x) \quad (26)$$

el cual puede ser escrito usando una expansión de Taylor, asumiendo funciones de activación suaves, como

$$\tilde{h}(x) = \tilde{W}^T (\hat{\sigma} - \hat{\sigma}' \hat{V}_h^T x) + \hat{W}^T \hat{\sigma}' \hat{V}^T x + w \quad (27)$$

donde

$$\hat{\sigma} \equiv \sigma(\hat{V}^T x), \quad \hat{\sigma}' \equiv \left. \frac{\partial \sigma(z)}{\partial z} \right|_{z=\hat{z}}, \quad (28)$$

y el término de error adicional

$$w(t) = \tilde{W}^T \hat{\sigma}' V^T x + W^T O(\tilde{V}^T x)^2 + \varepsilon_{l_h}(x) \quad (29)$$

está acotado de acuerdo con

$$w(t) \leq C_0 + C_1 \|\tilde{Z}\| + C_2 \|x\| \|Z\|. \quad (30)$$

El jacobiano  $\hat{\sigma}'$  es una función fácilmente computable de  $\hat{V}^T x$ .

Se asume que los pesos ideales de la red neuronal están acotados de modo que

$$\|Z\| \leq Z_m \quad (31)$$

con  $Z_m$  siendo una cota conocida, y que la trayectoria deseada está acotada de acuerdo con

$$\begin{bmatrix} q_d \\ \dot{q}_d \\ \ddot{q}_d \end{bmatrix} \leq Q \quad (32)$$

siendo  $Q$  una cota conocida.

**Definición**

La solución a

$$\dot{x} = f(x, u, t), \quad y = g(x, t)$$

es globalmente uniformemente ultimamente acotada (GUUB) si para todo  $x(t_0)$  existe un  $\varepsilon > 0$  y un número  $T(\varepsilon, x_0)$  tal que  $\|x(t)\| < \varepsilon$  para todo  $t \geq t_0 + T$ .  $\square$

Bajo todas las asunciones mencionadas arriba, un controlador con redes neuronales está definido por el siguiente teorema (Yesildirek, et al., 1994).

**Teorema**

Sea la trayectoria deseada y los pesos ideales desconocidos acotados de acuerdo con las asunciones dadas. Sea la entrada de control para (18) definida por

$$\bar{u} = \hat{h} + K_v r - v, \quad \text{for } K_v = K_v^T > 0 \quad (33)$$

con el término para robustez

$$v(t) = -K_z (\|\hat{Z}\| + Z_m) r \quad (34)$$

y ganancia  $K_z > C_2$ .

Sean los pesos de la red neuronal sintonizados por

$$\begin{aligned}\dot{\hat{W}} &= M(\hat{\sigma} - \hat{\sigma}'\hat{V}^T x)r^T - \kappa\|r\|M\hat{W} \\ \dot{\hat{V}} &= Nxr^T\hat{W}^T\hat{\sigma}' - \kappa\|r\|N\hat{V}\end{aligned}\quad (35)$$

con matrices constantes  $M = M^T > 0$ ,  $N = N^T > 0$  cualesquiera, y un parámetro de diseño escalar  $\kappa > 0$ .

Entonces el error de seguimiento filtrado  $r(t)$  y los errores de los pesos de la red neuronal  $\tilde{V}$ ,  $\tilde{W}$  son GUUB. Más aún, el error de seguimiento puede mantenerse tan pequeño como se quiera incrementando las ganancias  $K_v$ .  $\square$

La prueba de este teorema usa la teoría de Lyapunov y es dada en (Yesildirek, *et al.*, 1994), donde se dan cotas explícitas para  $\|r\|$  y  $\|\hat{Z}\|$ . Nótese que las reglas de entrenamiento en (35) incluyen los términos estándar de propagación hacia atrás mas un término adicional (*e-modification*) (Narendra and Annaswamy, 1987) y un término de corrección de segundo orden. Además, los pesos de la red neuronal pueden ser inicializados fácilmente en cero puesto que el control PD estabiliza al sistema mientras la red neuronal está aprendiendo. El controlador con redes neuronales está diseñado para controlar el robot mientras aprende para mejorar su desempeño, así no se requiere de entrenamiento fuera de línea.

La estructura general del controlador definido en las secciones 3.1 y 3.2 es mostrado en la Fig. 1.

### 3.3. Simulación

La simulación del controlador con redes neuronales fue realizada para un elemento flexible sencillo. El modelo del elemento flexible incluyó tres modos flexibles aunque sólo se compensaron los dos primeros modos (esto fue para corroborar que el controlador funciona bien aunque sólo compense un número finito de modos). El modelo se obtuvo como se describe en (Hastings and Book, 1986), usando los parámetros de la cama de prueba con un elemento flexible del Automation and Robotics Research Institute (ARRI). Las frecuencias modales de los tres primeros modos para este elemento flexible son 1.6Hz, 10.0Hz y 28.1Hz.

El controlador usó los siguientes parámetros:

$$\begin{aligned}K_v &= 36 \\ \Lambda &= \frac{200}{36} \\ K_z &= 0.2 \\ Z_m &= 50 \\ \frac{K_{pf}}{\varepsilon^2} &= [-8 \quad 10] \\ \frac{K_{df}}{\varepsilon} &= [0 \quad 0]\end{aligned}$$

El controlador con redes neuronales incluyó diez neuronas en la capa oculta y usó los siguientes parámetros

$$\begin{aligned}F &= 20 \\ G &= 20 \\ \kappa &= 0.000001\end{aligned}$$

Las funciones de activación en la capa oculta fueron seleccionadas como las funciones sigmoideas

$$\sigma_k(z) = \frac{1}{1 + e^{-k\alpha z}} \quad \text{for } k = 1, 2, \dots, 10, \alpha = 1$$

Se observó en la práctica que con las funciones de activación definidas de este modo, la red neuronal aprendió más rápido y fue capaz de reducir más el error de seguimiento.

Los resultados de la simulación se grafican en la Fig. 2. Nótese que después de cierto tiempo la red neuronal aprende el modelo del brazo reduciendo el error de seguimiento casi a cero. Usando valores más grandes en las tasas de aprendizaje  $F$  y  $G$  mejoró el desempeño de seguimiento (aprendizaje más rápido y menor error de seguimiento), pero produjo una respuesta transitoria peor (más oscilatoria) y mayor excitación de los modos flexibles (se incrementó la magnitud de  $q_{f1}$ ).

## 4. IMPLEMENTACION DEL CONTROLADOR EN LA CAMA DE PRUEBA PARA EL ELEMENTO FLEXIBLE

El controlador con redes neuronales discutido en la sección 3 fue implementado en una cama de prueba con un brazo flexible sencillo en el ARRI y algunos de los resultados obtenidos se presentan aquí.

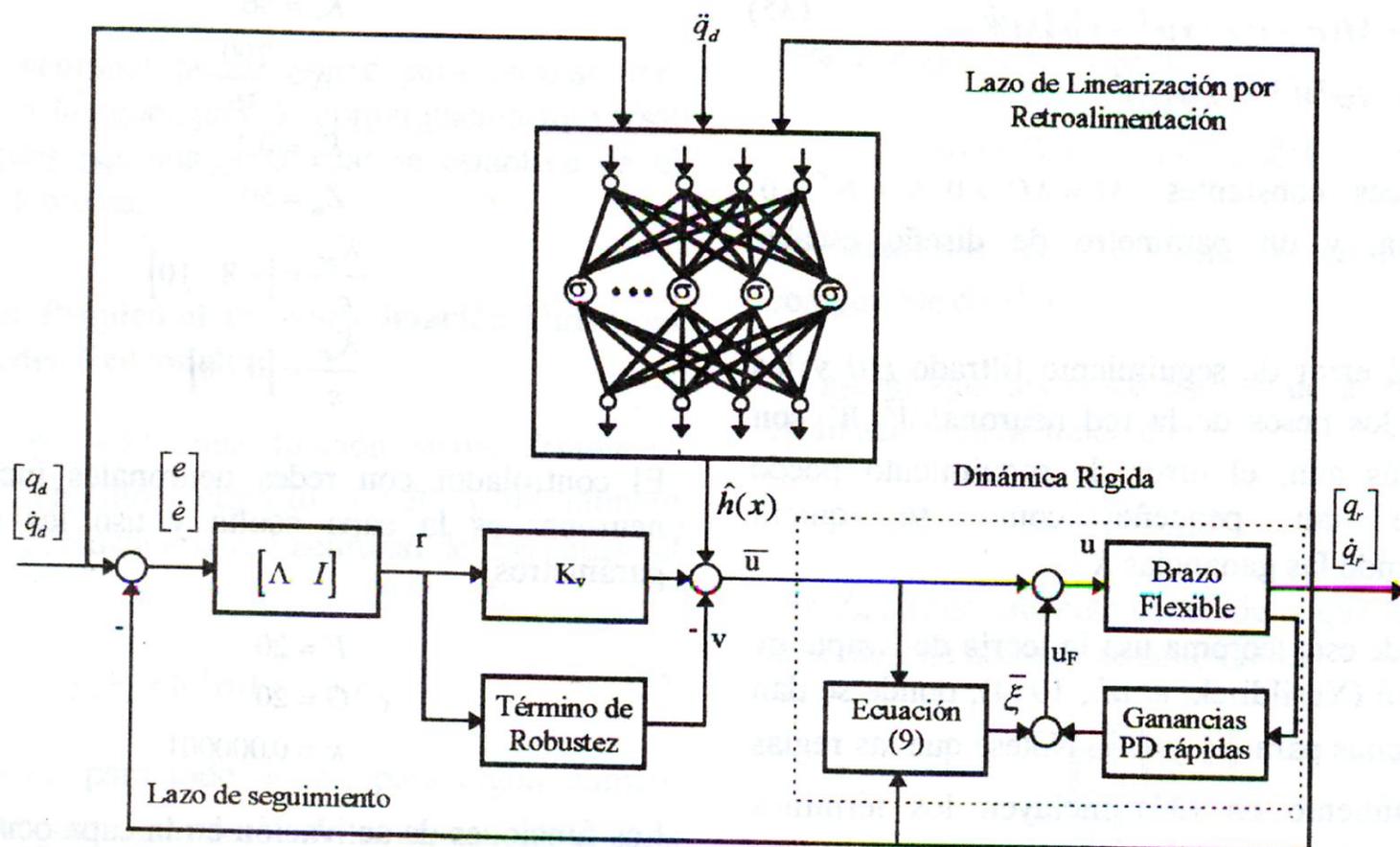


Fig. 1. Estructura general del controlador con redes neuronales para un brazo robótico flexible sencillo.

#### 4.1. Descripción de la implementación

La estructura actual del controlador implementado en la cama de prueba del ARRI es mostrada en la Fig. 3. Una lista de las principales características de la implementación práctica es:

- Sólo se consideraron los dos primeros modos flexibles.
- El término de robustez  $v$  no fue incluido.  $K_v$  fue suficientemente grande para evitar la necesidad de  $v$ . El término  $\bar{\xi}$  no fue incluido puesto que el modelo real del elemento flexible es desconocido. Como lo muestra (9), la implementación de  $\bar{\xi}$  requeriría el conocimiento exacto de las matrices del modelo.
- Aunque la dinámica del brazo flexible con un grado de libertad es lineal, se agregó un término de fricción adicional para chequear la capacidad del controlador para compensar las no linealidades del modelo.

- La red neuronal se compuso de diez neuronas en la capa oculta, con cinco entradas ( $x = [e \ \dot{e} \ \ddot{q}_d \ \dot{q}_d \ \ddot{q}_d]^T$ ) y una salida  $\hat{h}(x)$ .
- El controlador definido por (6), (17), (33) y (35) fue discretizado con un periodo de muestreo de 5ms. En el proceso de discretización las ecuaciones diferenciales en (35) fueron resueltas en línea usando integración trapezoidal.

Un diagrama de bloques describiendo la implementación práctica del controlador es mostrada en la Fig. 4. El hardware incluye las tarjetas de interfase y componentes externos necesarios para la medida de la posición angular del brazo  $q_r$  y los modos flexibles  $q_{f1}$  y  $q_{f2}$  (codificador óptico, strain gauges, acondicionadores de señal, y conversores A/D). Valores estimados de  $\dot{q}_r$ ,  $\dot{q}_{f1}$ , and  $\dot{q}_{f2}$  fueron calculados basados en muestras consecutivas de  $q_r$ ,  $q_{f1}$  y  $q_{f2}$  respectivamente. Además hay un conversor D/A conectado al servoamplificador que maneja el servomotor del brazo.

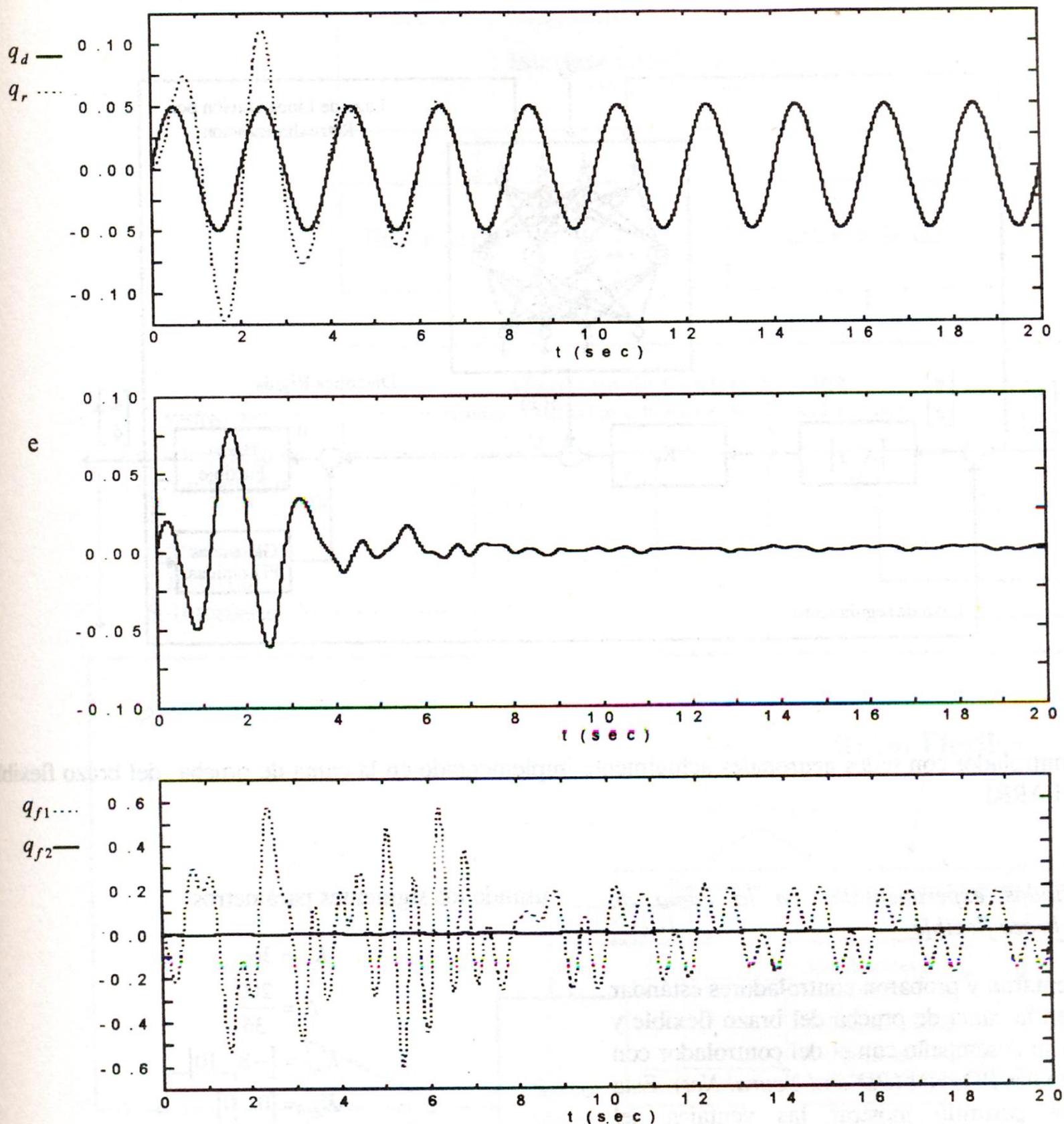


Fig. 2. Simulación del controlador con redes neuronales para un elemento flexible sencillo.

El software fue implementado en LabView y C. Las rutinas que ejecutan la acción de control en tiempo real fueron implementadas en C. La ejecución de estas rutinas es disparada periódicamente por las rutinas del timer del computador. Las rutinas de control muestrean las señales externas y usan los parámetros definidos en el buffer de parámetros para calcular la señal de control  $u$ . Algunas de las señales son almacenadas en el buffer de señales permitiendo que los VI's del LavView las monitoreen.

Los VI's del LabView trabajan como un interfase gráfico de usuario que permite iniciar el controlador, cambiar el modo de operación, definir las señales de

referencia, cambiar los parámetros del controlador, y monitorear las señales a través de cartas y gráficas. Estos VI's están ligados a las rutinas externas en C las cuales corren en el fondo en tiempo real. La comunicación entre las rutinas externas en C y los VI's en LabView se logra a través de algunos VI's que leen desde los buffers y escriben en ellos usando CIN's (Code Interface Nodes).

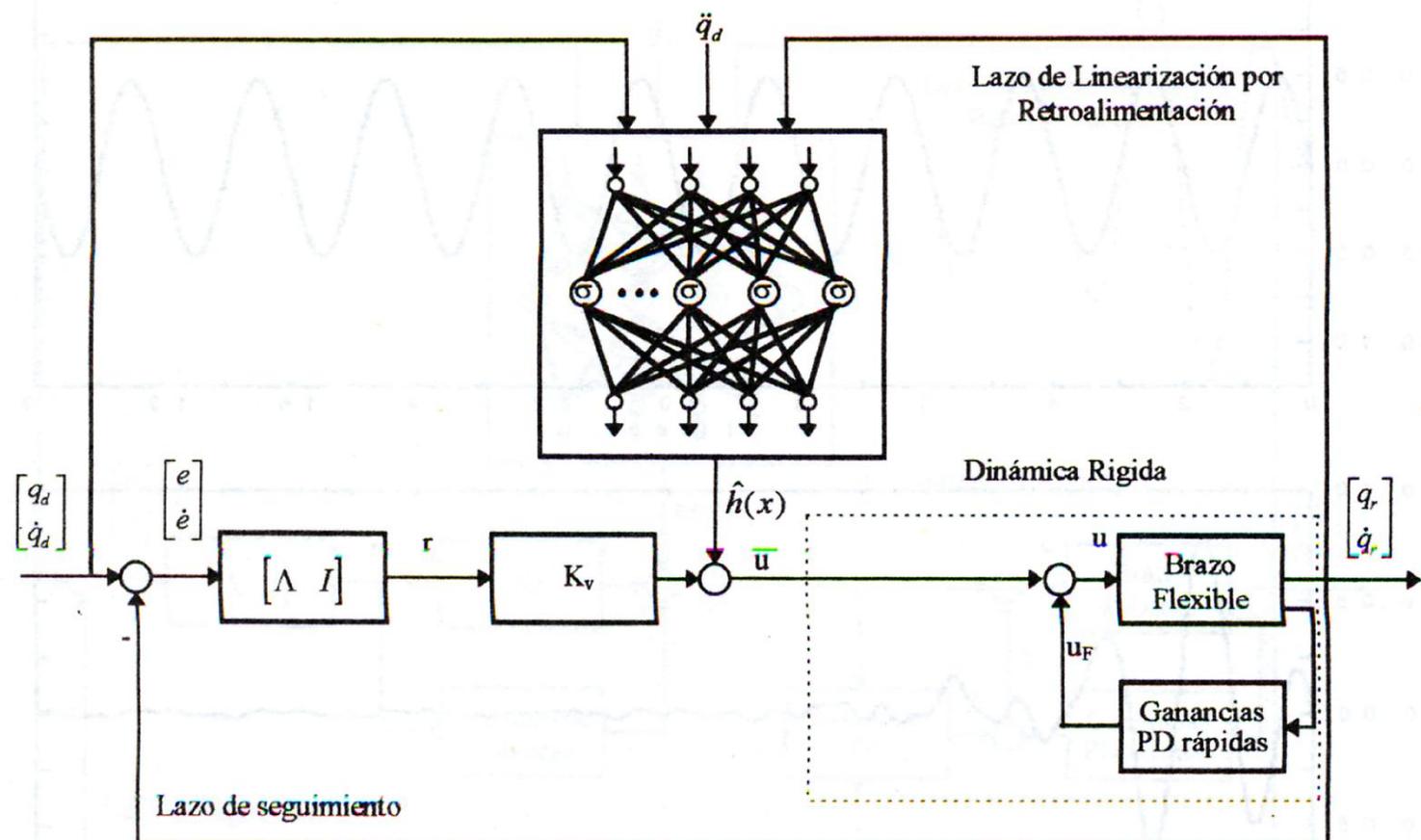


Fig. 3. Controlador con redes neuronales actualmente implementado en la cama de prueba del brazo flexible del ARRI.

#### 4.2. Resultados experimentales en la cama de prueba del brazo flexible

Se implementaron y probaron controladores estándar PD y PID en la cama de prueba del brazo flexible y se comparó su desempeño con el del controlador con redes neuronales PD+NN (*PD and Neural Net*). Esta comparación permitió mostrar las ventajas del controlador propuesto sobre los controladores estándar.

##### Control PD

Se implementó un controlador PD usando la ley de control

$$u = \bar{u} + u_F$$

con

$$\bar{u} = K_v r = K_v (\dot{e} + \Lambda e)$$

$$u_F = - \begin{bmatrix} K_{pf} & K_{df} \end{bmatrix} \begin{bmatrix} q_f \\ \dot{q}_f \end{bmatrix}$$

usando los siguientes parámetros

$$K_v = 36$$

$$\Lambda = \frac{200}{36}$$

$$K_{pf} = \begin{bmatrix} -8 & 10 \end{bmatrix}$$

$$K_{df} = \begin{bmatrix} 0 & 0 \end{bmatrix}$$

y la señal de referencia

$$q_d = 0.05 \sin(2\pi ft)$$

con frecuencia  $f=0.5\text{Hz}$ .

El desempeño del controlador PD de seguimiento (sin la red neuronal) se ilustra en la Fig. 5(a) sin el término de fricción adicional, y en la Fig. 5(b) con el término de fricción adicional. Nótese que el error de seguimiento es muy grande, su magnitud es comparable a la de la señal de referencia. Aunque la magnitud del error decrece al incrementar las ganancias del controlador, el error de seguimiento no es eliminado. Estas características se conservan en presencia de un término de fricción adicional.

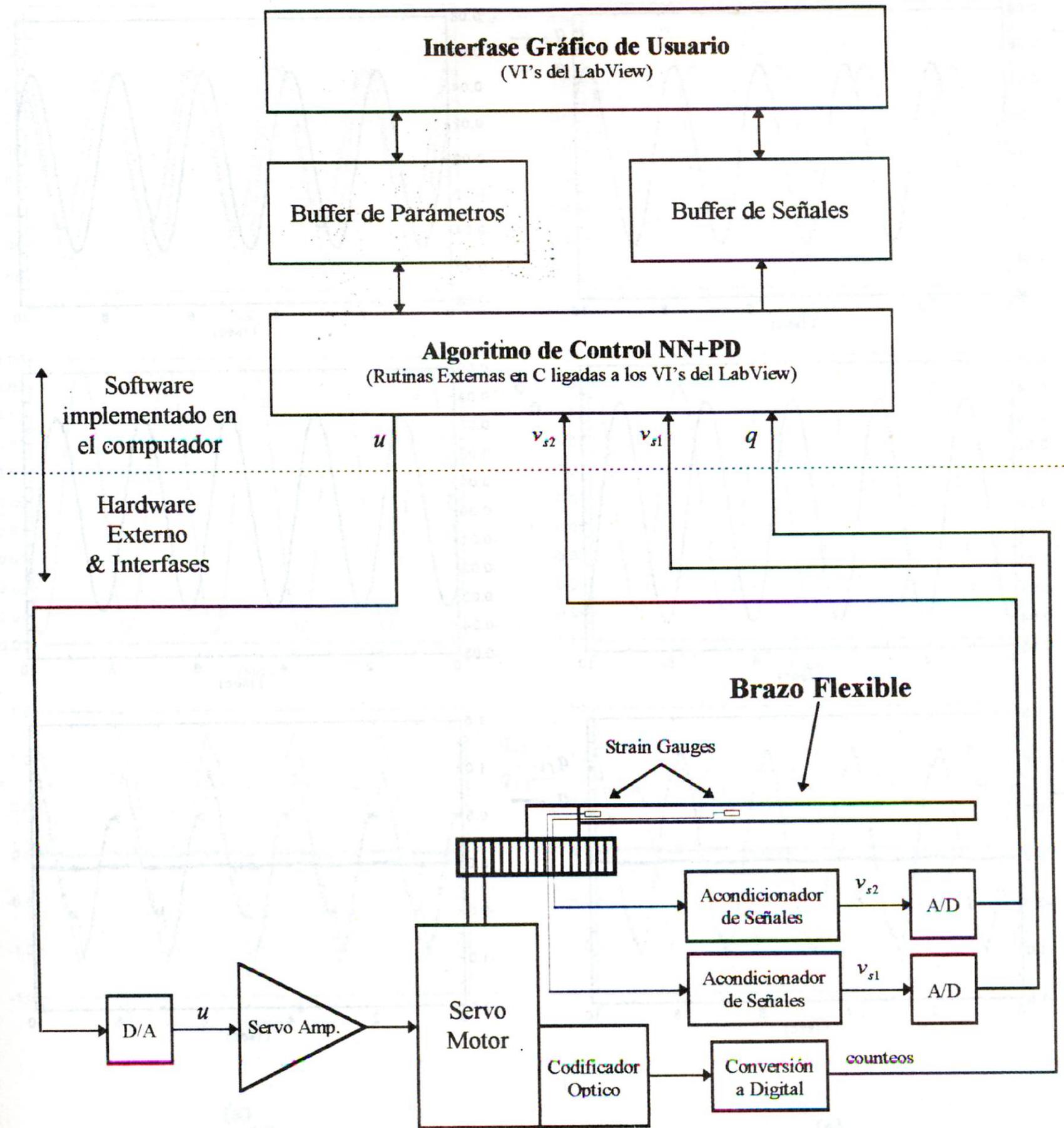


Fig. 4. Diagrama de bloques de la implementación del controlador con redes neuronales para la cama de prueba del brazo flexible del ARRI.

### Control PID

Se implementó un controlador PID usando la ley de control

$$u = \bar{u} + u_F$$

con

$$\bar{u} = K_v r + K_i \int e dt = K_v (\dot{e} + \Lambda e) + K_i \int e dt$$

$$u_F = - \begin{bmatrix} K_{pF} & K_{dF} \end{bmatrix} \begin{bmatrix} q_f \\ \dot{q}_f \end{bmatrix}$$

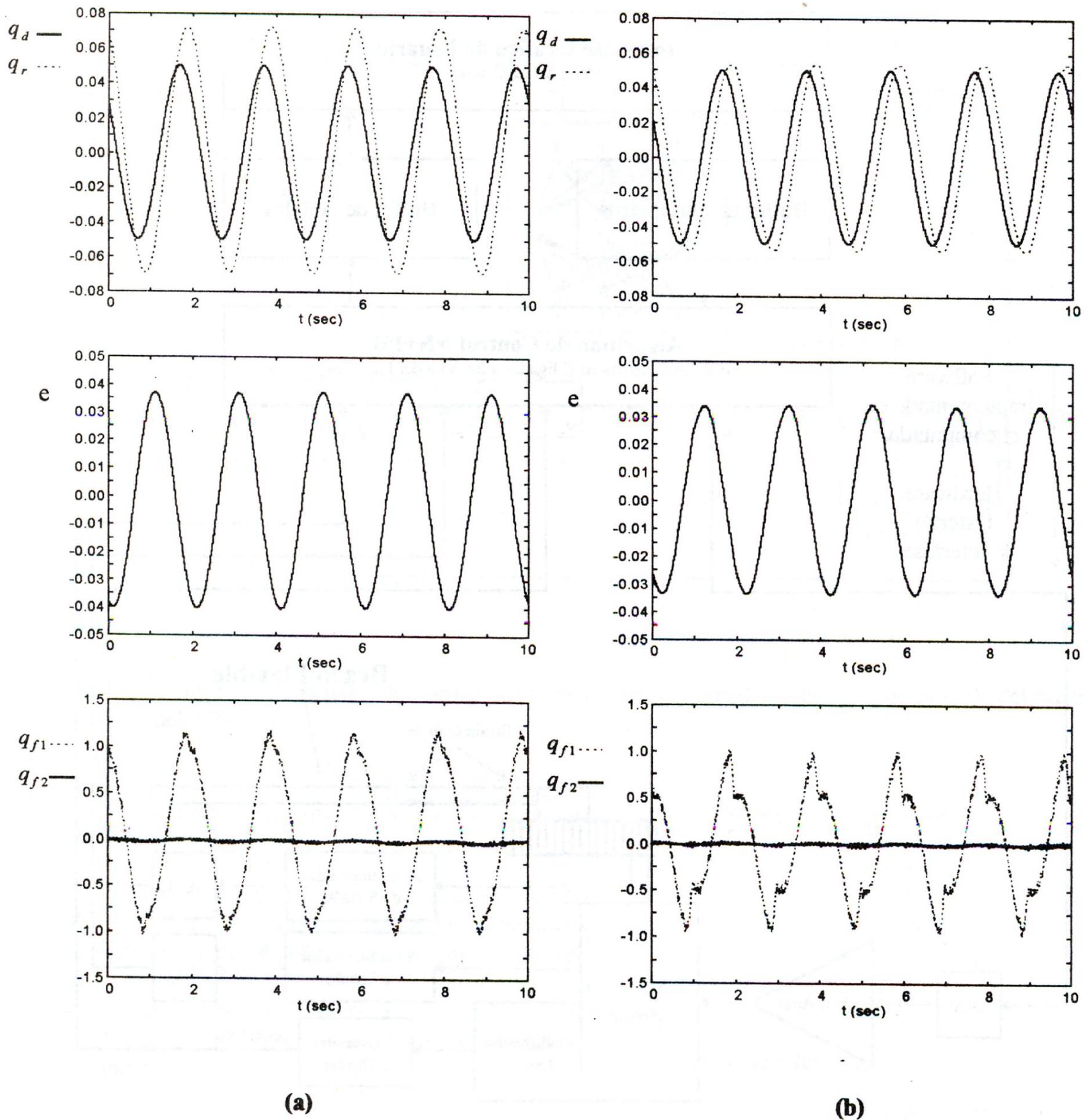


Fig. 5. Desempeño del controlador PD. (a) Sin fricción adicional. (b) Con fricción adicional.

usando los siguientes parámetros

$$\begin{aligned}
 K_v &= 36 \\
 \Lambda &= \frac{200}{36} \\
 K_i &= 100 \\
 K_{pf} &= [-8 \quad 10] \\
 K_{df} &= [0 \quad 0]
 \end{aligned}$$

y la señal de referencia

$$q_d = 0.05 \sin(2\pi ft)$$

con frecuencia  $f=0.5\text{Hz}$ .

El desempeño del controlador PID es ilustrado en la Fig. 6(a) sin el término de fricción adicional, y en la Fig. 6(b) con el término de fricción adicional. Supuestamente la parte integral del control PID

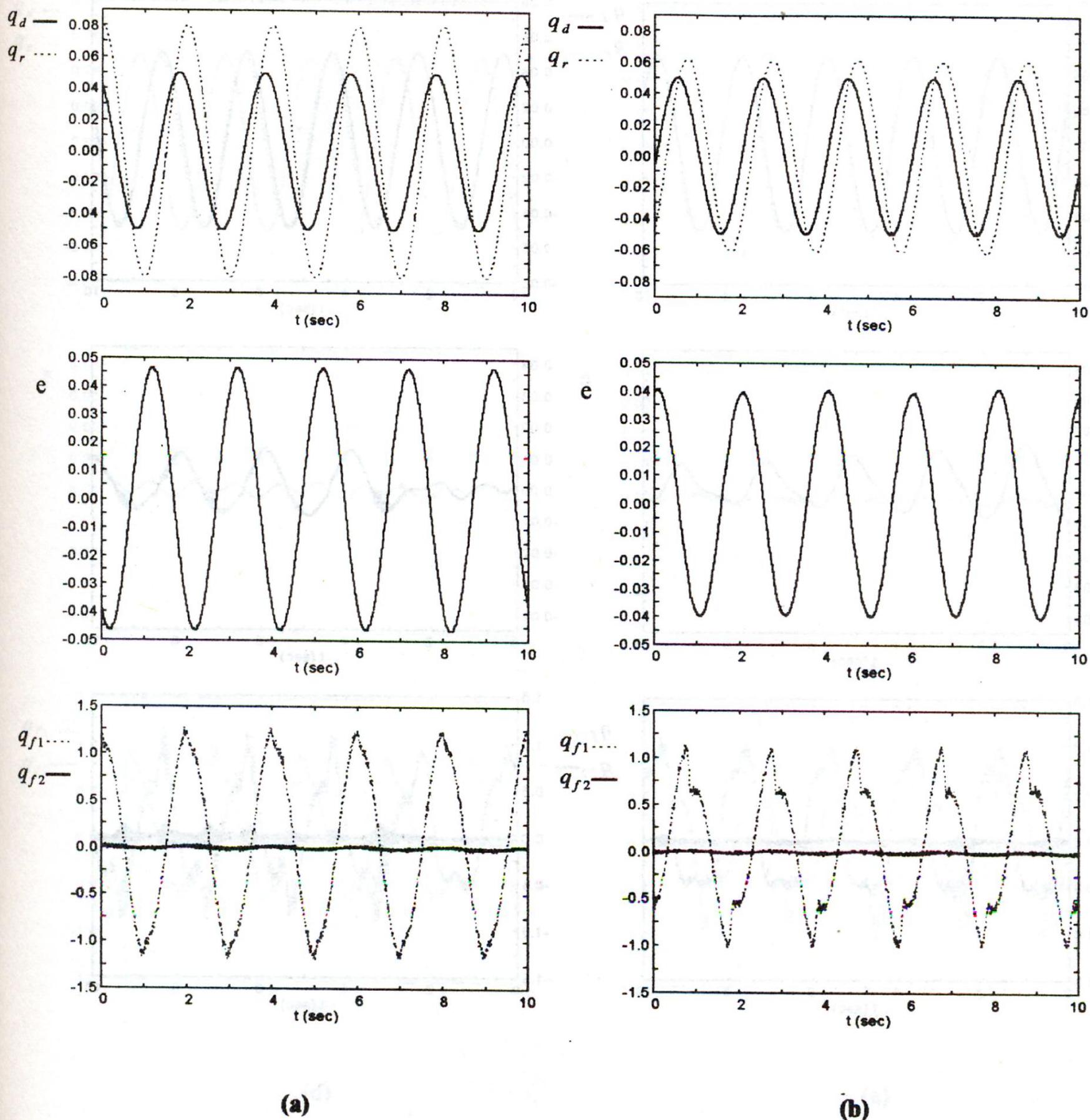


Fig. 6. Desempeño del controlador PID. (a) Sin fricción adicional. (b) Con fricción adicional.

debería eliminar el error de estado estacionario, pero esto sólo funciona para trayectorias deseadas constantes. En este caso, Con una trayectoria deseada variable, el error de seguimiento es incluso peor cuando la parte integral es introducida (note que el error de seguimiento es mayor que con el control PD). Como en el caso del controlador PD, el controlador PID no es capaz de compensar el término de fricción adicional.

#### Control NN+PD

El controlador con redes neuronales fue implementado como se describe en la sección 4.1 usando los mismos parámetros de la simulación en la sección 3.3, excepto que en este caso

$$F = 2$$

$$G = 20$$

$$\kappa = 0.000001$$

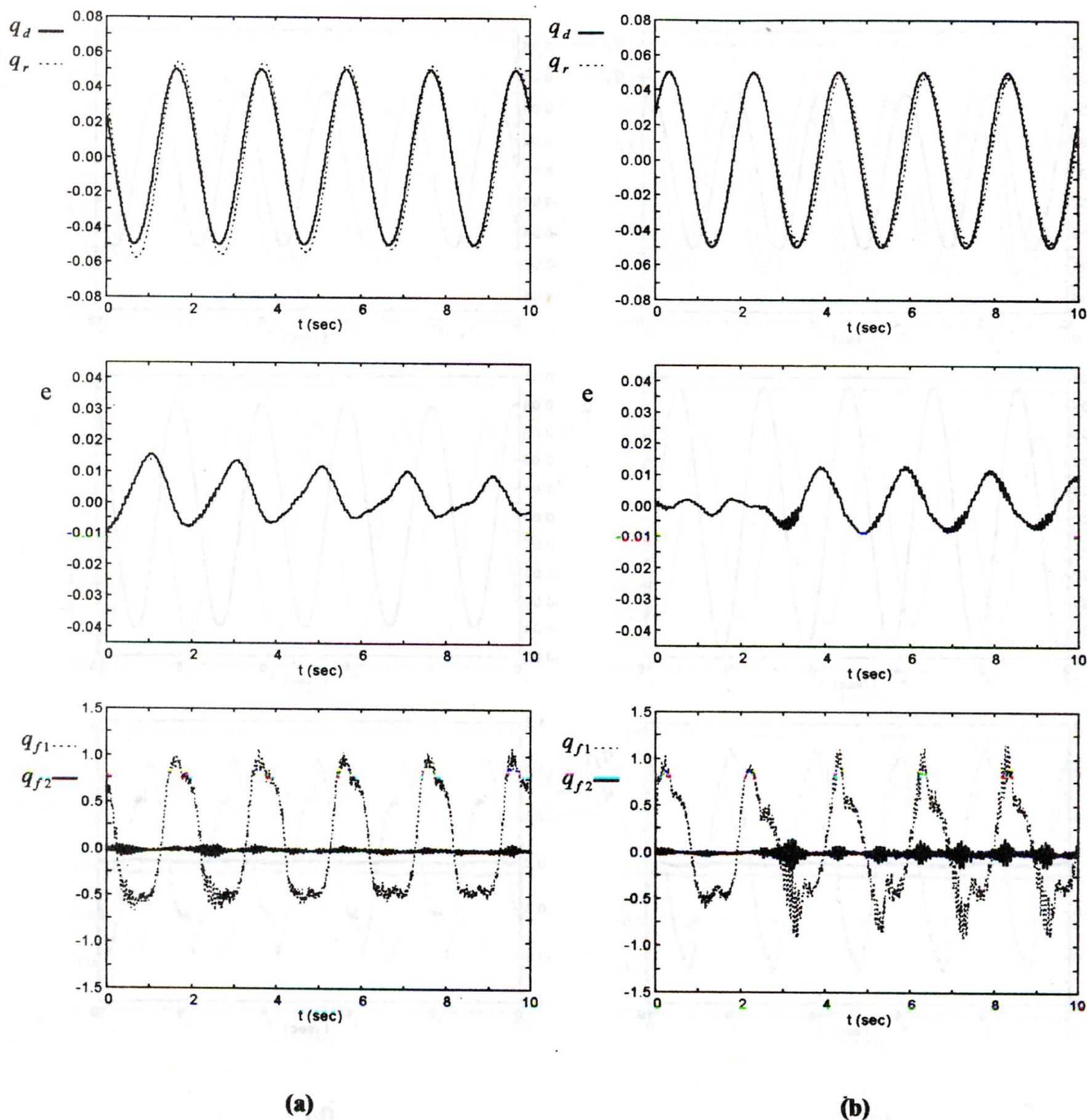


Fig. 7. Desempeño del controlador PD+NN antes de completar el aprendizaje. (a) Sin fricción adicional. (b) Con fricción adicional.

Este valor de  $F$  fue suficiente en la implementación práctica. Un valor mayor producía una respuesta muy oscilatoria.

La señal de referencia fue

$$q_d = 0.05 \sin(2\pi ft)$$

con frecuencia  $f=0.5\text{Hz}$ .

El desempeño del controlador con redes neuronales se observa en la Fig. 7 antes de que el aprendizaje es completo y en la Fig. 8 después de que el aprendizaje es completo. El entrenamiento de la red neuronal toma menos de un minuto, después del cual el error de seguimiento se reduce casi a cero. El aprendizaje es realmente activo todo el tiempo (entrenamiento en línea), pero se dice que el entrenamiento es completo en el instante en que la red neuronal ha aprendido el

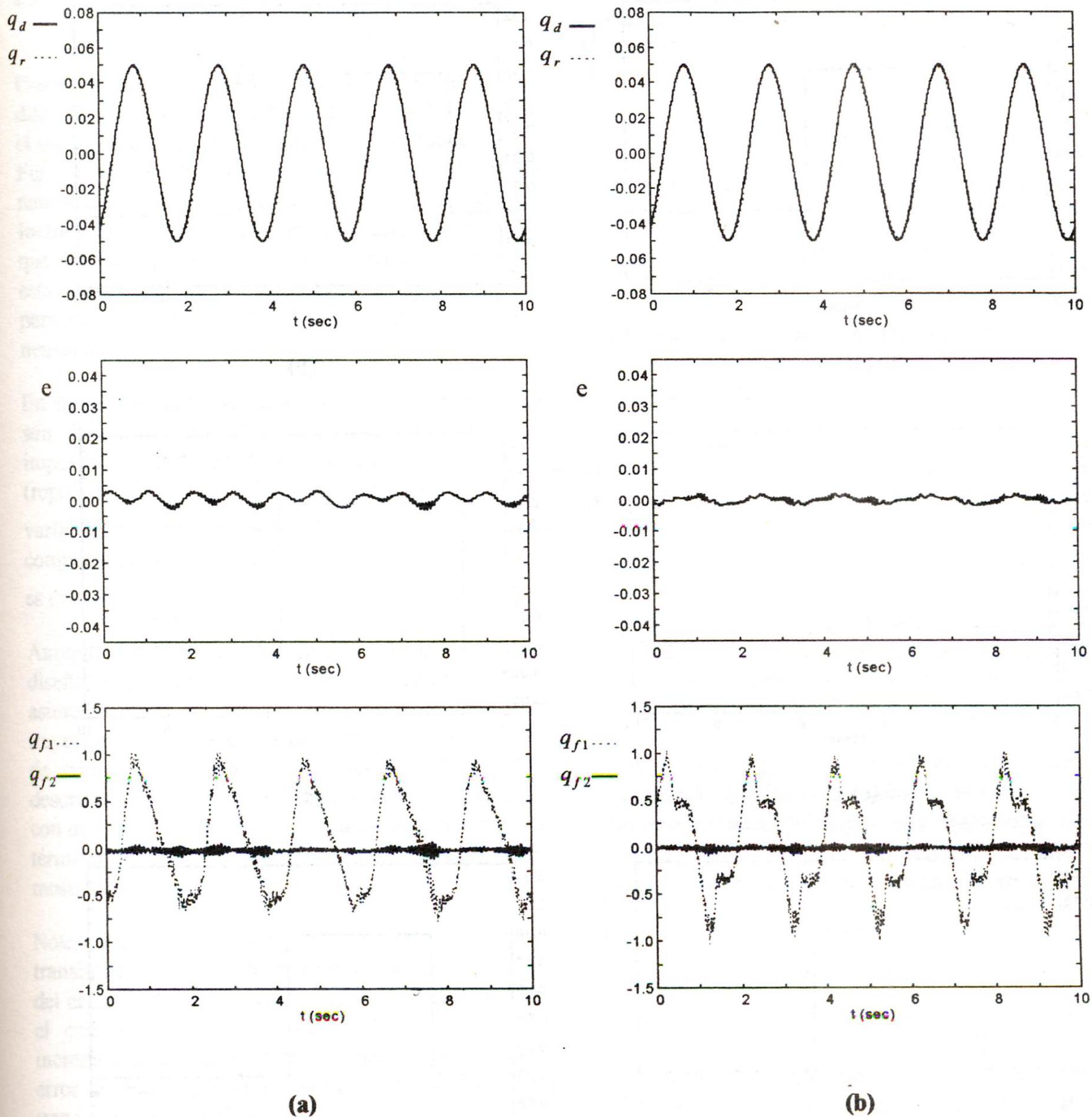


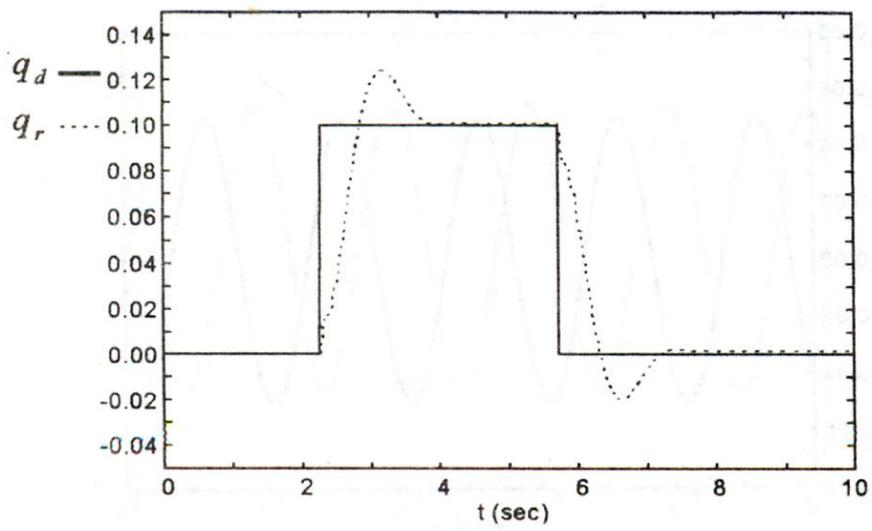
Fig. 8. Desempeño del controlador PD+NN despues de completar el aprendizaje. (a) Sin fricción adicional. (b) Con fricción adicional.

modelo del brazo bajo las condiciones actuales reduciendo el error a casi cero.

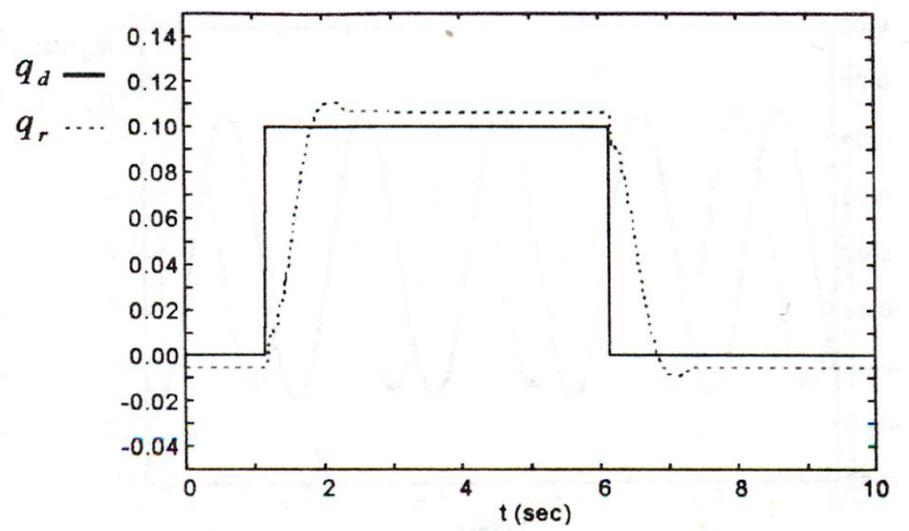
Nótese en la Fig. 8(a), sin el término de fricción adicional, y en la Fig. 8(b), con el término de fricción adicional, que el mismo controlador aprende el modelo del brazo readaptándose a cambios en él (cambios en el modelo como cambios en las características de fricción). Sin cambiar los parámetros del controlador, el controlador con redes

neuronales es capaz de llevar el error a casi cero en ambos casos.

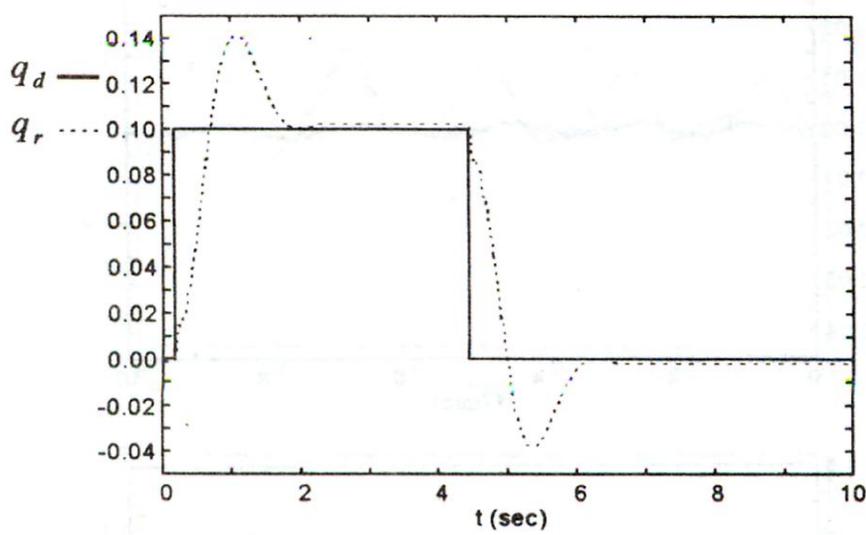
En la práctica se notó que un cambio de la señal de referencia incrementó el error de seguimiento temporalmente requiriendo una readaptación de la red neuronal. Sin embargo, después de algún tiempo, cuando la red neuronal aprendió las nuevas condiciones, fue capaz de deshacerse del error de seguimiento.



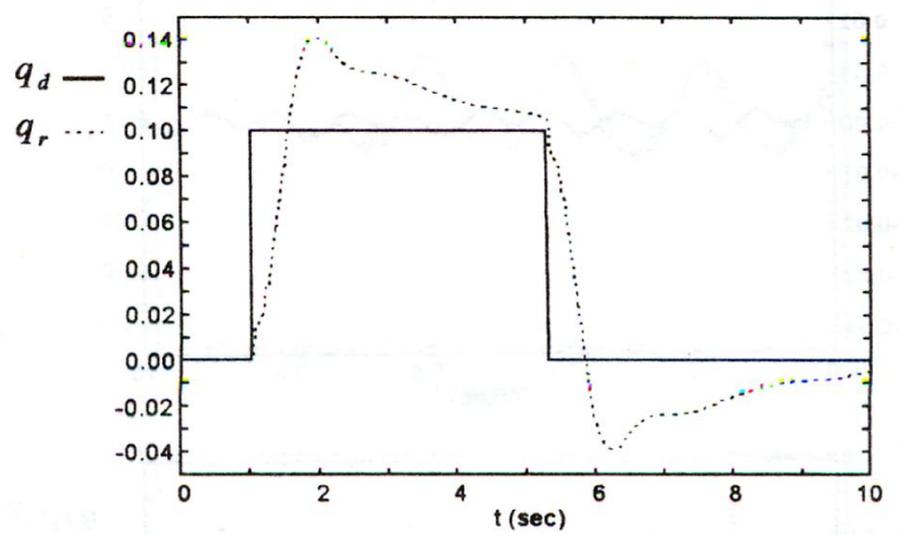
(a)



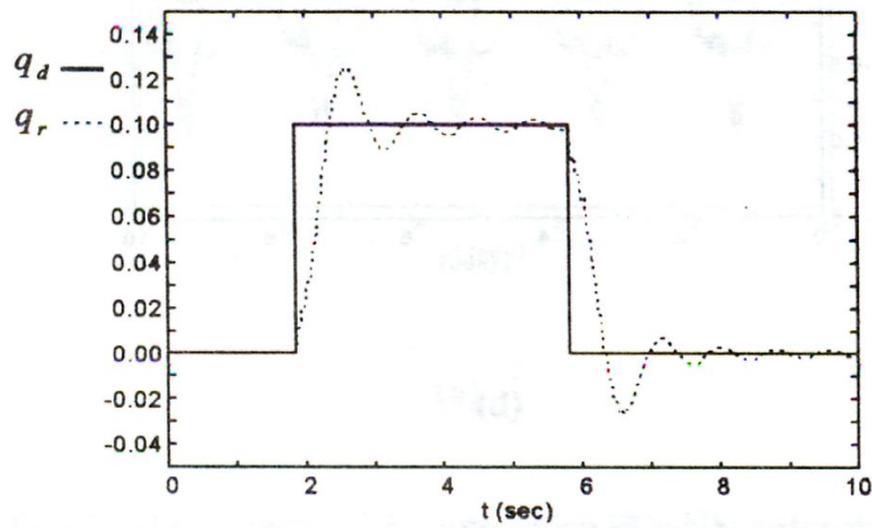
(b)



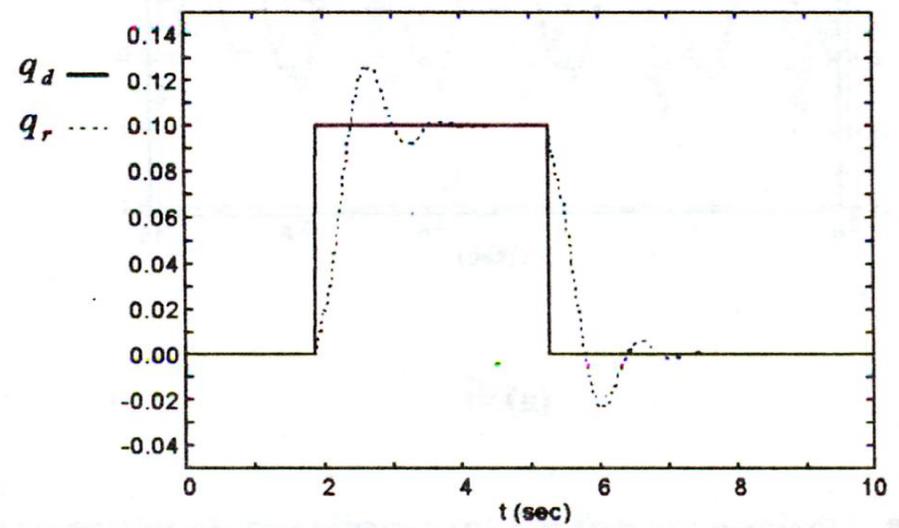
(c)



(d)



(e)



(f)

**Fig. 9.** Respuesta al escalón de los controladores. (a) Control PD. (b) Control PD con fricción adicional. (c) Control PID. (d) Control PID con fricción adicional. (e) Control PD+NN. (f) Control PD+NN con fricción adicional.

### 4.3. Comparación entre las diferentes aproximaciones

Comparando el desempeño de seguimiento de los diferentes controladores, mostrado en la Fig. 5 para el controlador PD, Fig. 6 para el controlador PID, y Fig. 7 y Fig. 8 para el controlador con redes neuronales, es clara la superioridad del último. Incluso el PID no puede ser un mejor controlador que el PID bajo una trayectoria deseada variable. En esta situación el controlador PD es mejor que el PID, pero no tan bueno como el controlador con la red neuronal.

En todos los casos se nota que los modos flexibles son bien amortiguados por el controlador. Es imposible eliminar el doblado del brazo (representado por  $q_{f1}$ ) bajo una trayectoria deseada variable porque es parte de su física, pero la compensación del modo de más alta frecuencia  $q_{f2}$  es evidente.

Aunque el controlador con redes neuronales no fue diseñado para seguir funciones escalón (ver la asunción dada por (32)), fue probado con una trayectoria deseada de tipo escalón para propósitos de comparación con los controladores PD y PID descritos arriba. Estos controladores fueron probados con una trayectoria de referencia escalón con y sin el término de fricción adicional. Los resultados son mostrados en la Fig. 9.

Nótese que el controlador PD tiene buena respuesta transitoria (Fig. 9(a)) pero no es capaz de deshacerse del error de estado estacionario, y se vuelve peor en el caso de la fricción adicional (Fig. 9(b)). El incrementar las ganancias en este caso mejora el error de estado estacionario pero hace la respuesta transitoria más oscilatoria.

El control PID presenta una respuesta transitoria peor con un sobreimpulso más alto (Fig. 9(c)) pero trata de eliminar el error de estado estacionario, aunque es bastante lento (Fig. 9(d)). Es posible incrementar la rapidez del controlador PID incrementando  $K_i$  pero esto produce una mala respuesta transitoria con un gran sobreimpulso y muy oscilatoria, además de que el desempeño de seguimiento se empeora.

El controlador con redes neuronales presenta una respuesta ligeramente oscilatoria pero el sobreimpulso no es muy alto (Fig. 9(e)), siendo comparable a la del controlador PD, y siempre lleva

el error de estado estacionario a cero, incluso es capaz de compensar el término de fricción adicional (Fig. 9(f)).

## 5. CONCLUSIONES

La implementación práctica de un controlador de seguimiento multi-lazo con redes neuronales para un brazo flexible sencillo ha sido probada y su desempeño comparado con el de controladores estándar PD y PID. Se agregó un término de fricción adicional en la implementación para mostrar la habilidad del controlador con redes neuronales para aprender y compensar las no linealidades.

El controlador incluye un lazo de seguimiento exterior PD, un lazo interior de perturbación singular para estabilizar la dinámica rápida, y un lazo interior con la red neuronal usado para linearizar por retroalimentación la dinámica lenta. Este controlador con redes neuronales no requiere de una fase de aprendizaje fuera de línea, los pesos de la red neuronal son fácilmente inicializados, y garantiza acotamiento del error de seguimiento y la señal de control.

Los resultados prácticos corroboran las simulaciones mostrando que los controladores estándar PD o PID no son capaces de seguir una trayectoria deseada variable, mientras que el controlador con redes neuronales lleva el error de seguimiento a casi cero readaptándose a los cambios en el modelo del brazo (términos de fricción adicionales).

## 6. REFERENCIAS

- Centikunt, S., B. Siciliano, and W. J. Book (1986). Symbolic modeling and dynamic analysis of flexible manipulators. In: *Proc. IEEE Int. Conf. Syst., Man, Cybern.* 798-803.
- Chen, F.-C. and C.-C. Liu (1994). Adaptively controlling nonlinear continuous-time systems using multilayer neural networks. *IEEE Trans. Automat. Control*, 39, 1306-1310.
- De Luca, A. and B. Siciliano (1991). Closed-form dynamic model of planar multilink lightweight robots. *IEEE Trans. on Systems Man and Cybernetics*, 21, 826-839.
- Hastings, G. G. and W. J. Book (1986). Verification of a linear dynamic model for flexible robotic manipulators. In: *Proc. IEEE Int. Conf. Robotics and Automation*. 1024-1029.

- Hastings, G. G. and W. J. Book (1987). A linear dynamic model for flexible robotic manipulators. *IEEE Control Systems Magazine*, 7, 61-64.
- Kokotovic, P. V., R. E. O'Malley Jr., and P. Sannuti (1976). Singular perturbations and order reduction in control theory-An overview. *Automatica*, 12, 123-132.
- Kokotovic, P. V., H. K. Kalil, and J. O'Reilly (1986). *Singular perturbation methods in control: analysis and design*. Academic Press, London.
- Kokotovic, P. V. (1984). Applications of singular perturbation techniques to control problems. *SIAM Review*, 26, 501-550.
- Lewis, F. L., C. T. Abdallah, D. M. Dawson (1993a). *Control of robot manipulators*. Macmillan, New York.
- Lewis, F. L., K. Liu, and A. Yesildirek (1993b). Neural net robot controller: structure and stability proofs. In: *Proc. 32<sup>nd</sup> IEEE Conf. Decision and Control*. 2785-2791.
- Lewis, F. L., K. Liu, and A. Yesildirek (1995). Neural net robot controller with guaranteed tracking performance. *IEEE Trans. Neural Networks*, 6, 703-715.
- Lin, J., and F. L. Lewis (1993). Dynamic equations of a manipulator with rigid and flexible links: derivation and symbolic computation. In: *Proc. 1993 American Control Conf.* 2868-2872.
- Lin, S. H., S. Tosunoglu, and D. Tesar (1990). Control of a six-degree-of-freedom flexible industrial manipulator. *IEEE Control Systems Magazine*, pp. 24-30, Apr. 1990.
- Madhavan, S. K. and S. N. Singh (1991). Inverse trajectory control and zero dynamics sensitivity of an elastic manipulator. In: *Proc. 1991 American Control Conference*. 1879-1884.
- Narendra, K.S. and A.M. Annaswamy (1987). A new adaptive law for robust adaptation without persistent excitation. *IEEE Trans. Automat. Control*, AC-32, 134-145.
- Narendra, K.S. (1991). Adaptive Control Using Neural Networks. In: *Neural Networks for Control* (W.T. Miller, R.S. Sutton, P.J. Werbos, (Ed.)), 115-142. MIT Press, Cambridge.
- Polycarpou, M.M. and P.A. Ioannou (1991). Identification and control using neural network models: design and stability analysis. *Tech. Report 91-09-01*, Dept. Elect. Eng. Sys., Univ. S. Cal.
- Rovithakis G.A. and M.A. Christodoulou (1994). Adaptive control of unknown plants using dynamical neural networks. *IEEE Trans. Systems, Man, and Cybernetics*, 24, 400-412.
- Sadegh, N. (1993). A perceptron network for functional identification and control of nonlinear systems. *IEEE Trans. Neural Networks*, 4, 982-988.
- Sanner, R.M. and J.-J.E. Slotine (1991). Stable adaptive control and recursive identification using radial gaussian networks. In: *Proc. IEEE Conf. Decision and Control*.
- Siciliano, B. and W. J. Book (1988). A singular perturbation approach to control of lightweight manipulators. *Int. J. Robotics Research*, 7, 79-90.
- Vandegrift, M. W., F. L. Lewis, and S. Zhu (1994). Flexible-link robot arm control by a feedback linearization/singular perturbation approach. *J. Robotic Systems*, 11, 591-603.
- Wang, D. and M. Vidyagarasar (1991). Transfer functions for a single flexible link. *Int. J. Robotics Research*, 10, 540-549.
- Yesildirek, A., M. W. Vandegrift, and F. L. Lewis (1994). A neural network controller for flexible-link robots. In: *IEEE Int. Symp. Intelligent Control*. 63-68.