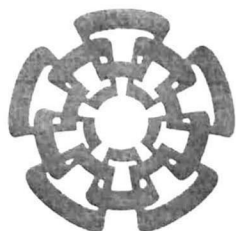




CTSG-429  
TESIS-2007

XX(132612.1)





# CINVESTAV

Centro de Investigación y de Estudios Avanzados del I.P.N.  
Unidad Guadalajara

---

## **Control Neuronal Descentralizado de Manipuladores Robóticos**

**CINVESTAV  
IPN  
ADQUISICION  
DE LIBROS**

Tesis que presenta:

**Armando Gaytán Godínez**

para obtener el grado de:

**Maestro en Ciencias**

en la especialidad de:

**Ingeniería Eléctrica**

Directores de Tesis

**Dr. Edgar Nelson Sánchez Camperos**

**Dr. Bernardino Castillo Toledo**

Guadalajara, Jalisco, Diciembre de 2006.

CLASIFICACION	TK165.58 .539 2006
ADICION	SSI - 430
FECHA	5-VII-2007
PROCL	DON - 2007

S.D 131992-2001



# **Control Neuronal Descentralizado de Manipuladores Robóticos**

**Tesis de Maestría en Ciencias  
Ingeniería Eléctrica**

Por:

**Armando Gaytán Godínez**  
Ingeniero en Sistemas Computacionales  
Instituto Tecnológico de Colima 1990-1996

Becario de CONACYT, expediente no. 112934

Directores de Tesis  
**Dr. Edgar Nelson Sánchez Camperos**  
**Dr. Bernardino Castillo Toledo**

CINVESTAV del IPN Unidad Guadalajara, Diciembre de 2006.

# ***Agradecimientos***

**Al CINVESTAV, por todos los valores recibidos en esta institución.**

**Al CONACYT, por el apoyo económico otorgado para mis estudios de maestría.**

**Al Colegio de Profesores del CINVESTAV, Unidad Guadalajara, por la oportunidad que se me ha dado para concluir mis estudios de maestría.**

**A todos los Doctores del CINVESTAV, Unidad Guadalajara, de quienes tuve el honor de recibir enseñanza.**

**A mis directores de tesis: Dr. Edgar Nelson Sánchez Camperos y Dr. Bernardino Castillo Toledo; del CINVESTAV, Unidad Guadalajara, México.**

**Al Dr. Maarouf Saad, de la Escuela de Tecnología Superior, Universidad de Québec, Canadá; un agradecimiento muy especial, por todo el apoyo que me ofreció durante el desarrollo de esta tesis; su colaboración ha sido indispensable para la realización de este trabajo.**



# Índice General

<b>1. Introducción</b>	<b>1</b>
1.1. Introducción al problema	1
1.2. Objetivo de la tesis	3
1.3. Organización de la tesis	4
<b>2. Esquema RHONN para identificación de sistemas dinámicos</b>	<b>6</b>
2.1. Introducción	6
2.2. El modelo RHONN	8
2.2.1. Propiedades de aproximación	11
2.3. Algoritmo de aprendizaje	12
2.3.1. Aprendizaje con error filtrado	13
2.4. Algoritmo de aprendizaje robusto	15
<b>3. Modelo de manipuladores robóticos</b>	<b>22</b>
3.1. Introducción	22
3.2. Ecuaciones fundamentales	23
3.3. Modelo del manipulador robótico de dos articulaciones	25
3.4. Modelo del manipulador robótico de siete articulaciones	27
3.5. Acotamiento de las dinámicas de manipuladores robóticos	29
3.6. Observaciones importantes para el control de manipuladores	30
<b>4. Identificación neuronal descentralizada de manipuladores robóticos</b>	<b>33</b>
4.1. Introducción	33
4.2. Modelo neuronal para identificación descentralizada	34
4.3. Esquema de identificación de la dinámica de la velocidad	38
4.3.1. Modelo con una interconexión de primer orden	38
4.3.2. Modelo con dos interconexiones de primer orden	39
4.3.3. Modelo con tres interconexiones, una de segundo orden	40
4.3.4. Modelo con interconexiones de alto orden	41
4.4. Esquema de identificación de la posición y de la velocidad	42
4.4.1. Modelo con dos neuronas, caso 1	43
4.4.2. Modelo con dos neuronas, caso 2	44
4.5. Identificación del manipulador robótico de dos articulaciones	45
4.5.1. Evaluación experimental	45
4.5.1.1. Descripción de experimentos	46
4.5.1.2. Interpretación de resultados	47

4.6. Identificación del manipulador robótico tipo ANAT	56
4.6.1. Evaluación experimental	57
4.6.1.1. Descripción de experimentos	57
4.6.1.2. Interpretación de resultados	58
<b>5. Control neuronal descentralizado de manipuladores robóticas</b>	<b>64</b>
5.1. Introducción	64
5.2. Método general de diseño del control	66
5.2.1. Control en el esquema de identificación de velocidad	69
5.2.1.1. Modelo con una interconexión de primer orden	70
5.2.1.2. Modelo con dos interconexiones de primer orden	70
5.2.2. Control con identificación de posición y velocidad	70
5.2.2.1. Modelo con dos neuronas, caso 1	71
5.2.2.2. Modelo con dos neuronas, caso 2	71
5.3. Control del manipulador robótico de dos articulaciones	72
5.3.1. Evaluación experimental	72
5.3.1.1. Descripción de experimentos	73
5.3.1.2. Interpretación de resultados	74
5.4. Control del manipulador robótico tipo ANAT	82
5.4.1. Evaluación experimental	82
5.4.1.1. Descripción de experimentos	83
5.4.1.2. Interpretación de resultados	84
5.5. Análisis de estabilidad	89
<b>6. Conclusiones y trabajo futuro</b>	<b>90</b>
6.1. Conclusiones	90
6.2. Trabajo futuro	91
<b>Bibliografía</b>	<b>92</b>
<b>APÉNDICE A. Gráficas de resultados</b>	<b>95</b>
<b>APÉNDICE B. Artículo presentado en “2006 IEEE International Symposium on Intelligent Control”.</b>	<b>116</b>



# *Resumen*

En este trabajo de tesis se propone un nuevo esquema de control neuronal descentralizado para manipuladores robóticos completamente actuados.

En el esquema propuesto se emplea una estructura particular de redes neuronales recurrentes. Con modelos neuronales independientes entre sí, se realiza la identificación en línea del modelo descentralizado del manipulador robótico; cada red neuronal es conectada en la topología serie-paralelo con la correspondiente articulación del manipulador, de esta manera la identificación se efectúa a través de variables locales únicamente.

Con el empleo de este esquema de identificación se evita la necesidad de conocer previamente tanto la estructura del modelo real, como la medición de sus parámetros físicos y geométricos.

El modelo de identificación resulta ser una colección de redes neuronales independientes entre sí, en el que la estructura de cada red neuronal corresponde al modelo de un sistema afín en el control; dado que el controlador diseñado para el modelo de identificación se aplica en paralelo al manipulador robótico, resulta fácil la síntesis de la ley de control para cada una de las uniones.

Se prueba la efectividad del esquema propuesto, a través de simulaciones, en un manipulador robótico de dos grados de libertad y en un manipulador robótico redundante de siete grados de libertad. Se experimentan diferentes alternativas de interconexión dentro de las redes neuronales, probando tanto con redes neuronales recurrentes de primer orden, como con redes neuronales recurrentes de alto orden.

Resulta un esquema atractivo para su implementación en tiempo real, para tareas de regulación y de seguimiento de trayectorias, por su efectividad, simplicidad y bajo costo de tiempo computacional.

# *Abstract*

This thesis presents a decentralized control scheme, based on a recurrent neural identifier, and its application to robotics manipulators.

The recurrent neural networks to have nonlinear approximation capabilities, a local neural identifier is proposed for each joint, a neural network model with a particular structure; identification in line of the robotic manipulator dynamics is realized in the decentralized scheme.

Each neural identifier use only local angular position and velocity measurements, the neural network models are connected with the joint correspondent. Therefore requesting not to know prior of model structure and elements, nor parametric measurements. The identification task is feasible with first order and high order neural networks.

The identifier model is independents neural networks collection. Is possible, for each neural network and therefore for each joint, to synthesize a control structure. The control scheme guarantee trajectory tracking; the error between the neuron states and the desired trajectories to result convergence to zero; the error between the joint positions and the desired trajectories to result convergence to the error of identification, that is minimized for the learning algorithm of the neural networks.

The applicability of the proposed scheme is illustrated, via simulations, by the applications to a two degree of freedom robotic manipulator and then to an Articulated Nimble Adaptable Truck (ANAT) one, with seven degree of freedom.

The obtained results are very encouraging. The local joint controllers allow trajectory tracking, with reduced computations. This control scheme is attractive for real-time applications.



# Capítulo 1

## Introducción

*En este capítulo, se presenta una introducción al problema de control descentralizado de manipuladores robóticos, se definen los objetivos de la tesis, y se describe brevemente la organización de la misma.*

### 1.1 Introducción al problema

Los manipuladores robóticos se han convertido en herramientas de trabajo muy importantes en la industria moderna, debido a que existe una necesidad cada vez más apremiante de aumentar la productividad y conseguir productos acabados de una calidad uniforme. Recientemente se observa además una tendencia al uso de manipuladores robóticos para la ejecución de tareas que resultan peligrosas para el ser humano y para operar en un medio ambiente inaccesible para los humanos; la cirugía de alta precisión es otro campo que está utilizando cada día más estos dispositivos. Estas demandas provocan a su vez la necesidad de nuevas técnicas de control de manipuladores robóticos que ofrezcan un mejor desempeño de acuerdo a la tarea o al grupo de tareas específico o a la diversidad de éstas [9], [32].

El control de manipuladores robóticos es un tema de investigación con un creciente interés en los últimos años; el rápido avance de la tecnología computacional ofrece cada vez más poder de implementación para los complejos algoritmos de control desarrollados dentro del

enfoque de diseño de un controlador central [11]. A pesar de ello, aún significa un gran reto obtener un control eficiente para esta clase de sistemas, por su dinámica compleja, altamente no lineal, con fuertes interconexiones, con parámetros difíciles de medir e incluso dinámicas difíciles de modelar. En el mejor de los casos, considerando solamente los términos dinámicos más importantes, el modelo matemático que resulta lleva a un algoritmo de control con gran número de operaciones matemáticas, lo cual afecta, por el alto costo computacional, la factibilidad de su implementación en tiempo real [8], [11], [20].

Por otra parte, dentro del área de la teoría de control de sistemas, desde hace más de tres décadas se ha venido desarrollando un enfoque alternativo en el que un sistema global se concibe como un conjunto de subsistemas interconectados para los cuales es posible diseñar controles independientes, considerando únicamente las variables locales inherentes a cada subsistema: el control descentralizado [15], [18]. Este enfoque surgió originalmente en el área de investigación de sistemas de gran escala, en los que un controlador central requiere intercambiar información con todos los subsistemas, lo que puede ser físicamente muy difícil; puede también suceder que la unidad controladora requiera de capacidades computacionales no disponibles, y además puede resultar un sistema de control de un extenso número de parámetros, con incertidumbre en sus mediciones [4], [15], [18].

Dentro del campo de la robótica el enfoque de control descentralizado ha sido tema de investigación en sistemas de múltiples vehículos cooperativos y manipuladores robóticos fijos cooperativos; en estos sistemas, es posible considerar a cada componente como un subsistema que puede ser representado por un modelo matemático con dinámicas locales y las dinámicas de interacción como términos adicionales. En los últimos años se han publicado interesantes investigaciones sobre control descentralizado de manipuladores robóticos individuales, en los que cada una de las articulaciones es un subsistema y el objetivo del enfoque es diseñar controles independientes en función solamente de la posición y la velocidad de la articulación correspondiente, compensando las fuertes dinámicas de acoplamiento, resultando un diseño simplificado, más factible de implementar en línea y más fácil de hacerlo tolerante a fallas [19], [20], [24].

Para fines de obtener un controlador descentralizado para manipuladores robóticos, diferentes investigadores han hecho uso de técnicas de inteligencia artificial, en particular

lógica difusa [19] y redes neuronales [20], [31]; los dos últimos explotan la probada capacidad de las redes neuronales dinámicas recurrentes para la aproximación de mapeos no lineales [21], [22]. En [25], se desarrolla un control descentralizado de manipuladores robóticos, descomponiendo el modelo dinámico del manipulador en un conjunto de subsistemas lineales con incertidumbres y se presentan simulaciones para un robot de dos articulaciones. En [20], se expone un enfoque de identificación y control neuronal descentralizado para manipuladores robóticos empleando modelos de tiempo discreto. En [31], se presenta un control descentralizado para manipuladores robóticos basado en la estimación de dinámicas de manera independiente para cada una de las uniones, utilizando redes neuronales. Obtienen resultados en tiempo real para un robot de tres articulaciones. En [13], se expone el diseño de un controlador descentralizado para manipuladores robóticos variantes en el tiempo; la estrategia consiste en diseñar una ley de control adaptable, en función del error y de variables locales. Los resultados son probados en simulación con un manipulador robótico de dos articulaciones.

Los conocimientos anteriores motivan a la investigación de una solución alternativa al problema de control descentralizado de manipuladores robóticos, haciendo uso de redes neuronales recurrentes.

## **1.2 Objetivo de la tesis**

El objetivo general de esta tesis es diseñar un esquema descentralizado de identificación y control para manipuladores robóticos.

El primer objetivo particular es usar redes neuronales recurrentes para implementar un enfoque descentralizado de identificación de manipuladores robóticos, usando sólo información local de cada articulación.

El segundo objetivo particular es desarrollar un método de síntesis de control con base en el modelo de identificación descentralizada, que permita alcanzar regulación y seguimiento de trayectorias en el espacio de las uniones.

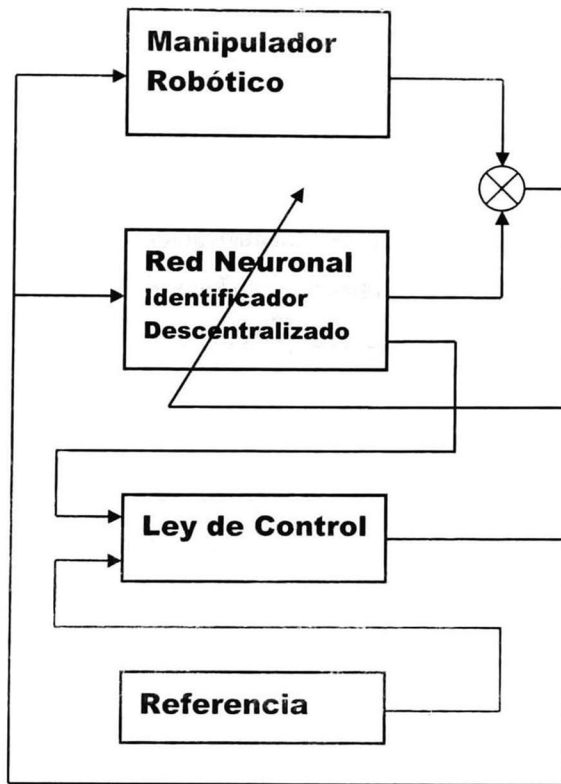


Figura 1.1. Esquema de Control Neuronal Descentralizado

### 1.3 Organización de la tesis

El contenido de esta memoria de tesis está organizado, a partir de este capítulo, de la siguiente manera:

Capítulo 2, en el que se exponen los fundamentos teóricos de la identificación de sistemas no lineales con modelos de redes neuronales; se describe la estructura general de los modelos que se emplean en este trabajo, sus propiedades de aproximación y el desarrollo de la ley de aprendizaje.

Capítulo 3. En éste se describe el modelo dinámico general de un manipulador robótico, las características de comportamiento dinámico fundamentales en las que se apoya esta investigación, el modelo particular del manipulador robótico de dos articulaciones y la



descripción del manipulador robótico de siete articulaciones, ambos empleados en las simulaciones realizadas dentro del desarrollo de este trabajo.

Capítulo 4, en el cual se presenta el modelo general de red neuronal utilizado como identificador descentralizado del manipulador robótico, se exponen los diferentes modelos ensayados en la identificación y se describen sus características particulares, se presentan resultados de simulaciones realizadas con cada uno de los manipuladores robóticos antes mencionados.

Capítulo 5, en el que se desarrolla el diseño general del controlador basado en el modelo descentralizado de identificación tipo red neuronal recurrente, se sintetiza el control para cada uno de los modelos aplicados a cada uno de los manipuladores robóticos y se presentan los resultados de simulaciones realizadas, en gráficas, y se interpretan dichos resultados.

Capítulo 6. Aquí se presentan las conclusiones respectivas y el trabajo a futuro.

Apéndice A, en el que se muestra un compendio de figuras, con gráficas correspondientes a resultados de simulaciones ejecutadas dentro del desarrollo de esta tesis. El lector puede acudir a estas gráficas si desea realizar una observación más completa de los resultados que se describen en los capítulos cuatro y cinco.

Apéndice B, artículo presentado en “2006 IEEE International Symposium on Intelligent Control”



## Capítulo 2

# Esquema RHONN para identificación de sistemas dinámicos

*Los modelos de red neuronal propuestos en esta tesis, para la aproximación descentralizada de la dinámica de manipuladores robóticos, están basados en el esquema de red neuronal recurrente de alto orden (RHONN, por sus siglas en inglés, “Recurrent High Order Neural Network”). En este capítulo se expone la estructura, las leyes de aprendizaje y sus propiedades de aproximación, convergencia y robustez.*

### 2.1 Introducción

Dado que para diseñar y analizar el controlador de un sistema se requiere de una caracterización matemática del modelo respectivo [3], [29] y ante la idea de considerar a cada par unión-eslabón del manipulador robótico como un subsistema a ser controlado de manera independiente, se hace necesario desarrollar un esquema de identificación descentralizada de la dinámica de cada una de las uniones.

Rovithakis y Christodoulou [29], describen que el problema de identificación consiste en seleccionar un modelo de identificación apropiado y ajustar sus parámetros de acuerdo a alguna ley de adaptación, de modo tal que la respuesta del modelo a una señal de entrada, o a una clase de señales de entrada, aproxime la respuesta del sistema real a la misma entrada.

Desde los primeros años de la década de 1990, se ha venido incrementando sustancialmente el interés por el uso de redes neuronales para el modelado e identificación de sistemas dinámicos, luego de que trabajos teóricos como los de Cybenko [6] y Funahashi [10], ambos en 1989, probaran que, aún con una sola capa oculta, las redes neuronales pueden aproximar cualquier función continua uniformemente sobre un dominio compacto.

Una red neuronal recurrente (RNN, por sus siglas en inglés, “Recurrent Neural Network”) es aquella red neuronal que involucra elementos dinámicos en la forma de conexiones de retroalimentación; la conectividad entre unidades (neuronas) es en los dos sentidos. Esto la distingue de una red neuronal prealimentada, donde la salida de una unidad es conectada sólo a unidades de la próxima capa. Una RHONN es una expansión de una RNN de primer orden, como la de Hopfield [12] y la de Cohen-Grossberg [5], con interacciones de alto orden entre neuronas. La idea del modelo RHONN y sus aplicaciones para identificación de sistemas dinámicos fue propuesta por Kosmatopoulos y colaboradores [22] en 1995; la gran capacidad de memoria de las redes neuronales había sido demostrada en [2], [26], mientras que las propiedades de estabilidad para estos modelos, con pesos fijos, habían sido estudiadas en [7]. El comportamiento dinámico y las propiedades de estabilidad de las redes neuronales recurrentes han sido extensamente estudiados por varios investigadores, entre ellos [22], [29].

Inicialmente los trabajos publicados mostraron buen desempeño en aplicaciones, pero carecieron de resultados analíticos concernientes a convergencia y estabilidad [30]. Posteriormente, se presenta una concentración de esfuerzos hacia el diseño y análisis de algoritmos de aprendizaje basados en la teoría de estabilidad de Lyapunov [22], [29], [30], dirigidos a establecer las demostraciones de estabilidad, convergencia y robustez.

En esta tesis se prueban estructuras tipo RHONN y tipo RNN como modelos de identificación. A continuación se expone el modelo general de una RHONN, notando que el caso más simple, de orden uno, corresponde al modelo RNN, el cual es el más utilizado en esta tesis. Se exponen las propiedades de aproximación del modelo RHONN, y el algoritmo de aprendizaje utilizado en este trabajo, incluyendo las modificaciones de robustez. En su mayor parte lo que prosigue en este capítulo está basado en los trabajos [22], [29], [30].

## 2.2 El modelo RHONN

En esta sección se explica como las redes neuronales recurrentes de alto orden son capaces de modelar una clase amplia de sistemas dinámicos. Si se utiliza un número suficiente de conexiones de alto orden en la red entonces, existen valores de pesos tales que, el comportamiento entrada-salida del modelo RHONN aproxima al de un sistema dinámico arbitrario cuya trayectoria de estados permanece en un dominio compacto.

En el caso más simple, un modelo RNN, el estado de cada neurona es gobernado por una ecuación diferencial de la forma:

$$\dot{x}_i = -a_i x_i + b_i \sum_j w_{ij} y_j, \quad (2.1)$$

donde  $x_i$  es el estado de la  $i$ -ésima neurona,  $a_i$ ,  $b_i$  son constantes,  $w_{ij}$  es el peso sináptico conectando la  $j$ -ésima entrada a la  $i$ -ésima neurona y  $y_j$  es la  $j$ -ésima entrada a la  $i$ -ésima neurona. Cada  $y_j$  es una entrada externa o el estado de una neurona pasado a través de una función sigmoideal (esto es,  $y_j = s(x_j)$ ), donde  $s(\cdot)$  es dicha función no lineal.

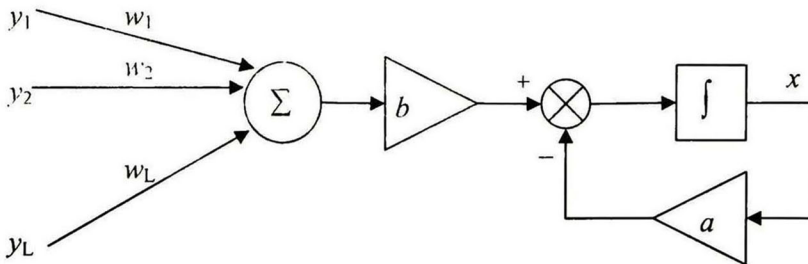


Figura 2.1. Esquema representativo para cada neurona dinámica del modelo (2.1).

En una red neuronal recurrente de segundo orden, la entrada a la neurona no es sólo una combinación lineal de los componentes  $y_j$ , sino también de sus productos  $y_j y_k$ ; se podría continuar en este sentido, para incluir interacciones de alto orden, representadas por tripletas  $y_j y_k y_l$ , cuádruplas, quintuplas, etc., formando así los modelos tipo RHONN.

Si se considera ahora una RHONN de  $n$  neuronas y  $m$  entradas, el estado de cada neurona es gobernado por una ecuación diferencial de la forma:

$$\dot{x}_i = -a_i x_i + b_i \sum_{k=1}^L w_{ik} \prod_{j \in I_k} y_j^{d_j(k)} \quad (2.2)$$

en la cual  $\{I_1, I_2, \dots, I_L\}$  es una colección de  $L$  subconjuntos no ordenados de  $\{1, 2, \dots, m+n\}$ ,  $a_i, b_i$  son coeficientes reales,  $w_{ik}$  son los pesos sinápticos (ajustables) de la red neuronal y  $d_j(k)$  son enteros no negativos. Los estados de la  $i$ -ésima neurona están representados por  $x_i$ , y  $\mathbf{y} = [y_1, y_2, \dots, y_{m+n}]^T$  es el vector de entrada a cada neurona definido por:

$$\mathbf{y} = \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_n \\ y_{n+1} \\ \vdots \\ y_{n+m} \end{bmatrix} = \begin{bmatrix} s(x_1) \\ s(x_2) \\ \vdots \\ s(x_n) \\ u_1 \\ \vdots \\ u_m \end{bmatrix},$$

donde  $\mathbf{u} = [u_1, u_2, \dots, u_m]^T$  es el vector de entrada externo a la red neuronal. La función  $s(\cdot)$  es monótonamente creciente, diferenciable y es usualmente representada por funciones sigmoideas de la forma:

$$s(x) = \frac{\phi}{1 + \exp(-\eta x)} - \delta,$$

en la cual los parámetros  $\phi, \eta$  representan respectivamente, la cota y la pendiente de la sigmoidea y  $\delta$  es una constante de corrimiento, de modo que  $s(x) \in (-\delta, \phi - \delta)$ . En el caso especial donde  $\phi = \eta = 1, \delta = 0$ , se tiene la función logística y con  $\phi = \eta = 2, \delta = 1$ , la función tangente hiperbólica; éstas son las funciones de activación sigmoideas más comúnmente usadas en aplicaciones de redes neuronales.

Para simplificar la expresión (2.2) se introduce un vector  $\mathbf{z}$ , de dimensión  $L$ , que es definido así:



$$\mathbf{z} = \begin{bmatrix} z_1 \\ z_2 \\ \vdots \\ z_L \end{bmatrix} = \begin{bmatrix} \prod_{j \in I_1} y_j^{d_j^{(1)}} \\ \prod_{j \in I_2} y_j^{d_j^{(2)}} \\ \vdots \\ \prod_{j \in I_L} y_j^{d_j^{(L)}} \end{bmatrix}, \quad (2.3)$$

entonces, el modelo RHONN (2.2) se convierte en

$$\dot{x}_i = -a_i x_i + b_i \left[ \sum_{k=1}^L w_{ik} z_k \right]. \quad (2.4)$$

Aún más, si se define el vector de pesos ajustables como  $\mathbf{w}_i = b_i [w_{i1} \ w_{i2} \ \dots \ w_{iL}]^T$ , entonces (2.4) se convierte en

$$\dot{x}_i = -a_i x_i + \mathbf{w}_i^T \mathbf{z},$$

donde los vectores  $\{\mathbf{w}_i : i = 1, 2, \dots, n\}$  representan los pesos ajustables de la red, mientras que los coeficientes  $\{a_i : i = 1, 2, \dots, n\}$  son parte de la arquitectura de la red y son determinados durante el entrenamiento.

Para garantizar que cada neurona  $x_i$  sea estable en el sentido entrada acotada-salida acotada (BIBO estable), se debe suponer que  $a_i > 0, \forall i = 1, 2, \dots, n$ . En el caso especial de un modelo Hopfield de tiempo continuo [12], se tiene  $a_i = 1/R_i C_i$ , donde  $R_i > 0$  y  $C_i > 0$  son la resistencia y capacitancia conectadas al  $i$ -ésimo nodo de la red, respectivamente.

El comportamiento dinámico de la red general es descrito en notación vectorial como:

$$\dot{\mathbf{x}} = \mathbf{A}\mathbf{x} + \mathbf{W}^T \mathbf{z}, \quad (2.5)$$

donde  $\mathbf{x} = [x_1, x_2, \dots, x_n]^T \in \mathfrak{R}^n$  es el vector de estados del modelo neuronal, mientras que  $\mathbf{W} = [\mathbf{w}_1, \mathbf{w}_2, \dots, \mathbf{w}_n]^T \in \mathfrak{R}^{n \times L}$ , es la matriz conformada por los vectores de pesos sinápticos, y  $\mathbf{A} = \text{diag}\{-a_1, -a_2, \dots, -a_n\} \in \mathfrak{R}^{n \times n}$  es una matriz diagonal. Dado que  $a_i > 0$ , para todo  $i \in \{1, 2, \dots, n\}$ ,  $\mathbf{A}$  es una matriz estable. Aunque no está explícitamente escrito,



cada elemento del vector  $\mathbf{z}$  es una función dependiente de los estados de la red neuronal, es decir, de los elementos del vector  $\mathbf{x}$ , y de la entrada externa  $\mathbf{u}$ .

Es importante hacer notar que en el vector de entrada a cada neurona, en sus últimos  $m$  componentes pueden incluirse las entradas externas directamente o bien haciéndolas pasar antes a través de funciones sigmoideas. La primera opción es conveniente para generar una estructura afin en el control [28].

## 2.2.1 Propiedades de aproximación

Se considera el problema de aproximación de un sistema dinámico no lineal general cuyo comportamiento entrada-salida está dado por

$$\dot{\chi} = F(\chi, \mathbf{u}), \quad (2.6)$$

donde  $\chi \in \mathcal{R}^n$  es el estado del sistema,  $\mathbf{u} \in \mathcal{R}^m$  es la entrada del sistema y  $F : \mathcal{R}^{n+m} \rightarrow \mathcal{R}^n$  es un campo vector suave definido en un conjunto compacto  $Y \subset \mathcal{R}^{n+m}$

El problema de aproximación consiste en determinar si con conexiones suficientes y de alto orden, existen pesos  $\mathbf{W}$ , tales que, el modelo RHONN (2.5) aproxime el comportamiento entrada-salida de un sistema dinámico arbitrario de la forma (2.6).

Para tener un problema bien definido, se supone que  $F(\cdot)$  es continua y satisface la condición local de Lipschitz, de modo que (2.6) tiene una única solución en el sentido de Caratheodory y  $(\mathbf{x}(t), \mathbf{u}(t)) \in Y$  para toda  $t$  en algún intervalo de tiempo  $J_T := \{t : 0 \leq t \leq T\}$ . El intervalo  $J_T$  representa el periodo de tiempo sobre el cual la aproximación es efectuada. Con base en la suposición anterior se obtiene el siguiente resultado.

**Teorema 2.2.1.** *Suponiendo que el sistema real (2.6) y el modelo de identificación (2.5) están inicialmente en el mismo estado  $\chi(0) = \mathbf{x}(0)$ ; entonces para cualquier escalar  $\varepsilon > 0$  y cualquier tiempo finito  $T > 0$ , existe un entero  $L$  y una matriz  $\mathbf{W}^* \in \mathcal{R}^{L \times n}$  tales que, el estado  $\mathbf{x}(t)$  del modelo RHONN (2.5) con  $L$  interconexiones de alto orden y valores de pesos  $\mathbf{W} = \mathbf{W}^*$  satisface esta condición:*

$$\sup_{0 \leq t \leq T} \|x(t) - \chi(t)\| \leq \varepsilon$$

Este teorema, cuya demostración puede encontrarse en [30], establece que si un número suficientemente grande de conexiones de alto orden se introduce al modelo RHONN entonces, es posible aproximar cualquier sistema dinámico con cualquier grado de precisión. Esto es estrictamente un resultado de existencia; no proporciona ningún método constructivo para obtener los pesos óptimos  $W^*$ . A continuación, se considera el problema de aprendizaje de ajustar los pesos adaptablemente, de modo que el modelo RHONN identifique sistemas dinámicos generales.

## 2.3 Algoritmo de aprendizaje

El contenido de esta sección está basado en los algoritmos de aprendizaje que se exponen en [30]. Se desarrolla la ley de ajuste de pesos bajo la suposición de que el sistema desconocido es modelado exactamente por una arquitectura RHONN de la forma (2.5). Se demuestra que esa ley de ajuste de pesos garantiza acotamiento de todas las señales y pesos y además, el error de salida converge a cero. Este análisis es extendido en la siguiente sección para cubrir el caso en el que existe una diferencia entre el sistema y el modelo RHONN con valores de pesos óptimos. Esta diferencia es conocida como el error de modelado.

Aunque la suposición de que no existe error de modelado no es real, el procedimiento de identificación de esta sección es útil por dos razones; primero, por que el análisis es más directo y más fácil de entender, y segundo, por que las técnicas desarrolladas para el caso en el que no hay error de modelado son también importantes en el diseño de leyes de adaptación de pesos en la presencia de errores de modelado.

Con base en la suposición de que no hay error de modelado, existen vectores de pesos óptimos desconocidos  $w_i^*$ , para  $i = 1, 2, \dots, n$ , tales que

$$\dot{\chi}_i = -a_i \chi_i + w_i^{*T} z(\chi, u),$$

para cada estado  $\chi_i$  del sistema dinámico desconocido (2.6), donde el  $i$ -ésimo estado inicial del sistema es  $\chi_i(0) = \chi_i^0$ .

A partir de aquí, solo para mayor simplicidad en el procedimiento matemático, los argumentos del campo vector  $\mathbf{z}$  son omitidos.

Como es usual en los procedimientos de identificación de sistemas, se supone que la entrada  $\mathbf{u}(t)$  y el estado  $\chi(t)$  permanecen acotados para todo  $t \geq 0$ . Con base en la definición de  $\mathbf{z}(\chi, \mathbf{u})$ , dada por (2.3), esto implica que  $\mathbf{z}(\chi, \mathbf{u})$  es también acotado. En la siguiente sección se presenta un enfoque para estimar los parámetros desconocidos  $\mathbf{w}_i^*$  del modelo RHÖNN, seleccionado por su estructura simple, que requiere sólo  $n$  filtros y por lo tanto, computación más simple, en comparación con el enfoque de regresor filtrado también presentado en [30].

### 2.3.1 Aprendizaje con error filtrado

El desarrollo de este esquema de identificación parte de la ecuación diferencial que describe al sistema desconocido, o sea,

$$\dot{\chi}_i = -a_i \chi_i + \mathbf{w}_i^{*T} \mathbf{z}, \quad (2.7)$$

para  $i = 1, 2, \dots, n$ . Con base en (2.22), el identificador es seleccionado como

$$\dot{\hat{x}}_i = -a_i \hat{x}_i + \mathbf{w}_i^T \mathbf{z}, \quad (2.8)$$

donde  $\mathbf{w}_i$  es la estimación del vector desconocido  $\mathbf{w}_i^*$ . En este caso el error de estimación de estado definido por  $e_i := \hat{x}_i - \chi_i$ , satisface

$$\dot{e}_i = -a_i e_i + \phi_i^T \mathbf{z}, \quad (2.9)$$

donde  $\varphi_i = \mathbf{w}_i - \mathbf{w}_i^*$ . Los pesos  $\mathbf{w}_i$ , para  $i = 1, 2, \dots, n$ , son ajustados de acuerdo a la siguiente ley de aprendizaje:

$$\dot{\mathbf{w}}_i = -\Gamma_i \mathbf{z} e_i, \quad (2.10)$$

donde la matriz de ganancia de aprendizaje  $\Gamma_i \in \mathfrak{R}^{L \times L}$  debe ser una matriz definida positiva. En el caso particular de que  $\Gamma_i = \gamma_i \mathbf{I}$ , con un escalar positivo  $\gamma_i$ , y la matriz identidad  $\mathbf{I}$ , entonces  $\Gamma_i$  en (2.10) puede reemplazarse por  $\gamma_i$ :

$$\dot{\mathbf{w}}_i = -\gamma_i \mathbf{z} e_i.$$

El siguiente teorema, presentado en [30], demuestra las propiedades de convergencia de este esquema de identificación.

**Teorema 2.3.1.** *Considerando al modelo RHONN error filtrado dado por (2.8) cuyos pesos son ajustados de acuerdo a (2.10). Entonces para  $i = 1, 2, \dots, n$ , se cumplen (a) y (b):*

- (a)  $e_i, \varphi_i \in \mathcal{L}_\infty$  ( $e_i$  y  $\varphi_i$  están uniformemente acotados)
- (b)  $\lim_{t \rightarrow \infty} e_i(t) = 0$

**Demostración:** Considérese la función candidata de Lyapunov

$$V = \frac{1}{2} \sum_{i=1}^n (e_i^2 + \varphi_i^T \Gamma_i^{-1} \varphi_i). \quad (2.11)$$

Luego, usando (2.9), (2.10), y el hecho de que  $d\varphi_i/dt = d\mathbf{w}_i/dt$ , la derivada en el tiempo de  $V$  en (2.11) satisface esta desigualdad:

$$\dot{V} = -\sum_{i=1}^n a_i e_i^2 \leq 0. \quad (2.12)$$

Dado que  $dV/dt \leq 0$ , de (2.12), se obtiene que  $e_i, \varphi_i \in L_\infty$  para  $i = 1, 2, \dots, n$ . Usando este resultado en (2.9) se tiene que  $de_i/dt \in L_\infty$ . Para demostrar que  $e_i(t)$  converge a cero, primero se observa que  $V$  es una función no creciente en el tiempo, y además está acotada, por lo tanto  $\lim_{t \rightarrow \infty} V(t) = V_\infty$ , donde  $V_\infty$  es finito; de esta manera, integrando ambos lados de (2.12) desde  $t = 0$  hasta  $\infty$ , y tomando las cotas se obtiene que

$$\int_0^\infty \sum_{i=1}^n a_i e_i^2(t) dt = V(0) - V_\infty,$$

y así, para  $i = 1, 2, \dots, n$ ,  $e_i(t)$  es cuadrado integrable. Además, usando (2.24), dado que  $e_i, \varphi_i, z$ , son todas acotadas,  $de_i/dt \in L_\infty$ . Entonces, aplicando el teorema de Barbalat [30] se obtiene que  $\lim_{t \rightarrow \infty} e_i(t) = 0$ . A partir de lo anterior y considerando (2.9) se obtiene que necesariamente  $\lim_{t \rightarrow \infty} \varphi_i(t) = 0$ , esto es, el error de estimación converge a cero.

## 2.4 Algoritmo de aprendizaje robusto

Para esta sección se toman los algoritmos de aprendizaje robusto propuestos en [30]. El análisis es extendido al caso en el que sí existe una diferencia entre el sistema y el modelo RHONN con valores de pesos óptimos.

La derivación del algoritmo de aprendizaje desarrollado en la sección previa hace la crucial suposición de que no hay error de modelado. Equivalentemente, se supuso que existen vectores de pesos  $w_i^*$  para  $i = 1, 2, \dots, n$ , tales que cada estado del sistema dinámico desconocido (2.6) satisface

$$\dot{\chi}_i = -a_i \chi_i + w_i^{*T} z(\chi, u). \quad (2.13)$$

En muchos casos esta suposición no se cumple, debido principalmente a un número insuficiente de términos de alto orden en el modelo RHONN. En tales casos, si se usa una ley adaptable como (2.10), para ajustar los pesos, la presencia de error de modelado puede



causar que los valores de los pesos ajustables diverjan; y, consecuentemente, el error de identificación también puede divergir.

En esta sección se modifica la ley de ajuste de pesos para evitar que estos parámetros evolucionen hacia valores inconvenientemente altos. Esta ley de ajuste modificada es referida en [30] como algoritmo de aprendizaje robusto. El comportamiento dinámico de cada estado del sistema (2.6) puede expresarse con esta ecuación diferencial:

$$\dot{\chi}_i = -a_i \chi_i + \mathbf{w}_i^{*T} \mathbf{z}(\boldsymbol{\chi}, \mathbf{u}) + v_i, \quad (2.14)$$

en la cual el error de modelado  $v_i(t)$  está dado por

$$v_i(t) = F_i(\boldsymbol{\chi}(t), \mathbf{u}(t)) + a_i \chi_i - \mathbf{w}_i^{*T} \mathbf{z}(\boldsymbol{\chi}(t), \mathbf{u}(t)).$$

La función  $F_i(\boldsymbol{\chi}, \mathbf{u})$  denota el  $i$ -ésimo componente del campo vector  $\mathbf{F}(\boldsymbol{\chi}, \mathbf{u})$ , mientras que el vector de pesos óptimo desconocido  $\mathbf{w}_i^*$  es definido como el valor del vector de pesos  $\mathbf{w}_i$  que minimiza la  $L_\infty$ -norma de la diferencia entre  $F_i(\boldsymbol{\chi}, \mathbf{u}) + a_i \chi_i$  y  $\mathbf{w}_i^{*T} \mathbf{z}(\boldsymbol{\chi}, \mathbf{u})$  para todo  $(\boldsymbol{\chi}, \mathbf{u}) \in Y \subset \mathfrak{R}^{n+m}$ , sujeto a la restricción de que  $|\mathbf{w}_i| < M_i$ , donde  $M_i$  es una constante grande previamente seleccionada. La región  $Y$  denota el más pequeño subconjunto compacto de  $\mathfrak{R}^{n+m}$  que incluye todos los valores que  $(\boldsymbol{\chi}, \mathbf{u})$  puede tomar, o sea,  $(\boldsymbol{\chi}(t), \mathbf{u}(t)) \in Y \subset \mathfrak{R}^{n+m}$  para todo  $t \geq 0$ . Dado que por suposición  $\mathbf{u}(t)$  es uniformemente acotada y el sistema dinámico a ser identificado es estable en el sentido entrada acotada-salida acotada (BIBO estable), la existencia de tal  $Y$  es garantizada. En este análisis no se requiere conocer la región  $Y$ , ni las cotas para el error de modelado  $v_i(t)$ .

En resumen, para  $i = 1, 2, \dots, n$ , el vector de pesos óptimo  $\mathbf{w}_i^*$  es definido así:

$$\mathbf{w}_i^* = \arg \min_{|\mathbf{w}_i| \leq M_i} \left\{ \sup_{(\boldsymbol{\chi}, \mathbf{u}) \in Y} |F_i(\boldsymbol{\chi}, \mathbf{u}) + a_i \chi_i - \mathbf{w}_i^T \mathbf{z}(\boldsymbol{\chi}, \mathbf{u})| \right\}$$

La razones para restringir  $\mathbf{w}_i^*$  a una esfera de radio  $M_i$  son dos: primeramente, para evitar cualquier problema numérico, y segundo, para permitir el uso de la modificación  $\sigma$  [16], la cual será desarrollada para evitar el problema de la divergencia de parámetros.

Con la formulación anterior, ahora tenemos un sistema de la forma (2.14) en lugar de (2.13). Se observa que a partir de que  $\chi(t)$  y  $\mathbf{u}(t)$  son acotados, el error de modelado  $v_i(t)$  es también acotado; esto es  $\sup_{t \geq 0} |v_i(t)| \leq v_i^-$  para una constante finita  $v_i^-$ .

A continuación se desarrolla al algoritmo de aprendizaje robusto basado en el identificador RHONN error filtrado. Por lo tanto, el identificador es seleccionado como en (2.8), esto es,

$$\dot{x}_i = -a_i x_i + \mathbf{w}_i^T \mathbf{z}, \quad (2.15)$$

donde  $\mathbf{w}_i$  es la estimación del vector de pesos óptimo desconocido  $\mathbf{w}_i^*$ , para  $i = 1, 2, \dots, n$ . Usando (2.14) y (2.15), el error de estado  $e_i = x_i - \chi_i$  satisface que

$$\dot{e}_i = -a_i e_i + \boldsymbol{\varphi}_i^T \mathbf{z} - v_i, \quad (2.16)$$

donde  $\boldsymbol{\varphi}_i = \mathbf{w}_i - \mathbf{w}_i^*$ . Debido a la presencia del error de modelado  $v_i$ , la ley de aprendizaje dada por (2.10) se modifica como sigue:

$$\dot{\mathbf{w}}_i = \begin{cases} -\Gamma_i \mathbf{z} e_i & \text{si } |\mathbf{w}_i| \leq M_i \\ -\Gamma_i \mathbf{z} e_i - \sigma_i \Gamma_i \mathbf{w}_i & \text{si } |\mathbf{w}_i| > M_i \end{cases} \quad (2.17)$$

donde  $\sigma_i$  es una constante positiva seleccionada por el diseñador. La ley de ajuste de pesos anterior es la misma que (2.10) si  $\mathbf{w}_i$  pertenece a una esfera de radio  $M_i$ . En el caso de que los pesos salgan de esa esfera, la ley de ajuste de pesos los modifica por la adición del término  $-\sigma_i \Gamma_i \mathbf{w}_i$ , cuyo objetivo es prevenir que los valores de los pesos tiendan al infinito. Esta modificación es conocida como la modificación  $\sigma$  [16].

Para el siguiente teorema se considera  $\mathbf{v} := [v_1 \ v_2 \ \dots \ v_n]^T$  y  $\mathbf{e} := [e_1 \ e_2 \ \dots \ e_n]^T$

**Teorema 2.4.1.** *Considérese el modelo (2.15), cuyos pesos son ajustados de acuerdo a (2.17), y el error filtrado dado por (2.16). Luego para  $i = 1, 2, \dots, n$ , se cumplen (a) y (b):*

(a)  $e_i, \varphi_i \in L_x$  ( $e_i$  y  $\varphi_i$  están uniformemente acotados)

(b) existe constantes  $\lambda, \mu$  tal que

$$\int_0^t |e(\zeta)|^2 d\zeta \leq \lambda + \mu \int_0^t |v(\zeta)|^2 d\zeta \quad (2.18)$$

**Demostración:** Considérese la función candidata de Lyapunov

$$V = \frac{1}{2} \sum_{i=1}^n (e_i^2 + \varphi_i^T \Gamma_i^{-1} \varphi_i). \quad (2.19)$$

Usando (2.16) y (2.17) puede demostrarse que

$$\dot{V} = \sum_{i=1}^n (-a_i e_i^2 - e_i v_i - I_{w_i}^* \sigma_i \varphi_i^T w_i), \quad (2.20)$$

donde  $I_{w_i}^*$  es la función indicador definida como  $I_{w_i}^* = 1$  si  $|w_i| > M_i$ , y  $I_{w_i}^* = 0$  si  $|w_i| \leq M_i$ .

A partir de que  $\varphi_i = w_i - w_i^*$ , puede verificarse que

$$\varphi_i^T w_i = \frac{1}{2} |\varphi_i|^2 + \frac{1}{2} |w_i|^2 - \frac{1}{2} |w_i^*|^2$$

Dado que, por definición,  $|w_i^*| \leq M_i$ , y  $|w_i| > M_i$  para  $I_{w_i}^* = 1$ , se tiene que

$$I_{w_i}^* \frac{\sigma_i}{2} (|w_i|^2 - |w_i^*|^2) \geq 0.$$

Haciendo uso de las dos expresiones matemáticas previas, a partir de (2.20) se obtiene que

$$\dot{V} \leq \sum_{i=1}^n \left( -a_i e_i^2 - I_{w_i}^* \frac{\sigma_i}{2} |\varphi_i|^2 - e_i v_i \right). \quad (2.21)$$

Por conveniencia para el procedimiento, esta desigualdad puede ser expresada de esta forma:

$$\dot{V} \leq \sum_{i=1}^n \left( -\frac{a_i}{2} e_i^2 - \frac{\sigma_i}{2} |\varphi_i|^2 \right) + \sum_{i=1}^n \left( (1 - I_{w_i}^*) \frac{\sigma_i}{2} |\varphi_i|^2 - \frac{a_i}{2} e_i^2 - e_i v_i \right).$$

Ahora, considerando (2.19) y que  $(a_i e_i + v_i)^2 \geq 0$ , se obtiene que

$$\dot{V} \leq -\alpha V + \sum_{i=1}^n \left( (1 - I_{w_i}^*) \frac{\sigma_i}{2} |\varphi_i|^2 + \frac{v_i^2}{2a_i} \right), \quad (2.22)$$

donde  $\alpha = \min \left\{ a_i, \frac{\sigma_i}{\lambda_{\max}(\Gamma_i)}; i = 1, 2, \dots, n \right\}$  y  $\lambda_{\max}(\Gamma_i^{-1}) > 0$  denota el máximo valor propio

de  $\Gamma_i^{-1}$ . A partir de

$$(1 - I_{w_i}^*) \frac{\sigma_i}{2} |\varphi_i|^2 = \begin{cases} \frac{\sigma_i}{2} |\varphi_i|^2 & \text{si } |w_i| \leq M_i \\ 0 & \text{en otro caso} \end{cases}$$

se tiene que  $(1 - I_{w_i}^*) (\sigma_i/2) |\varphi_i|^2 \leq \sigma_i M_i^2$ . Por lo tanto (2.22) puede reescribirse así:

$$\dot{V} \leq -\alpha V + K,$$

donde  $K := \sum_{i=1}^n (\sigma_i M_i^2 + v_i^{-2}/2a_i)$  y  $v_i^{-}$  es una cota superior para  $v_i$ ; de este modo, para  $V \geq V_0$ , considerando  $V_0 = K/\alpha$ , se tiene que  $dV/dt \leq 0$ , lo cual implica que  $V \in \mathcal{L}_\infty$ . Por lo tanto  $e_i, \varphi_i \in \mathcal{L}_\infty$ .

Para probar la segunda parte (b) del teorema, se parte de (2.21) para establecer que

$$\dot{V} \leq \sum_{i=1}^n \left( -a_i e_i^2 - e_i v_i \right),$$

y tomando en cuenta que  $(a_i e_i + v_i)^2 \geq 0$ , puede verificarse que se cumple esta desigualdad:

$$\dot{V} \leq \sum_{i=1}^n \left( -\frac{a_i}{2} e_i^2 + \frac{v_i^2}{2a_i} \right).$$

Integrando ambos lados de la expresión anterior se obtiene

$$V(t) - V(0) \leq \sum_{i=1}^n \left( -\frac{a_i}{2} \int_0^t e_i^2(\zeta) d\zeta + \frac{1}{2a_i} \int_0^t v_i^2(\zeta) d\zeta \right),$$

entonces es válida también la siguiente desigualdad:

$$V(t) - V(0) \leq -\frac{a_{\min}}{2} \int_0^t \|e(\zeta)\|^2 d\zeta + \frac{1}{2a_{\min}} \int_0^t \|v(\zeta)\|^2 d\zeta,$$

para la cual  $a_{\min} := \min\{a_i; i = 1, 2, \dots, n\}$ ; de esta manera,

$$\int_0^t \|e(\zeta)\|^2 d\zeta \leq \frac{2}{a_{\min}} [V(0) - V(t)] + \frac{1}{a_{\min}^2} \int_0^t \|v(\zeta)\|^2 d\zeta$$

cumple con la forma enunciada en el teorema:

$$\int_0^t \|e(\zeta)\|^2 d\zeta \leq \lambda + \mu \int_0^t \|v(\zeta)\|^2 d\zeta.$$

haciendo  $\lambda := (2/a_{\min}) \sup_{t \geq 0} [V(0) - V(t)]$  y  $\mu := 1/a_{\min}^2$ . Esto prueba la parte (b) del Teorema 2.4.1 y concluye su demostración.

En pocas palabras el teorema anterior establece que la ley de adaptación de pesos (2.17) garantiza que  $e_i$ , y  $\varphi_i$  permanecen acotados para todo  $i = 1, 2, \dots, n$ , y además, la “energía” del error de estado  $e(t)$  es proporcional a la “energía” del error de modelado  $v(t)$ . En el caso



especial de que el error de modelado sea cuadrado integrable, esto es,  $v \in \mathcal{L}_2$  entonces  $e(t)$  converge a cero asintóticamente.

**Observación 2.4.1.** Se observa que la modificación  $\sigma$  causa que la ley adaptable (2.17) sea discontinua; por lo tanto los resultados de existencia y unicidad de la solución de ecuaciones diferenciales son en general no aplicables. Para garantizar la existencia y unicidad de las soluciones, la trayectoria del comportamiento de  $w_i(t)$  puede hacerse “suave” sobre la hipersuperficie de discontinuidad  $\{w_i \in \mathcal{R}^L : |w_i| = M_i\}$  modificando la ley adaptable (2.17) de la siguiente manera:

$$\dot{w}_i = \begin{cases} -\Gamma_i z e_i & \text{si } \{ |w_i| < M_i \} \text{ o } \{ |w_i| = M_i \text{ y } w_i^T \Gamma_i z e_i > 0 \} \\ -\Gamma_i z e_i + \frac{w_i^T \Gamma_i z e_i}{w_i^T \Gamma_i w_i} \Gamma_i w_i & \text{si } \{ |w_i| = M_i \} \text{ y } \{ -\sigma_i w_i^T \Gamma_i w_i \leq w_i^T \Gamma_i z e_i \leq 0 \} \\ -\Gamma_i z e_i - \sigma_i \Gamma_i w_i & \text{si } \{ |w_i| > M_i \} \text{ o } \{ |w_i| = M_i \text{ y } -\sigma_i w_i^T \Gamma_i w_i > w_i^T \Gamma_i z e_i \} \end{cases} . \quad (2.23)$$

Se demuestra en [27], que esta ley adaptable conserva todas las propiedades de (2.17) y que, además, garantiza la existencia de una solución única, en el sentido de Caratheodory [30].

# Capítulo 3

## Modelo de manipuladores robóticos

*En este capítulo se exponen los modelos matemáticos de los manipuladores robóticos, para los cuales se desarrollan posteriormente los controladores descentralizados. Se describe en la introducción los elementos de un manipulador robótico y el procedimiento general de modelado. Se presentan las ecuaciones fundamentales para el modelado, el modelo de un manipulador robótico de dos grados de libertad y la descripción general del modelo utilizado en el caso del manipulador ANAT de siete grados de libertad; también, se describe el acotamiento de las dinámicas.*

### 3.1 Introducción

Un manipulador robótico rígido completamente actuado consiste en una secuencia de  $N$  número de cuerpos rígidos llamados elementos o eslabones (más la base del robot), conectados mediante  $N$  uniones o articulaciones, ya sean prismáticas o de revolución (giratorias), cada una con un actuador propio; cada par articulación-elemento constituye un grado de libertad. En esta tesis se omite la dinámica interna de los actuadores y se considera sólo sus salidas, fuerzas o torques, como entradas al modelo. De esta manera el manipulador está compuesto por  $N$  uniones y  $N$  eslabones, asociados cada uno de ellos a su respectiva unión. El modelo matemático se construye entonces con base en las variables de

las uniones, los parámetros físicos y geométricos de los eslabones y las fuerzas o torques aplicados a las uniones [9].

El modelo matemático del manipulador robótico, por el que queda descrito su comportamiento dinámico se obtiene, directa o indirectamente, de las leyes físicas de Newton. Se pueden aplicar sistemáticamente enfoques convencionales como las formulaciones de Euler-Lagrange y de Euler-Newton para desarrollar las ecuaciones de movimiento del manipulador, la primera está basada en la mecánica lagrangiana, mientras que la segunda parte directamente de la segunda ley de Newton. Independientemente del enfoque utilizado, se obtienen las ecuaciones dinámicas de movimiento para las distintas articulaciones del manipulador en términos de los parámetros geométricos e inerciales especificados para los distintos elementos y de las variables de articulación. Estas ecuaciones son útiles para efectos de simulación en computadora del movimiento del manipulador, el diseño de ecuaciones de control apropiadas para el manipulador y la evaluación del diseño y estructura cinemática [9], [17].

### 3.2 Ecuaciones fundamentales

Las ecuaciones de movimiento general de un manipulador se pueden expresar convenientemente mediante la aplicación directa de la formulación Euler-Lagrange a sistemas no conservativos [9], [17], para  $i = 1, 2, \dots, N$ :

$$\tau_i = \frac{d}{dt} \left( \frac{\partial L}{\partial \dot{q}_i} \right) - \frac{\partial L}{\partial q_i}$$

donde  $L$  es la función lagrangiana definida por  $L = K - P$ .  $K$  es la energía cinética total del manipulador,  $P$  la energía potencial total del manipulador,  $q_i$  la coordenada generalizada del manipulador y  $\dot{q}_i$  la primera derivada con respecto al tiempo de la coordenada generalizada  $q_i$ , y  $\tau_i$  la fuerza o par generalizado aplicado al sistema en la articulación  $i$  para mover el elemento  $i$ . Las coordenadas generalizadas se escogen convenientemente; en la

mayoría de los caso se hacen corresponder al ángulo de la articulación, si ésta es giratoria, o a la distancia recorrida por la articulación, si es prismática.

La energía cinética total está dada por:

$$K = \sum_{i=1}^n K_i = \sum_{i=1}^n \int dK_i = \sum_{i=1}^n \int \frac{1}{2} (\dot{x}_i^2 + \dot{y}_i^2 + \dot{z}_i^2) dm_i$$

donde  $\dot{x}_i, \dot{y}_i, \dot{z}_i$ , son las componentes de velocidad, en el sistema de coordenadas de la base, para cada partícula con masa diferencial  $dm_i$  del elemento número  $i$ .

La energía potencial total está dada por:

$$P = \sum_{i=1}^n P_i = \sum_{i=1}^n m_i g \bar{r}_{z_i}$$

donde  $\bar{r}_{z_i}$  es la componente en el eje vertical de la posición, en el sistema de coordenadas de la base, del centro de masa del elemento  $i$ ;  $m_i$  es la masa de tal elemento y  $g$  es la aceleración de la gravedad.

Finalmente resulta un modelo que puede ser simplificado y expresarse de la siguiente forma [9], [8], [19], [20], [25]:

$$\tau = M(q)\ddot{q} + V(q, \dot{q}) + F_g(q)$$

donde  $q, \dot{q}, \ddot{q} \in \mathfrak{R}^N$  son, respectivamente, posición, velocidad y aceleración de las  $N$  articulaciones,  $M(\cdot) \in \mathfrak{R}^{N \times N}$  es la matriz de inercia,  $\tau \in \mathfrak{R}^N$  es el vector que conformado por la fuerza o torque aplicado a cada una de las  $N$  articulaciones, y  $V(\cdot)$ ,  $F_g(\cdot) \in \mathfrak{R}^N$  son los términos correspondientes a fuerzas centrífugas y de Coriolis, y a fuerzas de gravedad, respectivamente. Pueden incluirse términos correspondientes a fuerzas de fricción [23], [24], [31], como en la siguiente ecuación:

$$\tau = M(q)\ddot{q} + V(q, \dot{q}) + F_f(\dot{q}) + F_g(q), \quad (3.1)$$



en la cual aparecen las fuerzas de fricción cinética en el término  $F_f(\cdot) \in \mathfrak{R}^N$ . Aún así este modelo no representa completamente al manipulador robótico, debido al conocimiento inexacto de los efectos de la fricción, la omisión de algunas dinámicas, y las variaciones e incertidumbres en los parámetros [8], [20], [33].

### 3.3 Modelo del manipulador robótico de dos articulaciones

Para efectos del desarrollo de esta tesis, se considera primero un modelo matemático de dos eslabones, con ambas articulaciones giratorias, con ejes de rotación horizontales, por lo que se incluye carga gravitacional, además de los términos correspondientes a la carga inercial, fuerzas de Coriolis y centrífuga [9], dado por:

$$\begin{aligned}
 \begin{bmatrix} \tau_1 \\ \tau_2 \end{bmatrix} &= \begin{bmatrix} \frac{1}{3}m_1l^2 + \frac{4}{3}m_2l^2 + m_2l^2 \cos(q_2) & \frac{1}{3}m_2l^2 + \frac{1}{2}m_2l^2 \cos(q_2) \\ \frac{1}{3}m_2l^2 + \frac{1}{2}m_2l^2 \cos(q_2) & \frac{1}{3}m_2l^2 \end{bmatrix} \begin{bmatrix} \ddot{q}_1 \\ \ddot{q}_2 \end{bmatrix} + \\
 &+ \begin{bmatrix} -\frac{1}{2}m_2l^2 \text{sen}(q_2) \dot{q}_2^2 - m_2l^2 \text{sen}(q_2) \dot{q}_1 \dot{q}_2 \\ \frac{1}{2}m_2l^2 \text{sen}(q_2) \dot{q}_1^2 \end{bmatrix} + \\
 &+ \begin{bmatrix} \frac{1}{2}m_1gl \cos(q_1) + \frac{1}{2}m_2gl \cos(q_2 + q_1) + m_2gl \cos(q_1) \\ \frac{1}{2}m_2gl \cos(q_2 + q_1) \end{bmatrix}
 \end{aligned} \tag{3.2}$$

con elementos de la misma longitud. En [19] se presenta otro modelo:

$$\begin{aligned}
 \begin{bmatrix} \tau_1 \\ \tau_2 \end{bmatrix} &= \begin{bmatrix} (m_1 + m_2)l_1^2 + m_2l_2^2 + 2m_2l_1l_2 \cos(q_2) & m_2l_2^2 + m_2l_1l_2 \cos(q_2) \\ m_2l_2^2 + m_2l_1l_2 \cos(q_2) & m_2l_2^2 \end{bmatrix} \begin{bmatrix} \ddot{q}_1 \\ \ddot{q}_2 \end{bmatrix} + \\
 &+ \begin{bmatrix} -m_2l_1l_2 \text{sen}(q_2) \dot{q}_2^2 - 2m_2l_1l_2 \text{sen}(q_2) \dot{q}_1 \dot{q}_2 \\ m_2l_1l_2 \text{sen}(q_2) \dot{q}_1^2 \end{bmatrix} + \\
 &+ \begin{bmatrix} (m_1 + m_2)gl_1 \cos(q_1) + m_2gl_2 \cos(q_2 + q_1) \\ m_2gl_2 \cos(q_2 + q_1) \end{bmatrix}
 \end{aligned} \tag{3.3}$$



el cual considera el caso en que las longitudes de los eslabones difieran entre sí. En (3.2) la longitud de los eslabones está representada por  $l$  y en (3.3) por  $l_1$  y  $l_2$ , para el primero y el segundo eslabón, respectivamente. En (3.2) y (3.3) las masas del primero y segundo eslabón están representadas por  $m_1$  y  $m_2$ , respectivamente;  $\tau_1$  y  $\tau_2$  representan los torques aplicados en la primera y en la segunda articulación, respectivamente; la posición, la velocidad y la aceleración angulares de la primera y la segunda articulación, están representadas por  $q_1, q_2, \dot{q}_1, \dot{q}_2, \ddot{q}_1, \ddot{q}_2$ .

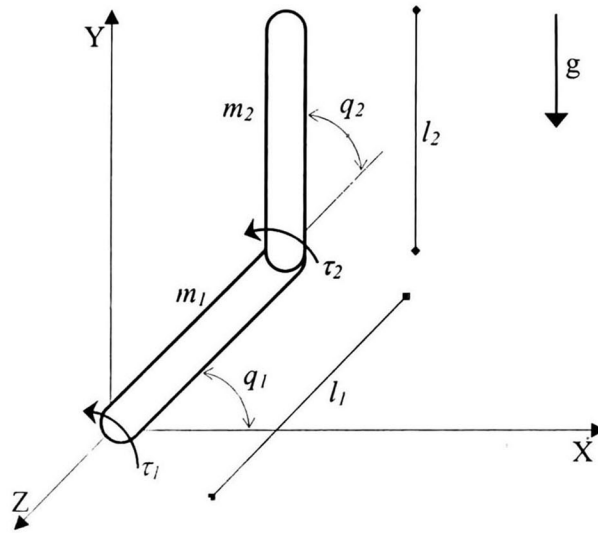


Figura 3.1. Manipulador robótico planar de dos articulaciones revolutivas.

En [14], [25] se usa también un modelo para longitudes distintas en sus dos eslabones, mientras que en [1] se usa un modelo de manipulador de masa y longitud iguales. Por otra parte en [1], [13], [14], [21], [22], [23] se utiliza un modelo en el que no hay fuerzas gravitacionales, suponiendo los dos ejes de rotación absolutamente verticales. En [24] el modelo considera eslabones con centro de masa ubicados en posiciones distintas del centro longitudinal. En [13] el modelo es variante en el tiempo. En base a estos antecedentes en la literatura consultada se considera que cualquiera de los dos modelos (3.2) y (3.3) es apropiado para los fines de la presente tesis.

### 3.4 Modelo del manipulador robótico de siete articulaciones

El manipulador robótico de siete articulaciones utilizado en esta tesis es conocido como robot hiperredundante ANAT (por sus siglas en inglés, “Articulated Nimble Adaptable Trunk”), consta de una articulación prismática y seis giratorias; las primeras tres de éstas son paralelas entre sí y las últimas tres perpendiculares entre sí, siendo éstas últimas las que conforman la muñeca del manipulador [23].

Un manipulador es cinemáticamente redundante cuando tiene un número de grados de libertad mayor al número de variables de unión mínimo necesario para realizar una tarea dada; de acuerdo a ello los manipuladores robóticos clasificados como redundantes, tienen la propiedad de poseer movimientos internos que no influyen en la trayectoria del efector final del manipulador y que le permite realizar tareas auxiliares tales como, salvar obstáculos y optimizar el gasto de energía del manipulador dentro de un medio ambiente difícil [23].

El manipulador robótico tipo ANAT que se considera en esta tesis es un modelo de un manipulador robótico real de la Escuela de Tecnología Superior de la Universidad de Québec (ETS-UQ), Montreal, Québec, Canadá. La Figura 3.2 muestra una fotografía de este robot; en la Figura 3.3 se presenta el dibujo correspondiente, donde se aprecian las articulaciones [23].

El modelo matemático de este manipulador robótico incluye las fuerzas inerciales, fuerzas de Coriolis y centrífugas, fuerzas de fricción y fuerzas gravitacionales, de acuerdo a la ecuación (3.1); en las simulaciones desarrolladas durante este trabajo de tesis se utiliza un programa computacional, ejecutable en Matlab®, que implementa el modelo matemático correspondiente. El programa contiene un módulo que calcula los valores de la matriz de inercia y otro módulo que calcula los valores de los vectores resultantes de las fuerzas de gravedad y de las fuerzas de Coriolis y centrífugas; las fuerzas de fricción se obtienen multiplicando el vector de velocidades por un coeficiente de fricción constante. El usuario del programa está habilitado para inicializar los siguientes parámetros del manipulador robótico: masa y longitud de cada eslabón, y coeficiente de fricción cinética en cada articulación.

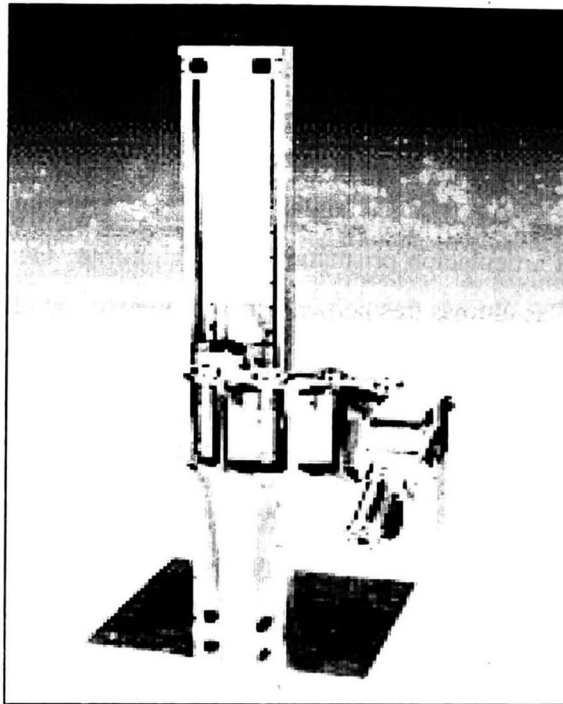


Figura 3.2. Fotografía del robot ANAT (cortesía de la ETS-UQ).

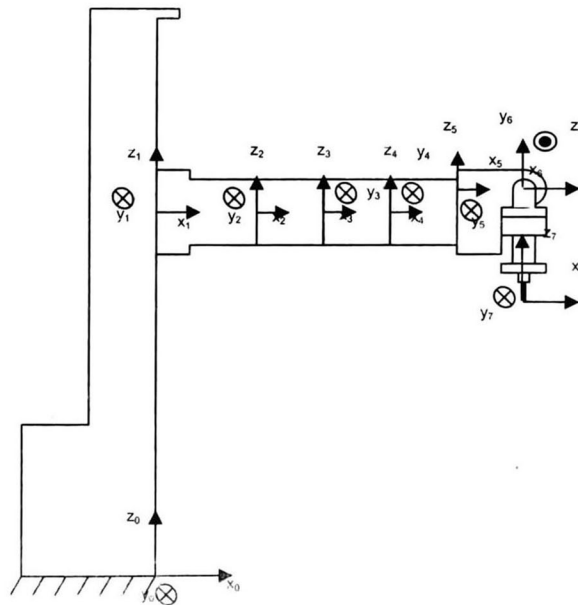


Figura 3.3. Dibujo esquemático del robot ANAT (cortesía de la ETS-UQ).

En la Figura 3.4 se muestra un dibujo que ilustra como el manipulador robótico sigue una trayectoria rectangular con el extremo de su efector final; es decir, definida en el espacio de trabajo, a la vez que utiliza su propiedad de redundante para salvar dos obstáculos de forma cilíndrica [23].

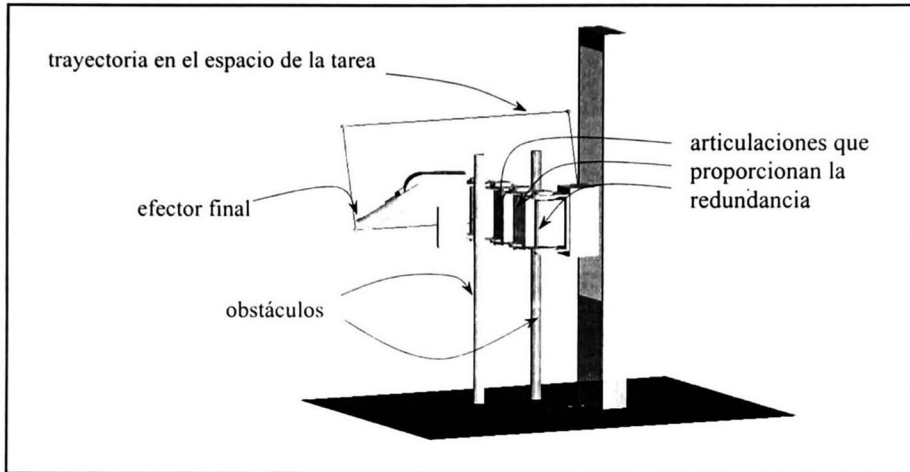


Figura 3.4. Dibujo del robot ANAT en un ambiente de trabajo (cortesía de la ETS-UQ).

### 3.5 Acotamiento de las dinámicas del modelo

Las dinámicas de los modelos matemáticos de los manipuladores robóticos resultan acotadas; este aspecto es importante para la posibilidad de utilizar un esquema neuronal de identificación.

De acuerdo al análisis de los modelos de dos eslabones y a la revisión de literatura que incluye este aspecto [8], [24], las dinámicas del robot están acotadas de la siguiente manera:

$$\begin{aligned}
 \|M(q)\ddot{q}\| &\leq c_m \|\ddot{q}\| \\
 \|V(q, \dot{q})\| &\leq c_v \|\dot{q}\|^2 \\
 \|F_f(\dot{q})\| &\leq c_f \|\dot{q}\| \\
 \|F_g(q)\| &\leq c_g
 \end{aligned}
 \tag{3.4}$$

donde  $c_m$ ,  $c_v$ ,  $c_f$  y  $c_g$ , son valores constantes reales positivos. Estos acotamientos pueden ser obtenidos fácilmente en los casos particulares de los modelos de manipuladores robóticos de dos uniones. Puede verse, por ejemplo, en (3.6) y (3.7) que el valor de  $c_m$  está dado por el valor de  $\|M(\mathbf{q})\|$  cuando  $q_2$  toma cualquier valor del conjunto  $\{\dots, -2\pi, -\pi, 0, \pi, 2\pi, \dots\}$ ; es decir, cuando el manipulador está en completa extensión. Esto corresponde a que la inercia es mayor a causa de que los centros de masa están en su máxima distancia;  $c_v$ , por su parte, es calculado con el valor de  $\|V(\cdot)\|$  cuando  $q_2$  toma cualquier valor del conjunto  $\{\dots, -3\pi/2, -\pi/2, \pi/2, 3\pi/2, \dots\}$ , es decir, cuando los eslabones están en perpendicularidad, situación en la que los efectos centrífugos y de Coriolis del movimiento de un eslabón sobre otro son mayores para cada par de velocidades. Finalmente, en cuanto a  $c_f$  su valor corresponde a los valores de  $\|F_g(\mathbf{q})\|$  cuando el brazo manipulador está completamente extendido y además en posición horizontal, cuando ambos valores, el de  $q_1$  y el de  $q_2$  son elementos del conjunto  $\{\dots, -2\pi, -\pi, 0, \pi, 2\pi, \dots\}$ . Por otra parte, al considerar la fuerza de fricción cinética se comporta linealmente con respecto a la velocidad, el valor de  $c_f$  corresponde directamente al coeficiente de fricción máximo.

### 3.6 Observaciones importantes para el control de manipuladores

El objetivo del control de un manipulador robótico basado en computadora es mantener la respuesta dinámica del mismo de acuerdo a especificaciones previas y a los objetivos deseados. En general el rendimiento de un manipulador depende directamente de la eficacia de los algoritmos de control [9]. El problema de control consiste en obtener modelos dinámicos del manipulador físico y a continuación especificar leyes o estrategias de control correspondientes para conseguir la respuesta y rendimiento del sistema deseado [9].

La obtención del modelo dinámico del manipulador basado en la formulación Euler-Lagrange es simple y sistemática. Suponiendo el movimiento del cuerpo rígido, las ecuaciones de movimiento resultantes, excluyendo la dinámica de los dispositivos de control electrónico, huelgo y el rozamiento de los engranes, son un conjunto de ecuaciones diferenciales no lineales acopladas de segundo orden, como se observa en las secciones anteriores en el presente capítulo. En [9], se afirma que las ecuaciones de movimiento



dinámico para un manipulador tipo Stanford de seis articulaciones son altamente no lineales y constan de carga inercial, fuerzas de reacción de acople entre las articulaciones (Coriolis y centrífuga) y efectos de carga de la gravedad. Más aún, estos pares/fuerzas dependen de los parámetros físicos del manipulador, de la configuración instantánea de las articulaciones, de la velocidad y aceleración de las articulaciones y de la carga que está soportando el manipulador. Utilizando las ecuaciones de movimiento generalizadas de D'Alembert, que expresan explícitamente todas las interacciones y fuerzas de reacción de acople, se pueden calcular el predominio de los efectos traslacionales y rotacionales para cada punto de referencia a lo largo de una trayectoria específica; de esta manera los términos o elementos menos dominantes se pueden desprestigiar al calcular las ecuaciones de movimiento mecánico del manipulador. Tal simplificación depende de la trayectoria específica que se considere. El modelo simplificado resultante retiene la mayoría de las interacciones y fuerzas/pares de reacción y acople con un tiempo de cálculo reducido [9]. Las tareas anteriores, dirigidas a obtener un modelo exacto, o bien un modelo inexacto pero aceptable, pueden ser muy laboriosas y probablemente ineficaces si los requisitos de control incluyen mayor precisión o robustez.

De acuerdo a como se establece en [19], si se diseña un procedimiento de control basado en que el modelo dinámico del robot es conocido, al existir una gran cantidad de incertidumbres el método probablemente fallará. Además la mayor parte de los controladores son generalmente diseñados suponiendo un conocimiento exacto de la estructura del modelo y no incluyen fricción no lineal, vibraciones y otras incertidumbres del sistema. En un medio ambiente real las inevitables incertidumbres del modelo, que resultan de la dificultad de medir satisfactoriamente la fricción, cambios en la carga, etc., presentan dificultades adicionales a la tarea del controlador [8]. En [14], se menciona que en situaciones prácticas existen incertidumbres inevitables en el modelo del manipulador, como cambios en la carga, lo cual aunado a las altas no linealidades de sus dinámicas, hace muy difícil la tarea de control. En [20], se afirma que además de las dinámicas altamente no lineales y de acoplamiento, las propiedades inerciales a menudo desconocidas de los objetos manipulados hacen difícil realizar seguimiento de trayectoria. En [13], se menciona que en muchas situaciones reales parámetros como la masa de la carga o de los elementos del manipulador pueden ser variantes en el tiempo. En [31], se menciona que existen

incertidumbres muy importantes estructuradas y no estructuradas en los modelos matemáticos que se utilizan para manipuladores robóticos. En [25], se menciona que debido a las dinámicas no lineales, a los acoplamientos y a las dinámicas variantes en el tiempo, especialmente cuando los eslabones están moviéndose simultáneamente y a altas velocidades, el control basado en el modelo decrece su desempeño y se aumenta el error de seguimiento.

Los manipuladores robóticos son una clase de sistemas pasivos; la energía total que posee el robot en un instante determinado es, siempre, menor a la energía que le ha sido suministrada hasta ese instante por las fuerzas y/o torques aplicados en las articulaciones. Este concepto es importante para fines de control por que permite, mediante los efectos disipativos, llevar al sistema a un punto de mínima energía, y es entonces posible colocar al sistema en el punto deseado manipulando a través del control el estado de mínima energía para que coincida con el estado de referencia.

El acotamiento de los distintos términos en el modelo dinámico de acuerdo a (3.4), junto con la propiedad de pasividad que poseen los manipuladores robóticos, ofrece la posibilidad de que el sistema pueda ser aproximados por un modelo tipo red neuronal y la de construir un controlador, considerando las dinámicas de acoplamiento involucradas en dichos término como disturbios del subsistema (red neuronal local), para llevar a las articulaciones hacia sus referencias.

# Capítulo 4

## Identificación neuronal descentralizada de manipuladores robóticos

*En este capítulo se expone el esquema de identificación neuronal descentralizada propuesto para llevar a cabo la aproximación en línea de la dinámica de manipuladores robóticos; se presentan las ecuaciones de diferentes redes neuronales que se distinguen entre sí por el número de neuronas y el orden de sus interconexiones, y se muestran los resultados de simulación de la identificación de estados de un manipulador robótico de dos eslabones y de un robot ANAT de siete eslabones.*

### 4.1 Introducción

El esquema de identificación neuronal que se propone en esta tesis está basado en el esquema general de identificación de sistemas no lineales con redes neuronales recursivas de alto orden. Se considera de manera especial el caso de primer orden, con las leyes de aprendizaje y con las propiedades de aproximación, convergencia y robustez que se exponen en el capítulo dos de este documento. Como se menciona en el capítulo anterior, el modelo matemático simplificado de un manipulador robótico puede ser considerado como:

$$\tau = M(q)\ddot{q} + V(q, \dot{q}) + F_f(\dot{q}) + F_g(q), \quad (4.1)$$



donde  $q, \dot{q}, \ddot{q} \in \mathbb{R}^N$  son, respectivamente, posición, velocidad y aceleración de las  $N$  articulaciones,  $M(\cdot) \in \mathbb{R}^{N \times N}$  es la matriz de inercia,  $\tau \in \mathbb{R}^N$  es la fuerza o torque aplicado a las  $N$  articulaciones, y  $V(\cdot)$ ,  $F_f(\cdot) \in \mathbb{R}^N$ ,  $F_g(\cdot) \in \mathbb{R}^N$  son los términos correspondientes a los efectos centrífugos y de Coriolis, fuerzas de fricción cinética y a fuerzas por gravedad, respectivamente.

El esquema de identificación que se propone, por ser descentralizado como se explica más adelante, hace caso omiso de las interacciones entre las articulaciones aún cuando se sabe que las dinámicas de acoplamiento pueden ser muy importantes en los manipuladores robóticos. Sin embargo se aprovecha el hecho de que estas dinámicas cumplen con la condición de acotamiento expresada en (3.4). Esto implica que cada una de las redes neuronales locales identificando la dinámica total de una articulación del manipulador robótico puede considerar los efectos de acoplamiento incluidos en cada uno de los términos como perturbaciones acotadas. Entonces desde el punto de vista descentralizado la red local aproxima la dinámica local incluyendo dichas perturbaciones.

Para aplicar un enfoque descentralizado se supone que (4.1) puede dividirse en  $N$  subsistemas, cada uno formulado como:

$$\ddot{q}_i = f_i(q_i, \dot{q}_i, \tau_i) + \Psi_i(q_1, q_2, \dots, q_N, \dot{q}_1, \dot{q}_2, \dots, \dot{q}_N, \tau_1, \tau_2, \dots, \tau_N), \quad (4.2)$$

donde  $i=1,2,\dots,N$ ,  $f_i(\cdot)$  depende sólo de las variables locales, específicamente de la posición local de la articulación, de su velocidad y de la fuerza o torque aplicado por su actuador;  $\Psi_i(\cdot)$  representa los efectos de todas interconexiones sobre el subsistema  $i$ , y es una función dependiente de las variables de todos los subsistemas o articulaciones.

## 4.2 Modelo neuronal para identificación descentralizada

Para desarrollar la identificación descentralizada del modelo dinámico de un manipulador robótico de  $N$  articulaciones, se propone una colección de  $N$  redes neuronales independientes entre sí. Cada red neuronal identifica a un subsistema como (4.2) en función de variables locales únicamente; es decir, en función de la posición, la velocidad y el torque

o fuerza de entrada de la articulación correspondiente, omitiendo los efectos de las interconexiones, que son absorbidos por la dinámica del modelo de red neuronal descentralizada, considerándolos como perturbaciones externas.

En la siguiente figura se ilustra, de forma esquemática, la identificación neuronal descentralizada del manipulador robótico. El sistema real, conformado por un conjunto de  $N$  subsistemas fuertemente interconectados, es identificado por un modelo de  $N$  redes neuronales completamente independientes entre sí.

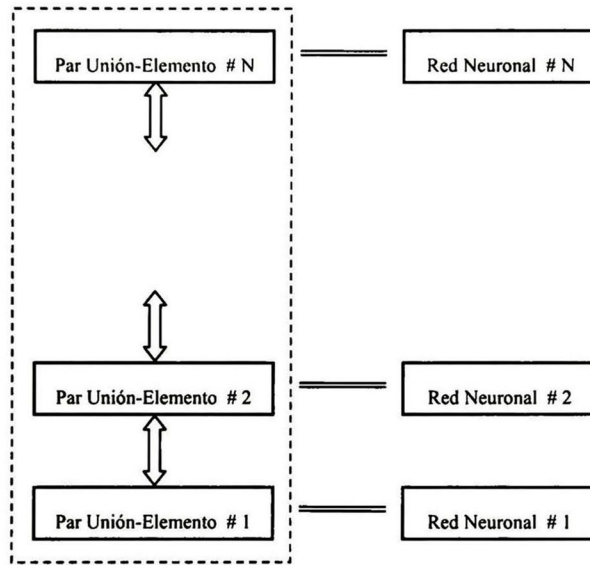


Figura 4.1. Esquema de identificación neuronal descentralizada del manipulador robótico, sólo con variables locales; las interconexiones, representadas por flechas, no existen en el modelo neuronal.

La estructura del modelo de red neuronal utilizada corresponde a la topología de conexión serie-paralelo:

$$\dot{x}_i = A_i x_i + W_i^T \rho_i(x_i, u_i), \quad (4.3)$$

donde  $x_i \in \mathcal{R}^2$  es el vector de estados de la  $i$ -ésima red neuronal,  $x_i \in \mathcal{R}^2$  es el vector de estados de la  $i$ -ésima articulación del modelo real del manipulador robótico, cuyos elementos son la posición y la velocidad de la articulación,  $u_i \in \mathcal{R}$  es la fuerza o torque aplicado a la  $i$ -ésima articulación,  $A_i \in \mathcal{R}^{2 \times 2}$  es una matriz diagonal estable,  $W_i$  es la matriz



de pesos ajustables,  $\rho_i(\chi_i, u_i)$  es la matriz de interconexiones,  $i=1,2,\dots,N$ . La matriz  $W_i$  es conformada por dos vectores columna, cuyos valores son pesos ajustables; uno está asociado a las interconexiones que entran a la primera neurona, la cual identifica a la posición de la articulación y el otro está asociado a las interconexiones que entran a la segunda neurona, la cual identifica a la velocidad de la articulación. La matriz  $\rho_i(\chi_i, u_i)$  es conformada por dos vectores columna cuyos elementos corresponden a los valores de las interconexiones que entran a una y otra neurona, respectivamente; si la interconexión es de primer orden dicho valor es la salida de una función sigmoideal, directamente, y en el caso de interconexiones de alto orden, es un producto de las salidas de las funciones sigmoideales correspondientes.

Dado que se pretende desarrollar identificación estrictamente descentralizada, el modelo (4.3) no incluye los términos de interconexión entre subsistemas. Para considerarlos deberían sumarse términos diseñados explícitamente para la aproximación de los efectos de las interconexiones, pero que requieren información de otros subsistemas; por esta razón el esquema propuesto no puede realizar una identificación directa de las dinámicas de interconexión. El modelo neuronal, considerando las interconexiones, tendría la forma:

$$\dot{x}_i = A_i x_i + W_i^T \rho_i(\chi_i, u_i) + W_i'^T \rho_i'(\chi_1, \chi_2, \dots, \chi_N, u_1, u_2, \dots, u_N),$$

donde  $W_i'^T$  es una matriz de pesos sinápticos relacionados con la identificación de los efectos de las interconexiones, y  $\rho_i'(\cdot)$  es una matriz cuyos elementos son funciones de activación sigmoideales asociadas a entradas provenientes de las interconexiones. Este modelo puede ser reemplazado por el modelo (4.3), bajo la suposición de que todo valor en el espacio correspondiente a las dinámicas de interconexión está contenido en el espacio de estados de (4.3).

El valor de cada uno de los pesos en la matriz  $W_i = [w_{i1}, w_{i2}]$  se ajusta de acuerdo a la siguiente ley:

$$\dot{w}_{ijk} = -\gamma_{ij} \rho_{ijk}(\chi_i) e_{ij},$$

donde  $i=1,2,\dots,N$ ,  $j=1,2$ ,  $k=1,2,\dots,L_{ij}$ , siendo  $L_{ij}$  el número de interconexiones que entran a

la  $j$ -ésima neurona de la  $i$ -ésima red neuronal (subsistema),  $\gamma_{ij}$  es un número real positivo, que representa el índice de aprendizaje y  $e_{ij}$  es el error de identificación del estado  $j$  de la articulación  $i$ , definido como  $e_{ij} = x_{ij} - \chi_{ij}$ .

El modelo (4.3) se modifica para obtener un modelo afin en el control, sin que esto afecte las propiedades del modelo [28], [29]. Se obtiene la estructura de red neuronal de dos estados:

$$\begin{aligned}\dot{x}_{i1} &= -a_{i1}x_{i1} + \mathbf{w}_{i1}^T \boldsymbol{\rho}_{i1}(\chi_i) + x_{i2} \\ \dot{x}_{i2} &= -a_{i2}x_{i2} + \mathbf{w}_{i2}^T \boldsymbol{\rho}_{i2}(\chi_i) + u_i\end{aligned}\tag{4.4}$$

donde  $i = 1, 2, \dots, N$ ,  $x_{i1}$  y  $x_{i2}$  son los estados (neuronas) de la  $i$ -ésima red neuronal, que identifican respectivamente a  $\chi_{i1}$  y  $\chi_{i2}$ , posición y velocidad de la  $i$ -ésima articulación;  $a_{i1}$ ,  $a_{i2}$  son constantes reales positivos,  $\boldsymbol{\rho}_{ij}(\chi_i)$  es el vector de interconexiones preprocesadas por su función de activación sigmoideal, posiblemente de alto orden, que entran a la  $i$ -ésima neurona, con su correspondiente vector de pesos sinápticos  $\mathbf{w}_{ij}$ , y  $u_i$  es la fuerza o torque aplicado a la  $i$ -ésima articulación.

Para evitar que en algún momento los valores absolutos de los pesos sinápticos puedan incrementarse desproporcionadamente, particularmente cuando se usan redes neuronales realmente masivas o cuando desafortunadamente los parámetros fijos de las redes neuronales, como los coeficientes  $a_{ij}$  y  $\gamma_{ij}$ , han sido inconvenientemente elegidos, debe emplearse una modificación robusta de la ley de aprendizaje de acuerdo a (2.17) o (2.23).

Con base en el modelo (4.4) se construyen modelos particulares de redes neuronales en dos enfoques distintos que sirven para realizar identificación de la dinámica de un manipulador robótico de dos eslabones; la diferencia fundamental es que en el primero se identifica solamente un estado (la velocidad), aprovechando el hecho de que el primer estado (la posición) es obtenido calculando la integral del segundo estado; mientras que el segundo enfoque está pensado para superar la diferencia en la identificación de la posición cuando las condiciones iniciales de la red difieren de las del manipulador robótico. En realidad este segundo enfoque sirve más que nada para fines didácticos en el problema de identificación de manipuladores robóticos que se trata en esta tesis, pues se supone que tanto la posición y la velocidad reales están disponibles, y además la topología de redes neuronales utilizada es serie-paralelo.

## 4.3 Esquema de identificación de la dinámica de la velocidad

Este esquema de red neuronal es diseñado con el propósito de llevar a cabo la identificación del modelo dinámico de la velocidad de cada articulación; se toma en cuenta, por obviedad, que la dinámica de la posición queda determinada al suponer que la posición real y su derivada, o sea la velocidad real, están siempre disponibles. Este esquema corresponde a un caso especial del modelo (4.4) haciendo  $a_{ij} = 0$  y eliminando toda interconexión en la primera neurona, el modelo resulta así:

$$\begin{aligned}\dot{x}_{i1} &= x_{i2} \\ \dot{x}_{i2} &= -a_{i2}x_{i2} + \mathbf{w}_{i2}^T \boldsymbol{\rho}_{i2}(\chi_i) + u_i\end{aligned}$$

Para determinar los efectos de usar distintos arreglos de interconexiones y del orden de las mismas, primeramente se ensayan diferentes modelos de red neuronal recurrente con interconexiones de primer orden: un modelo con una única neurona y luego un modelo de dos neuronas. Posteriormente, se utiliza un modelo con interconexiones de segundo orden, y finalmente un modelo de red neuronal recurrente de alto orden.

En las secciones siguientes se prefiere emplear  $q_i$  en lugar de  $\chi_{i1}$  y  $\dot{q}_i$  en lugar de  $\chi_{i2}$ , que son los estados de cada subsistema a identificar. La entrada  $u_i$  en cada red neuronal, corresponde a la fuerza o torque  $\tau_i$ , aplicado a la articulación correspondiente.

### 4.3.1 Modelo con una interconexión de primer orden

Se emplea una sola interconexión en la única neurona de la red neuronal para identificar la dinámica de la velocidad angular en cada una de las articulaciones. Se elige la interconexión con la que se obtienen mejores resultados en la identificación, la proveniente de la velocidad de la articulación.

Aunque sólo se requiere de un solo peso sináptico y una sola interconexión, se denominan  $w_{i22}$  y  $\rho_{i22}(\cdot)$  por consistencia con posteriores modelos en los que aparecen mayor número



de interconexiones. Las ecuaciones de las variables de estado son:

$$\begin{aligned} \dot{x}_{i1} &= x_{i2} \\ \dot{x}_{i2} &= -a_{i2}x_{i2} + w_{i22}\rho_{i22}(\dot{q}_i) + u_i \end{aligned} \quad (4.5)$$

La función de activación empleada en este modelo, y en todos los demás es  $\tanh(\cdot)$ , en este caso resulta:

$$\rho_{i22}(\dot{q}_i) = \tanh(\dot{q}_i).$$

El error de identificación es:

$$e_{i2} = x_{i2} - \dot{q}_i,$$

y la ley de aprendizaje:

$$\dot{w}_{i22} = -\gamma_{i22}e_{i2}\rho_{i22}(\dot{q}_i).$$

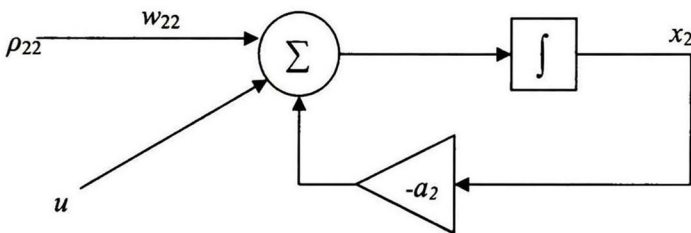


Figura 4.2. Esquema representativo de la neurona  $x_2$  en cada red neuronal del modelo (4.5).

### 4.3.2 Modelo con dos interconexiones de primer orden

Se emplean dos interconexiones para identificar la dinámica de la velocidad en cada una de las articulaciones; se usan los dos estados de la articulación en las interconexiones.

Las ecuaciones de variables de estado, funciones de activación, error de identificación y ley de aprendizaje son las siguientes:

$$\begin{aligned}\dot{x}_{i1} &= x_{i2} \\ \dot{x}_{i2} &= -a_{i2}x_{i2} + w_{i21}\rho_{i21}(q_i) + w_{i22}\rho_{i22}(\dot{q}_i) + u_i\end{aligned}\tag{4.6}$$

$$\rho_{i21}(q_i) = \tanh(q_i)$$

$$\rho_{i22}(\dot{q}_i) = \tanh(\dot{q}_i)$$

$$e_{i2} = x_{i2} - \dot{q}_i$$

$$\dot{w}_{i21} = -\gamma_{i21}e_{i2}\rho_{i21}(q_i)$$

$$\dot{w}_{i22} = -\gamma_{i22}e_{i2}\rho_{i22}(\dot{q}_i)$$

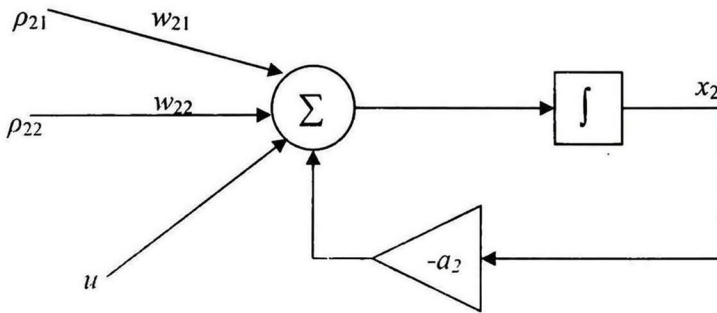


Figura 4.3. Esquema representativo de la neurona  $x_2$  en cada red neuronal del modelo (4.6).

### 4.3.3 Modelo con tres interconexiones, una de segundo orden

Se emplean tres interconexiones para identificar la dinámica de la velocidad en cada una de las articulaciones; se usan los dos estados de la articulación en las interconexiones, agregando como tercera interconexión un producto de segundo orden de funciones de activación.

El modelo neuronal queda descrito por:

$$\begin{aligned}\dot{x}_{i1} &= x_{i2} \\ \dot{x}_{i2} &= -a_{i2}x_{i2} + w_{i21}\rho_{i21}(q_i) + w_{i22}\rho_{i22}(\dot{q}_i) + w_{i23}\rho_{i23}(q_i)\rho_{i23}(\dot{q}_i) + u_i\end{aligned}\tag{4.7}$$

$$\rho_{i21}(q_i) = \tanh(q_i)$$

$$\rho_{i22}(\dot{q}_i) = \tanh(\dot{q}_i)$$

$$\rho_{i23}(q_i, \dot{q}_i) = \rho_{i21}(q_i)\rho_{i22}(\dot{q}_i)$$



$$e_{i2} = x_{i2} - \dot{q}_i$$

$$\dot{w}_{i21} = -\gamma_{i21} e_{i2} \rho_{i21}(q_i)$$

$$\dot{w}_{i22} = -\gamma_{i22} e_{i2} \rho_{i22}(\dot{q}_i)$$

$$\dot{w}_{i23} = -\gamma_{i23} e_{i2} \rho_{i23}(q_i, \dot{q}_i)$$

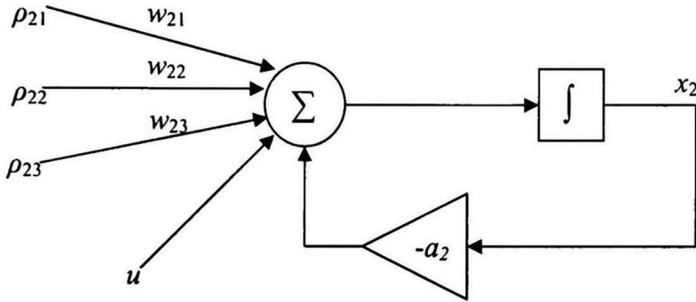


Figura 4.4. Esquema representativo de la neurona  $x_2$  en cada red neuronal del modelo (4.7).

#### 4.3.4 Modelo con interconexiones de alto orden

Se emplean cuatro interconexiones en la red neuronal para identificar la dinámica de la velocidad en cada una de las articulaciones; se usan los dos estados de la articulación en las interconexiones, la tercera y la cuarta interconexiones son de alto orden.

$$\dot{x}_{i1} = x_{i2}$$

$$\dot{x}_{i2} = -a_{i2}x_{i2} + w_{i21}\rho_{i21}(q_i) + w_{i22}\rho_{i22}(\dot{q}_i) + w_{i23}\rho_{i23}(q_i, \dot{q}_i) + w_{i24}\rho_{i24}(q_i, \dot{q}_i) + u_i \quad (4.8)$$

$$\rho_{i21}(q_i) = \tanh(q_i)$$

$$\rho_{i22}(\dot{q}_i) = \tanh(\dot{q}_i)$$

$$\rho_{i23}(\dot{q}_i) = \rho_{i22}^3(\dot{q}_i)$$

$$\rho_{i24}(q_i, \dot{q}_i) = \rho_{i21}^2(q_i) \rho_{i22}^2(\dot{q}_i)$$

$$e_{i2} = x_{i2} - \dot{q}_i$$

$$\begin{aligned}\dot{w}_{i21} &= -\gamma_{i21} e_{i2} \rho_{i21}(q_i) \\ \dot{w}_{i22} &= -\gamma_{i22} e_{i2} \rho_{i22}(\dot{q}_i) \\ \dot{w}_{i23} &= -\gamma_{i23} e_{i2} \rho_{i23}(q_i, \dot{q}_i) \\ \dot{w}_{i24} &= -\gamma_{i24} e_{i2} \rho_{i24}(q_i, \dot{q}_i)\end{aligned}$$

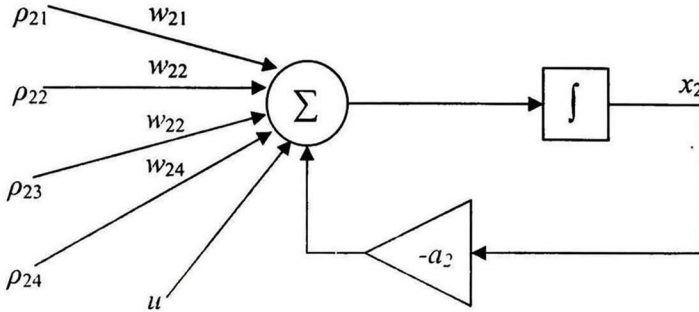


Figura 4.5. Esquema representativo de la neurona  $x_2$  en cada red neuronal del modelo (4.8).

## 4.4 Esquema de identificación de la posición y de la velocidad

Este esquema de red neuronal es diseñado con el propósito de llevar a cabo la identificación del modelo dinámico de la posición y la velocidad de cada articulación. Aún cuando lo anterior puede representar una desventaja, este esquema tiene a favor que puede ser utilizado para el caso cuando las condiciones iniciales del modelo real y del modelo neuronal son diferentes. Para este esquema se presentan dos ensayos, el primero usa solamente una neurona para identificación de la posición y una para la identificación de la velocidad, mientras que el segundo usa dos neuronas para identificación de velocidad. En una sección posterior, en la que se muestran resultados de simulación, puede observarse que no existe mejora que justifique el empleo de redes de alto orden, al menos del tipo de red neuronal utilizado y bajo la observación de que es un esquema de identificación que no usa arreglos masivos de neuronas. Se evita el uso de gran número de neuronas con la finalidad de obtener un esquema computacional para implementarse en tiempo real; en otro sentido la implementación de aprendizaje en línea efectúa una aproximación de la dinámica en cada instante de tiempo y puede prescindirse de la realización de un mapeo global de la dinámica del sistema.

#### 4.4.1 Modelo con dos neuronas, caso 1

Se emplea una sola interconexión en la segunda neurona, para identificar la dinámica de la velocidad en cada una de las articulaciones. El modelo empleado es el siguiente:

$$\begin{aligned}\dot{x}_{i1} &= -a_{i1}x_{i1} + w_{i11}\rho_{i11}(q_i) + x_{i2} \\ \dot{x}_{i2} &= -a_{i2}x_{i2} + w_{i22}\rho_{i22}(\dot{q}_i) + u_i\end{aligned}\quad (4.9)$$

$$\rho_{i11}(q_i) = \tanh(q_i)$$

$$\rho_{i22}(\dot{q}_i) = \tanh(\dot{q}_i)$$

$$e_{i1} = x_{i1} - q_i$$

$$e_{i2} = x_{i2} - \dot{q}_i$$

$$\dot{w}_{i11} = -\gamma_{i11}e_{i1}\rho_{i11}(q_i)$$

$$\dot{w}_{i22} = -\gamma_{i22}e_{i2}\rho_{i22}(\dot{q}_i)$$

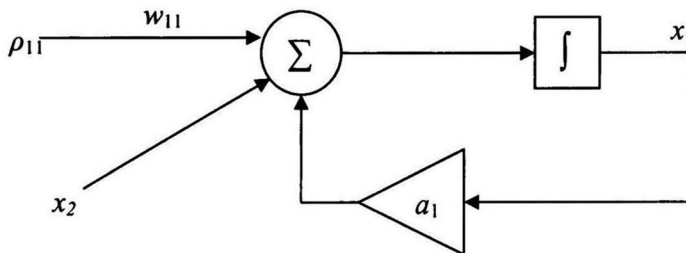


Figura 4.6. Esquema representativo de la primera neurona en cada red neuronal del modelo (4.9).

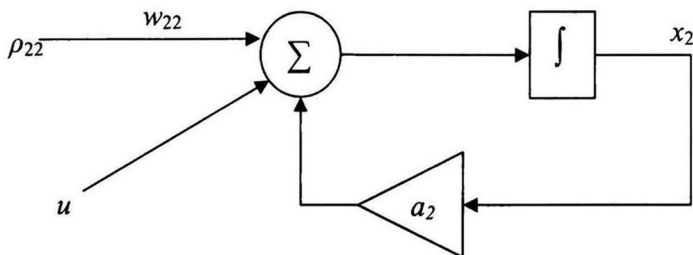


Figura 4.7. Esquema representativo de la segunda neurona en cada red neuronal del modelo (4.9).

## 4.4.2 Modelo con dos neuronas, caso 2

Se emplea dos interconexiones en la segunda neurona, una desde la posición real y otra desde la velocidad real, para identificar la dinámica de la velocidad en cada una de las articulaciones. El modelo empleado es el siguiente:

$$\dot{x}_{i1} = -a_{i1}x_{i1} + w_{i11}\rho_{i11}(q_i) + x_{i2} \quad (4.10)$$

$$\dot{x}_{i2} = -a_{i2}x_{i2} + w_{i21}\rho_{i21}(q_i) + w_{i22}\rho_{i22}(\dot{q}_i) + u_i$$

$$\rho_{i11}(q_i) = \tanh(q_i)$$

$$\rho_{i21}(q_i) = \tanh(q_i)$$

$$\rho_{i22}(\dot{q}_i) = \tanh(\dot{q}_i)$$

$$e_{i1} = x_{i1} - q_i$$

$$e_{i2} = x_{i2} - \dot{q}_i$$

$$\dot{w}_{i11} = -\gamma_{i11}e_{i1}\rho_{i11}(q_i)$$

$$\dot{w}_{i21} = -\gamma_{i21}e_{i2}\rho_{i21}(q_i)$$

$$\dot{w}_{i22} = -\gamma_{i22}e_{i2}\rho_{i22}(\dot{q}_i)$$

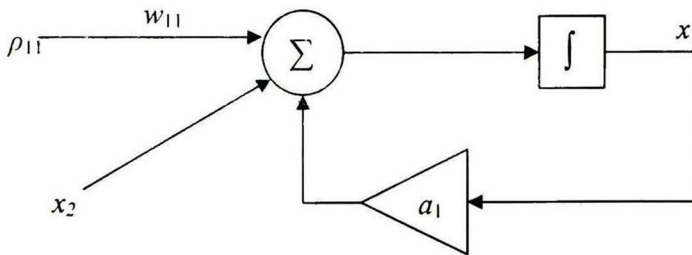


Figura 4.8. Esquema representativo de la primera neurona en cada red neuronal del modelo (4.10).

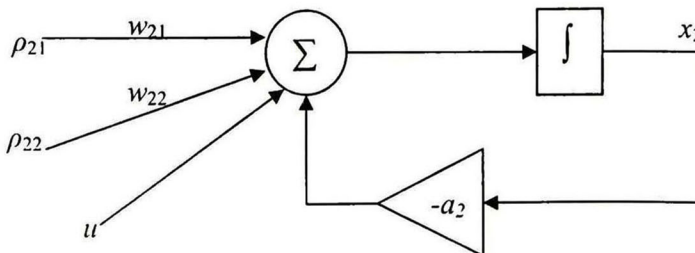


Figura 4.9. Esquema representativo de la segunda neurona en cada red neuronal del modelo (4.10).



## 4.5 Identificación del manipulador robótico de dos uniones

Los modelos neuronales propuestos en este capítulo, para la identificación descentralizada de manipuladores robóticos, se prueban, a través de simulaciones, como alternativas de solución al problema de identificación del sistema (3.2), el manipulador robótico de dos uniones revolativas, descrito en el capítulo anterior.

Las pruebas consisten en simular computacionalmente el comportamiento simultáneo del manipulador robótico y del modelo neuronal en el esquema de identificación descentralizada. La misma entrada es aplicada a ambos modelos y el identificador ajusta sus parámetros a partir del error en la identificación de los estados, tal como se representa en Figura 4.10.

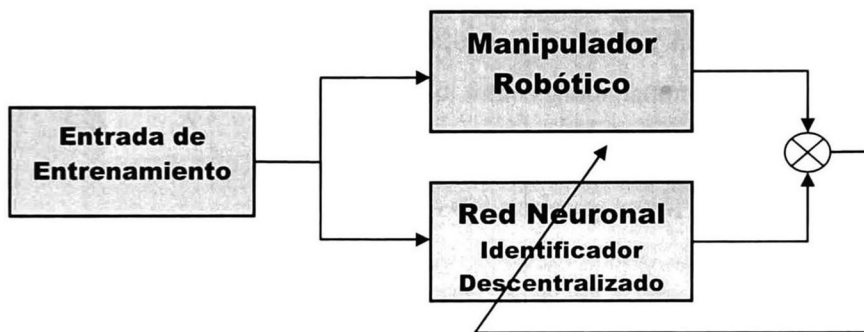


Figura 4.10. Esquema de identificación descentralizada.

### 4.5.1 Evaluación experimental

Para llevar a cabo la evaluación experimental las simulaciones son implementadas en MATLAB<sup>®</sup> y Simulink<sup>®</sup> (marcas registradas de MathWorks, Inc.). Los resultados de las pruebas son evaluados mediante gráficas de comportamiento del error, de trayectoria de los estados y de evolución de los pesos.

El modelo matemático (3.2), del manipulador robótico es implementado a través de un programa en MATLAB<sup>®</sup>; cada modelo neuronal es implementado en Simulink<sup>®</sup> y el esquema completo se ejecuta en Simulink<sup>®</sup>. Los valores de los parámetros del manipulador

robótico son  $m_1 = 1 \text{ Kg}$ ,  $m_2 = 1 \text{ Kg}$ ,  $l = 1 \text{ m}$ . Para las redes neuronales los valores de parámetros fijos empleados son  $a_{ij} = 1000$ , para cada neurona,  $\gamma_{jk} = 10^6$ , para cada peso sináptico. El índice de aprendizaje se selecciona con un valor muy alto para asegurar que las redes neuronales ajusten sus pesos con una velocidad suficientemente alta, ante entradas de entrenamiento con frecuencias muy altas, para poder alcanzar errores de identificación en milésimas de unidad. Con valores mucho menores para  $\gamma_{jk}$ , y su vez para  $a_{ij}$ , las redes neuronales también se comportan aceptablemente ante estas entradas, aunque el error de identificación, obviamente, es mayor.

Primero se evalúan los modelos (4.5), (4.6), (4.7) y (4.8), que corresponden al esquema de identificación de la dinámica de la velocidad, descrito en la sección 4.3. El propósito es probar la efectividad de cada uno de los modelos y compararlos entre sí. Posteriormente, con el mismo propósito, se evalúan los modelos (4.9) y (4.10), que son del esquema de identificación de la posición y de la velocidad, expuesto en la sección 4.4. Los resultados permiten además realizar comparaciones entre los dos esquemas.

#### **4.5.1.1 Descripción de los experimentos**

Para todos los casos se aplican las mismas entradas producidas por un generador de señales senoidales de frecuencia variable (“chirp”); la amplitud de dichas señales de entrada tienen valores de 4 Nm y su frecuencia varía hasta valores de 10 Hz a través del tiempo en que se realizan las simulaciones. En otras simulaciones, no reportadas en esta memoria de tesis, se probaron entradas con mayores frecuencias y amplitudes, sin pérdidas en la capacidad de identificación. Los resultados de los experimentos reportados se muestran en gráficas que permiten hacer comparaciones y medir la efectividad de los distintos modelos de identificación.

Se realiza la simulación con cada uno de los modelos mencionados. Se obtiene la trayectoria de estado de la posición de cada articulación y de su identificador; para cada articulación se obtiene el error de identificación, tanto de la posición como de la velocidad. Se obtiene además la evolución de los pesos sinápticos correspondientes al modelo neuronal utilizado.

### 4.5.1.2 Interpretación de resultados

En Figura 4.11 están impresas las gráficas de trayectoria de la posición de la primera articulación y de su identificación por la neurona-estado correspondiente, cuando se emplea el modelo neuronal (4.5). En Figura 4.12, se muestra lo propio para el modelo (4.9).

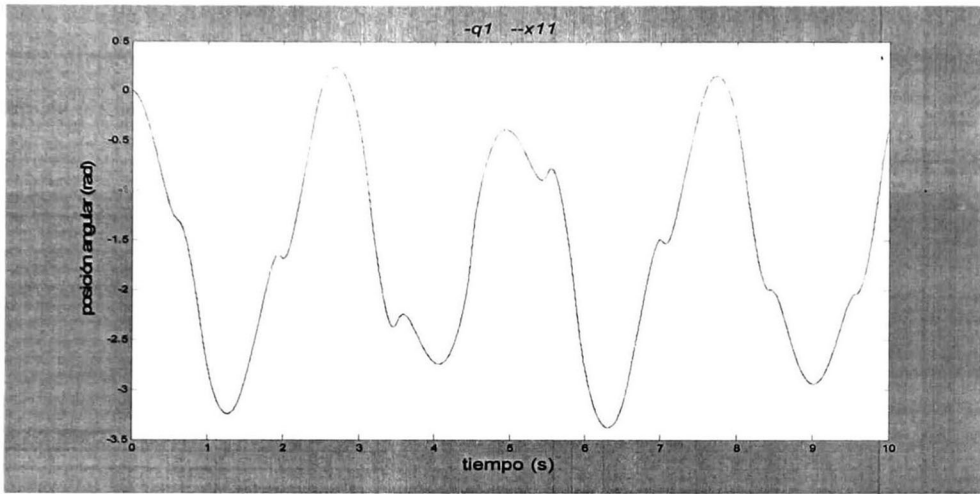


Figura 4.11. Identificación de  $q_1$ , con modelo (4.5).

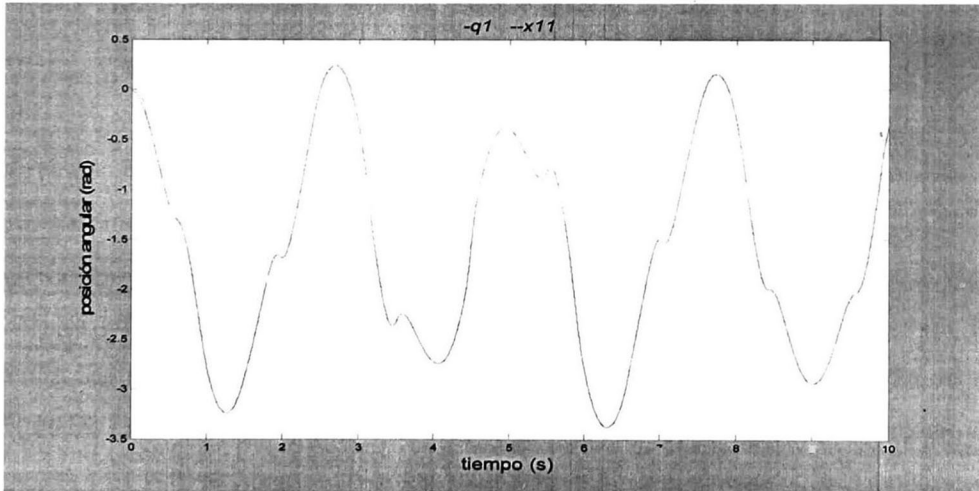


Figura 4.12. Identificación de  $q_1$ , con modelo (4.9).

Se observa en estas figuras el buen desempeño de ambos modelos de identificación; el error de identificación relativo es tan pequeño que no se distingue la diferencia entre la posición

real y su identificador; aunque con el modelo (4.9), para un corto intervalo de tiempo inicial, se observa una excepción, corresponde esto a un periodo de transición, hacia una región favorable, en el ajuste del peso sináptico de la primera neurona.

Al sustituir el modelo (4.5) por cualquiera de los otros modelos en el mismo esquema de identificación, ya sea (4.6), (4.7) o (4.8), las gráficas obtenidas son iguales a las que aquí se muestran. Igualmente, al emplear el modelo (4.10), del mismo esquema de identificación que (4.9), se obtiene prácticamente la misma efectividad en la identificación y las gráficas parecen idénticas a las del modelo (4.9).

Similares resultados se obtienen para la segunda articulación, en Apéndice A pueden observarse las gráficas correspondientes.

En Figura 4.13 se muestran las gráficas de identificación de velocidad con el modelo (4.5). A esta escala no se observa la diferencia entre el estado de la planta y el estado del identificador, esto es una muestra del alto desempeño del modelo neuronal.

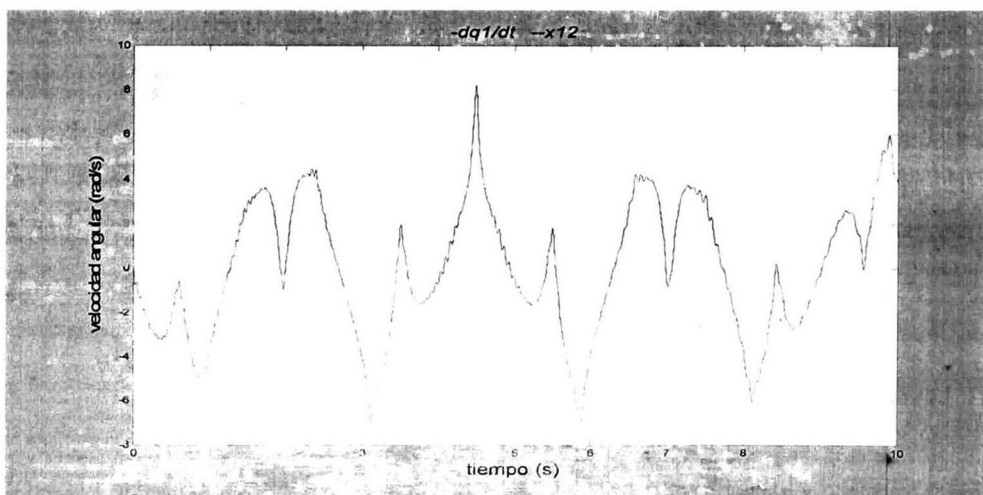


Figura 4.13. Identificación de  $dq_1/dt$  con modelo (4.5).

La gráfica que corresponde a la segunda articulación puede ser vista en Apéndice A; además de las que corresponden al modelo (4.9). El comportamiento de la segunda neurona, en cada red neuronal, está dado por la misma ecuación diferencial en los modelos (4.5) y (4.9), por lo tanto, se obtienen las mismas gráficas para ambos. Esta misma relación existe entre los modelos (4.6) y (4.10). En otro sentido, cuando se sustituyen estos modelos, por cualquiera de los otros modelos propuestos, el resultado es el mismo en términos



prácticos; el error de identificación relativo es tan pequeño, que vistos a una escala en la que se visualizan los estados reales, los estados del identificador se igualan con estos.

En gráficas que se muestran a continuación sí pueden observarse las diferencias de desempeño entre los distintos modelos de identificación, que realmente existen, aunque en pocos detalles.

En Figura 4.14 se muestra el error de identificación de  $q_1$ , cuando se emplea el modelo neuronal (4.5). Se observa que el error permanece en una región de acotamiento muy pequeña alrededor de cero. Esto explica por qué en términos prácticos no se aprecia la diferencia entre la señal de la planta y su identificador.

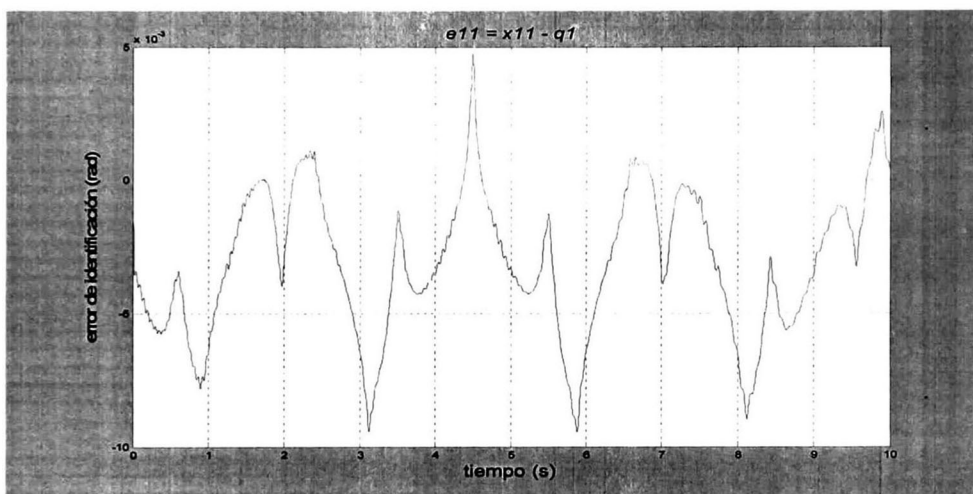


Figura 4.14. Error de identificación de  $q_1$ , con modelo (4.5).

En las siguientes gráficas se muestra el error de identificación de  $q_1$ , que se obtiene con los modelos (4.6) y (4.7); las trayectorias son muy parecidas entre sí, pero puede observarse una mejora uniforme, aunque muy pequeña, con el modelo (4.7), el cual incluye en su estructura una interconexión de segundo orden en cada red neuronal, a diferencia del (4.6), en el que las interconexiones son todas de primer orden. Se interpreta este resultado como una muestra de que el hecho de agregar interconexiones de segundo orden puede reducir uniformemente el error de identificación, aunque no sustancialmente. El valor más negativo con (4.6) es aproximadamente  $-6.5 \times 10^{-3}$ , con (4.7) es aproximadamente  $-5.9 \times 10^{-3}$ ; los más positivos, aproximadamente  $3.8 \times 10^{-3}$  y  $3.3 \times 10^{-3}$ , respectivamente; y así, consistentemente a lo largo de toda la trayectoria, el valor absoluto del error es ligeramente menor con (4.7).

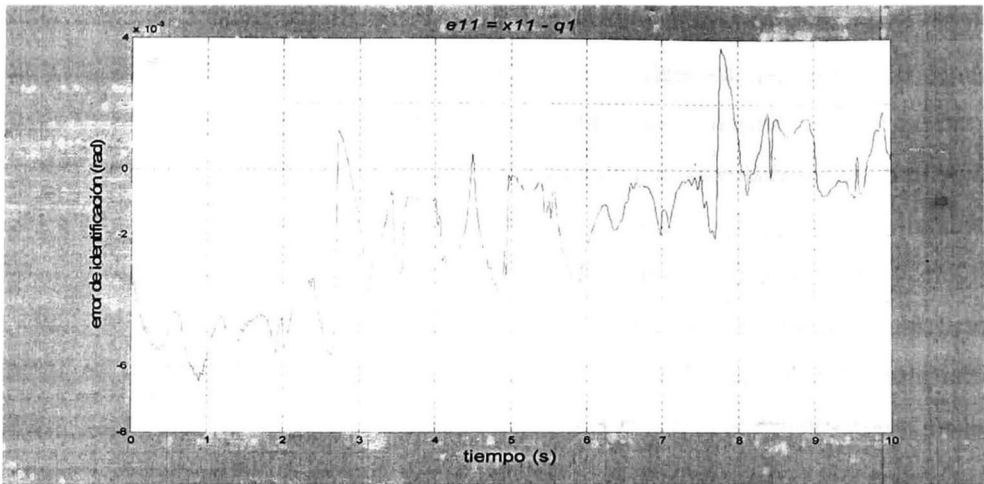


Figura 4.15. Error de identificación de  $q_1$ , con modelo (4.6).

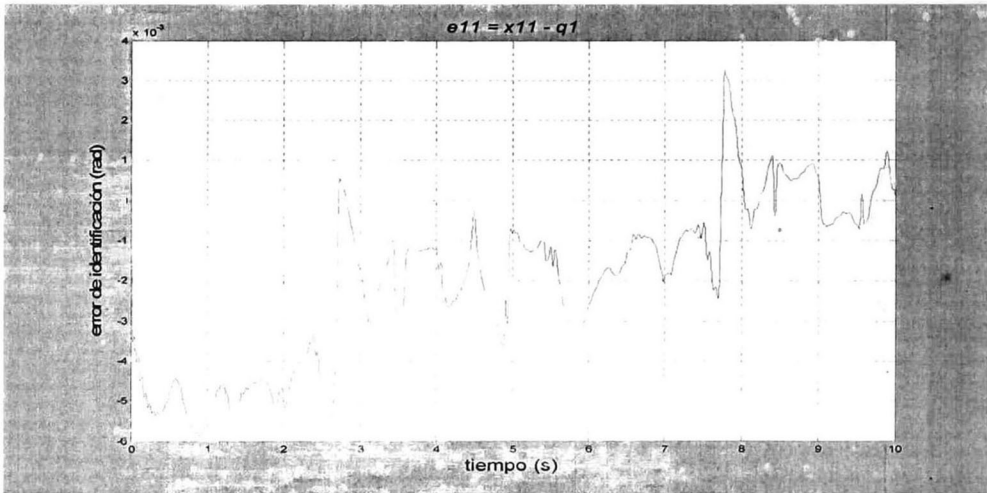


Figura 4.16. Error de identificación de  $q_1$ , con modelo (4.7).

El comportamiento del error de identificación de  $q_1$ , cuando se emplea el modelo neuronal con interconexiones de alto orden, con la estructura de (4.8) se muestra en la gráfica de Figura 4.17. Con este modelo no se observa una mejora consistente en la identificación, la región de acotamiento del error de identificación se reduce para la segunda articulación (ver Apéndice A), pero para la primera se obtiene un incremento. Se interpreta este resultado como evidencia de que con el uso de interconexiones de alto orden, no se supera necesariamente el desempeño alcanzado con redes neuronales recurrentes, con interconexiones de primer orden únicamente. Lo anterior no significa una contradicción

absoluta a la afirmación de que, en general, la capacidad de aproximación es mejorada al aumentar el número de interconexiones de alto orden; probablemente se requiera aún un número considerablemente mayor y una mejor selección de interconexiones. Sin embargo, esto aumentaría significativamente el número de operaciones computacionales, lo cual se trata de evitar en el propósito de esta tesis.

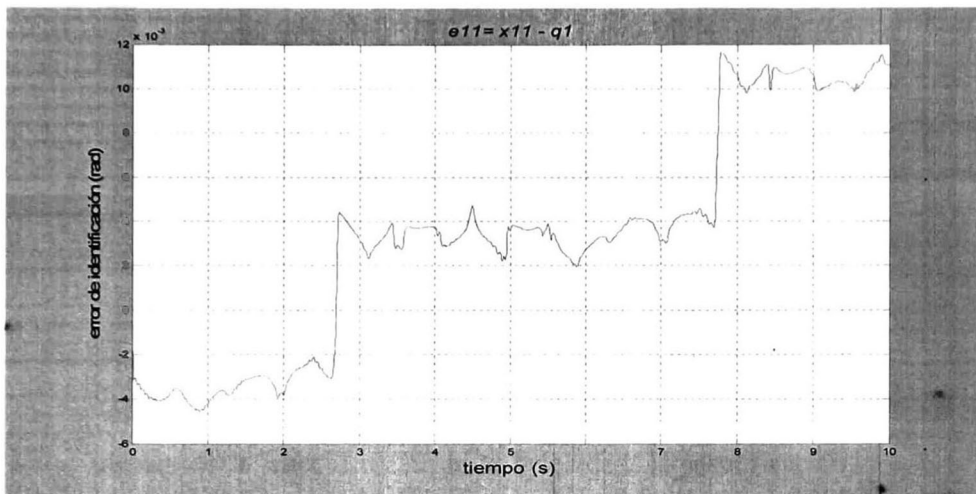


Figura 4.17. Error de identificación de  $q_1$ , con modelo (4.8).

El comportamiento del error de identificación de velocidad, que se obtiene con el modelo (4.5), se muestra en Figura 4.18, para la primera articulación.

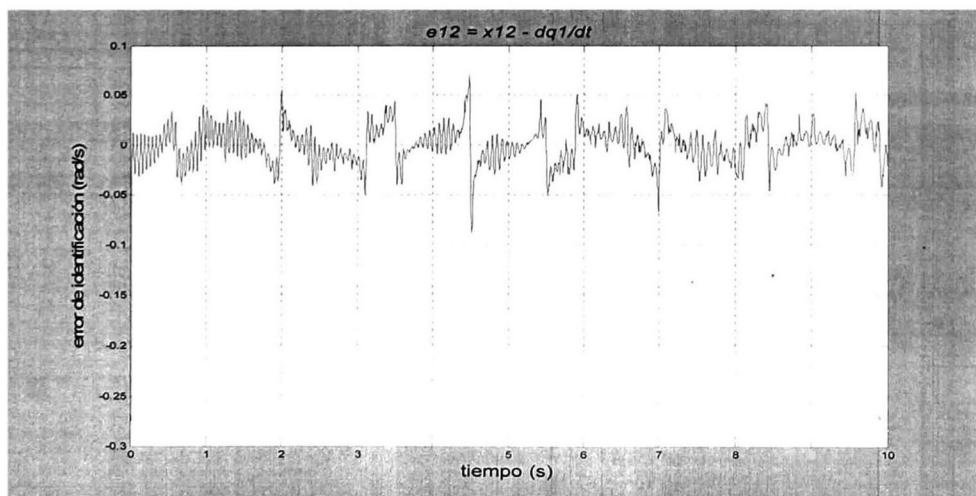


Figura 4.18. Error de identificación de  $dq_1/dt$ , con modelo (4.5).

Un error finalmente acotado, muy pequeño y que oscila alrededor de cero, se obtiene con los modelos neuronales que se prueban en el desarrollo de esta tesis. Sin embargo, la región de acotamiento es más reducida con el modelo (4.5), en comparación con los modelos (4.6), (4.7) y (4.8). En Figura 4.19 a Figura 4.21 se observan los resultados de estos tres modelos. Se observan más picos (la amplitud del error crece y decrece más rápidamente) en la medida en que se tienen más interconexiones y éstas son de mayor orden. Estos picos, bajo el riesgo de resultar amplificados con el sistema en lazo cerrado, puede ser una característica no deseable en el comportamiento del error de identificación.

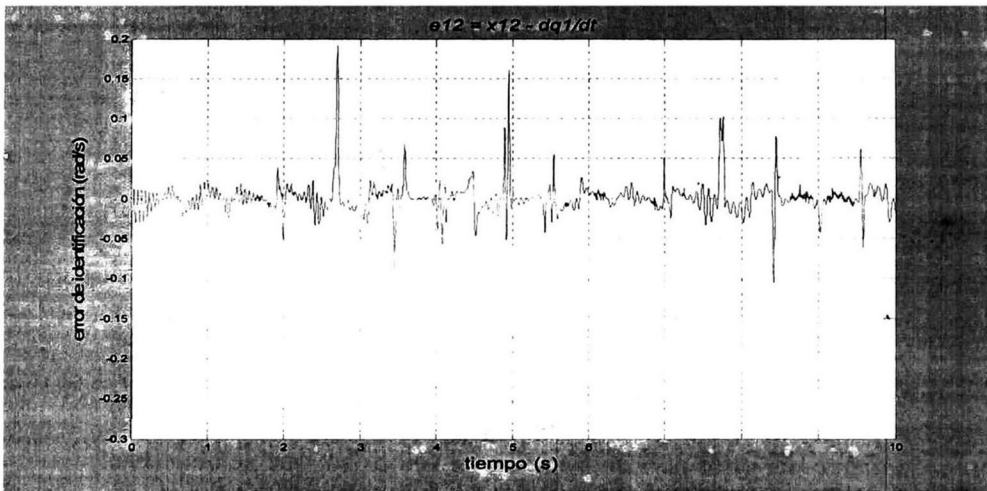


Figura 4.19. Error de identificación de  $dq_1/dt$ , con modelo (4.6).

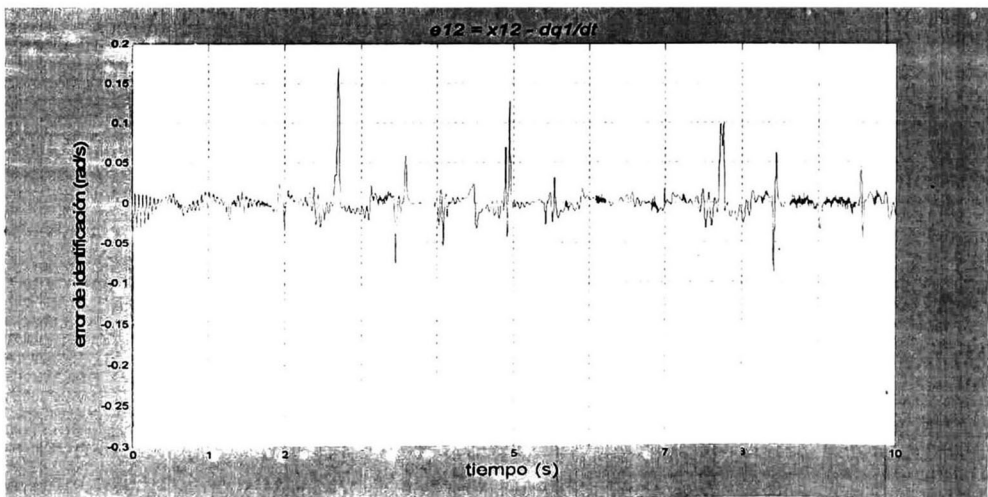


Figura 4.20. Error de identificación de  $dq_1/dt$ , con modelo (4.7).

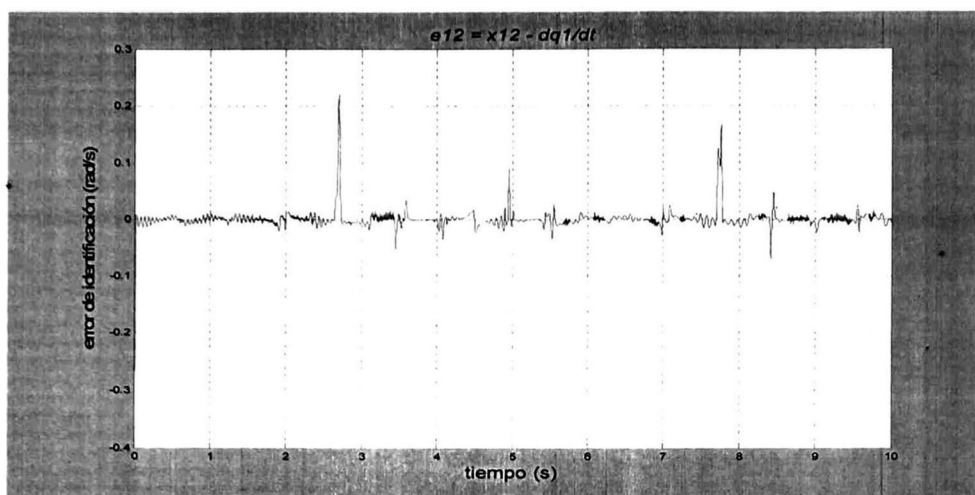


Figura 4.21. Error de identificación de  $dq_1/dt$ , con modelo (4.8).

Las gráficas de error de identificación, de posición y de velocidad, que corresponden a la segunda articulación, para cada uno de los modelos, se encuentran en Apéndice A.

A continuación se presentan los resultados de los modelos probados en el esquema que desarrolla, además de la identificación de la velocidad, la identificación neuronal de la posición. En la sección 4.4 de esta tesis se describen los modelos (4.9) y (4.10), que son diseñados con base en el esquema mencionado.

Al inicio de esta sección, en Figura 4.12, están las gráficas de trayectoria de la posición de la primera articulación y de su identificación por la neurona-estado correspondiente, cuando se emplea el modelo neuronal (4.9); ahí se menciona que sólo en un corto período inicial, las trayectorias difieren un poco. Las gráficas siguientes muestran el error de identificación, a partir de ellas puede realizarse una comparación más detallada.

En Figura 4.22 se muestra el error de identificación de la posición, cuando se emplea el modelo neuronal (4.9), para la primera y segunda articulación, respectivamente. Lo propio para el modelo (4.10), se encuentra en Figura 4.23. Al comparar los resultados con los dos modelos se observa un mejor comportamiento del modelo (4.10); esto se debe a que dicho modelo identifica a la variable de velocidad con una interconexión adicional que procede de una función sigmoideal dependiente de la posición, lo cual afecta indirecta y favorablemente a la identificación de la variable de posición, ya que el estado que identifica a la velocidad es una entrada en la neurona que identifica a la posición.

En Figura 4.24 y Figura 4.25, el error de identificación de la velocidad, con (4.9) y (4.10).



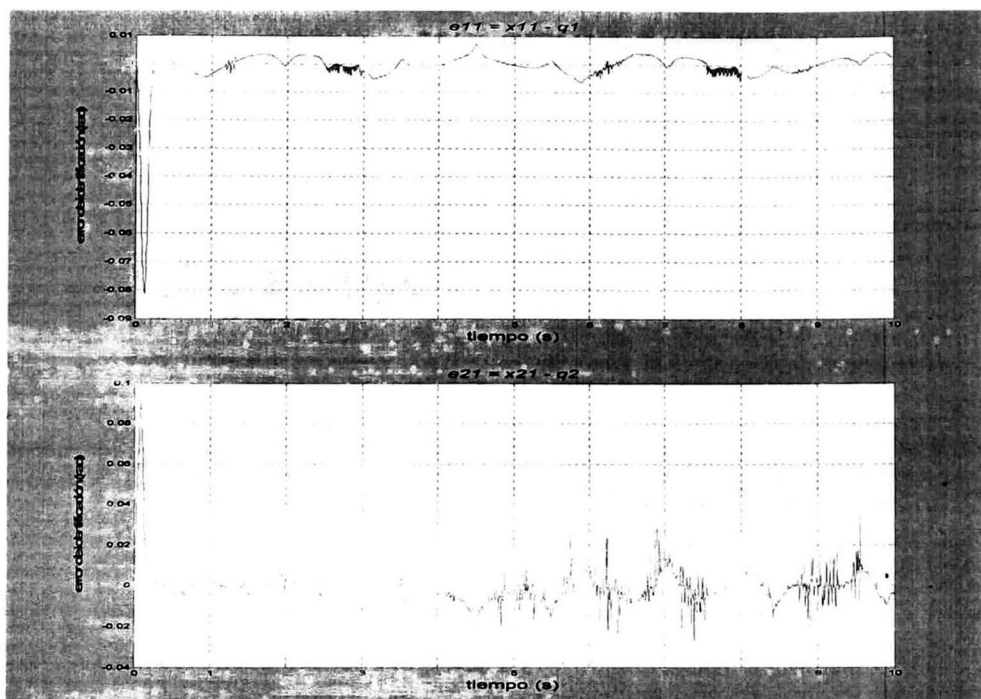


Figura 4.22. Error de identificación de  $q_1, q_2$ , con modelo (4.9).



Figura 4.23. Error de identificación de  $q_1, q_2$ , con modelo (4.10).

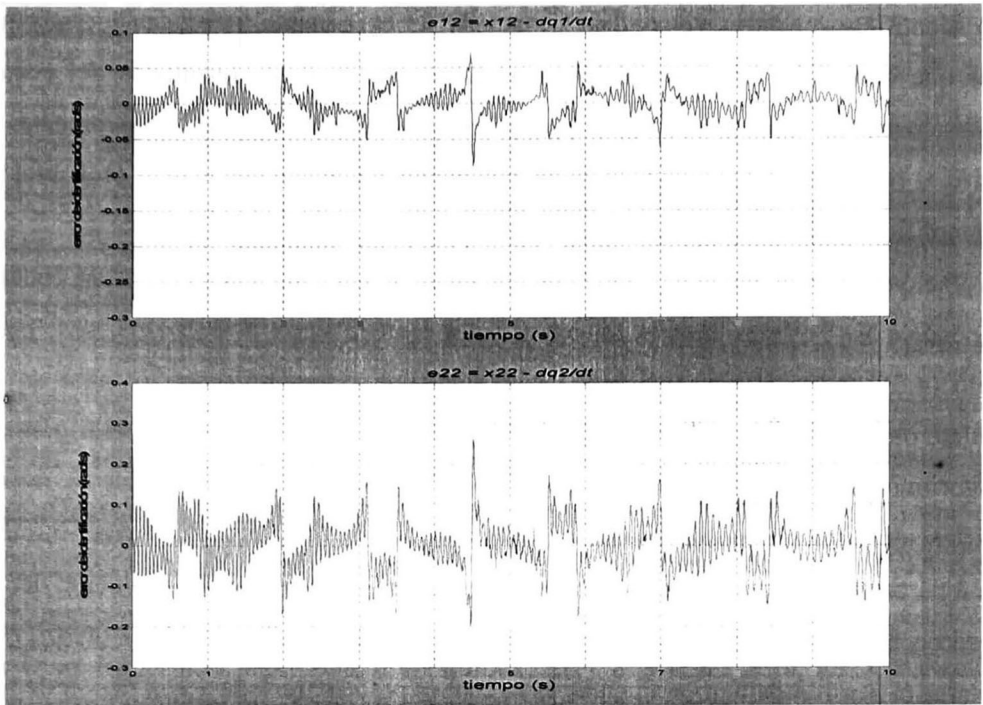


Figura 4.24. Error de identificación de  $dq_1/dt$ ,  $dq_2/dt$ , con modelo (4.9).

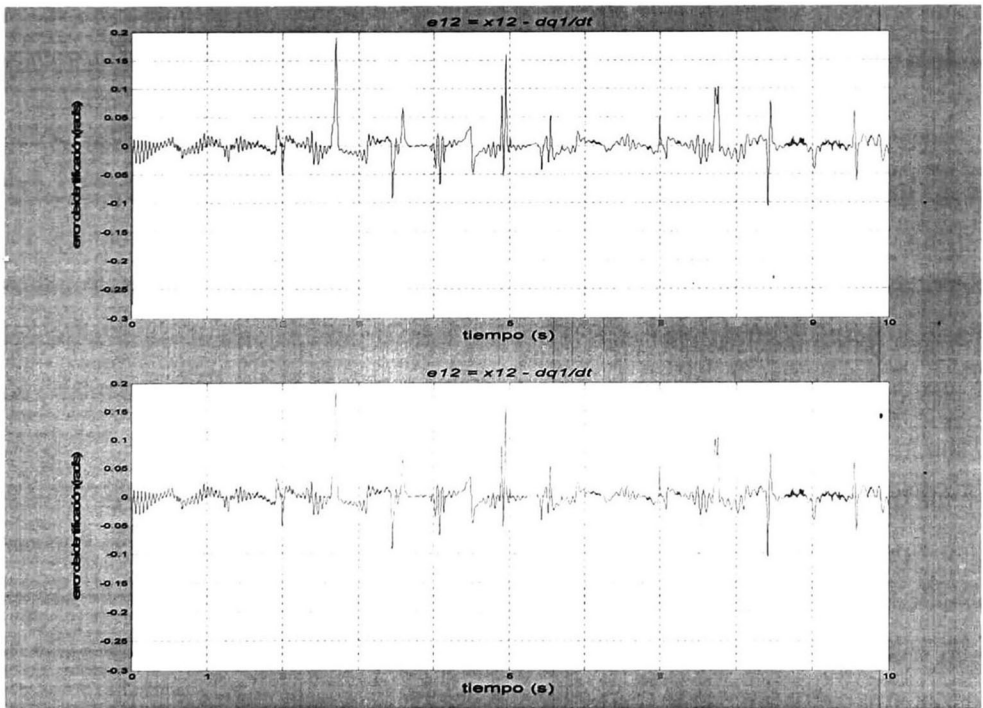


Figura 4.25. Error de identificación de  $dq_1/dt$ ,  $dq_2/dt$ , con modelo (4.10).

En la identificación de la velocidad, la cota del error con (4.10) es considerablemente mayor que con (4.9). Se repite el hallazgo mencionado en los resultados del esquema anterior: cuando se agregan interconexiones el error puede ser disminuido ligeramente, pero se corre el riesgo de la aparición de picos con mayor magnitud, los cuales probablemente sean amplificados con el sistema en lazo cerrado.

La evolución de todos los pesos sinápticos, para cada uno de los seis modelos, se encuentra en Apéndice A. Los resultados más importantes, con relación a los pesos, son: para el modelo (4.5), cada red neuronal posee sólo una interconexión, y por lo tanto un peso sináptico; obviamente no hay posibilidad de divergencia, entonces no es necesario emplear un algoritmo de aprendizaje robusto. En el resto de los modelos, en lo que se refiere a identificación de la velocidad, la interconexión utilizada en (4.5) es la que mayor influencia tiene, a partir de la evolución del peso relacionado. Por otra parte, en la evolución del peso asociado a la interconexión de la primera neurona, en los modelos (4.9) y (4.10), puede detectarse el periodo de tiempo inicial tras el cual el peso se traslada a una región de mejor desempeño.

## **4.6 Identificación del manipulador robótico tipo ANAT**

Todos los modelos de identificación propuestos en esta tesis muestran, en general, un buen desempeño, en la identificación del manipulador robótico de dos articulaciones. Para la identificación de la dinámica del manipulador robótico tipo ANAT de siete articulaciones se utilizan sólo modelos del tipo de redes neuronales recurrentes, únicamente con interconexiones de primer orden, con el propósito de probar los modelos más simples. Estos son los modelos (4.5), (4.6), (4.9) y (4.10).

El sistema que se utiliza para ser identificado, es el modelo computacional del manipulador robótico tipo ANAT, que se describe en el capítulo anterior de esta tesis. Las pruebas consisten en simular computacionalmente el comportamiento simultáneo del manipulador robótico y del modelo neuronal en el esquema de identificación descentralizada. La misma entrada es aplicada a ambos modelos y el identificador ajusta sus parámetros a partir del error en la identificación de los estados, tal como se representa en Figura 4.10.

## **4.6.1 Evaluación experimental**

Para llevar a cabo la evaluación experimental las simulaciones son implementadas en MATLAB® y Simulink® (marcas registradas de MathWorks, Inc.). Los resultados de las pruebas son evaluados mediante gráficas de comportamiento del error, de trayectoria de los estados y de evolución de los pesos.

El modelo matemático del manipulador robótico es implementado a través de módulos ejecutables en MATLAB® y el modelo neuronal descentralizado es implementado en Simulink®, ambos se integran en un programa para completar el esquema de identificación, que se ejecuta en la plataforma de Simulink®. El manipulador robótico es considerado con las siguientes características: cada uno de sus eslabones tiene una masa de 0.5 Kg; las longitudes nominales de los mismos son, en el orden usual con respecto a la base: 0.57 m, 0.16641 m, 0.12278 m, 0.0845 m, 0.1756 m, 0.25107 m y 0.02343 m; los coeficientes de fricción cinética, en el mismo orden: 0.1, 0.2, 0.1, 0.2, 0.1, 0.2, y 0.1.

El propósito principal de los experimentos es probar la efectividad del esquema de identificación neuronal descentralizada con un modelo que contiene mayor número de interacciones dinámicas, con mayor número de subsistemas, todos interactuando entre sí.

### **4.6.1.1 Descripción de los experimentos**

En todos los experimentos se aplican las mismas entradas, producidas por un generador de señales senoidales de frecuencia variable (“chirp”); para este robot se utilizan diferentes rangos de frecuencia y amplitud con el fin de observar posibles efectos de las dinámicas de acoplamiento ante entradas con mayor diversidad entre una y otra articulación, las frecuencias con valores de hasta 100 Hz, amplitudes entre 10 Nm y 20 Nm, para uniones giratorias, y 1000 N para la unión prismática.

Se realiza la simulación con cada uno de los modelos mencionados. Se obtiene la trayectoria de estado de la posición de cada articulación y de su identificador; para cada articulación se obtiene el error de identificación, tanto de la posición como de la velocidad; se obtiene además la evolución de los pesos sinápticos del modelo neuronal utilizado.



### 4.6.1.2 Interpretación de resultados

En las gráficas de Figura 4.26 se muestran la identificación de la posición para tres de las siete articulaciones, correspondientes al desempeño del modelo (4.10).

Las tres articulaciones son seleccionadas como una muestra de la diversidad en las trayectorias de las uniones. Al igual que en la identificación del manipulador de dos grados de libertad, a escalas en las que se aprecia la trayectoria completa, prácticamente no es posible distinguir la diferencia entre el estado del sistema y el estado del identificador.

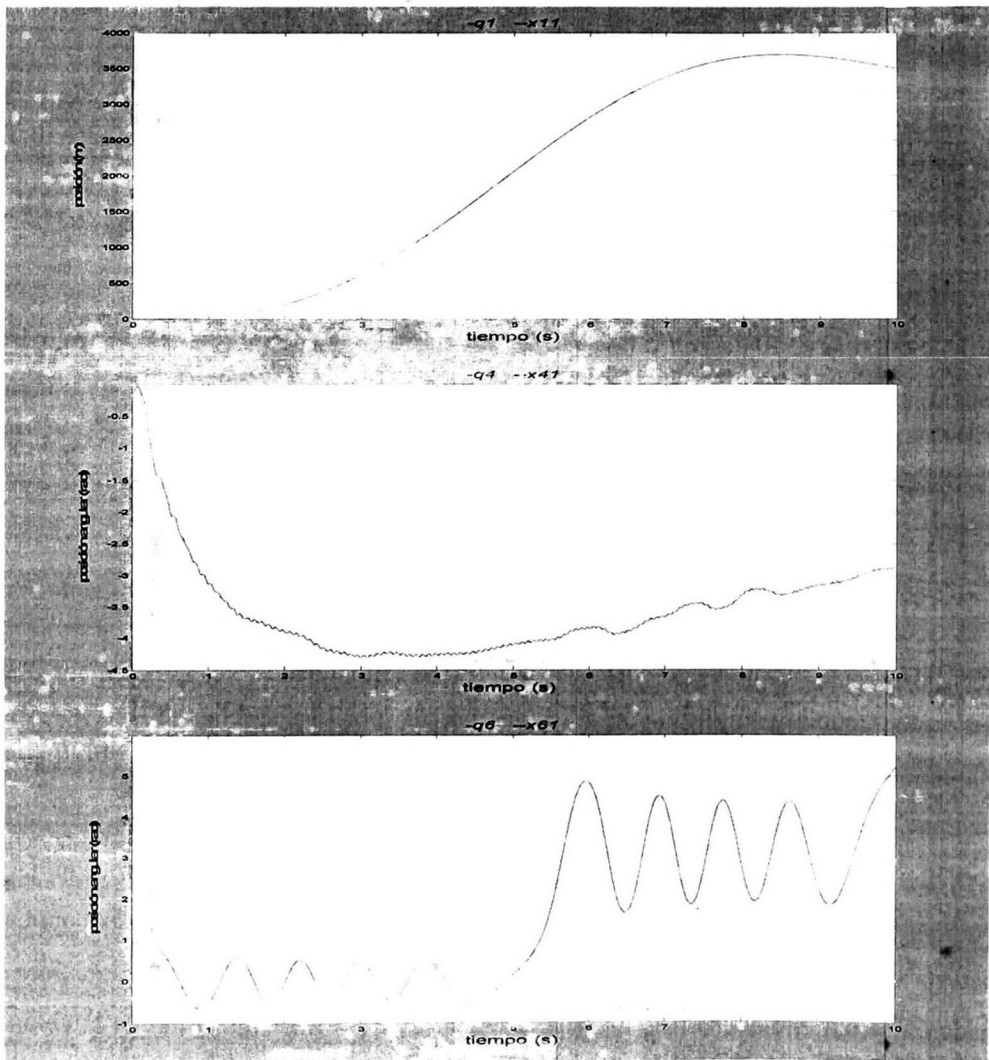


Figura 4.26. Identificación de  $q_1, q_4, q_6$ , con modelo (4.10).



Este experimento aporta más información, puede observarse la elevada magnitud de la posición en la primera articulación, las variaciones de alta frecuencia en la cuarta articulación, más las oscilaciones de la sexta articulación. Esto permite pronosticar el buen desempeño de la identificación frente a diversas condiciones en la evolución de los estados. En Apéndice A están las gráficas para todas las articulaciones, en algunas gráficas de identificación de la posición puede verse una pequeña diferencia entre el identificador y la posición de la articulación por un período inicial muy pequeño. Este periodo corresponde al tiempo que tarda cada identificador, a partir de condición inicial cero, en aproximar a la variable de posición del sistema real, el cual se simula con condiciones iniciales diferentes de cero. Con excepción de lo anterior, los resultados son similares para todos los modelos, si son vistos desde las gráficas de evolución de las trayectorias de los estados del manipulador y del identificador.

Se coloca a la planta y al identificador en distinto estado inicial, en lo que se refiere a la posición, para fines experimentales. En realidad, dado que se dispone de la medición de las posiciones las condiciones iniciales pueden ser iguales.

Las gráficas posteriores, de Figura 4.27, muestran la identificación de la velocidad para las mismas articulaciones. No se observa la diferencia entre la trayectoria del estado de la planta y la trayectoria del estado del identificador. En ambos sistemas, los estados correspondientes a las velocidades parten de condiciones iniciales cero.

Es notable, que para algunas articulaciones, la velocidad tiene una trayectoria de alta frecuencia, producidas precisamente por las altas frecuencias generadas por la función que provee las entradas de entrenamiento, y por la transmisión de dinámicas de uno a otra articulación por el fuerte acoplamiento dinámico entre sus elementos. Este comportamiento resulta idóneo para propósitos de prueba, si el modelo responde adecuadamente con este entrenamiento, se espera que su desempeño en situaciones reales sea aún más favorable.

Cada identificador debe adaptarse a las características dinámicas del estado que identifica, en este experimento las siete redes neuronales se ajustan exitosamente al comportamiento del estado real que identifica; esto es un resultado importante, dado que la red neuronal hace uso sólo de variables locales, mientras que la dinámica de la articulación correspondiente esta siendo afectada por dinámicas derivadas del resto de las articulaciones.

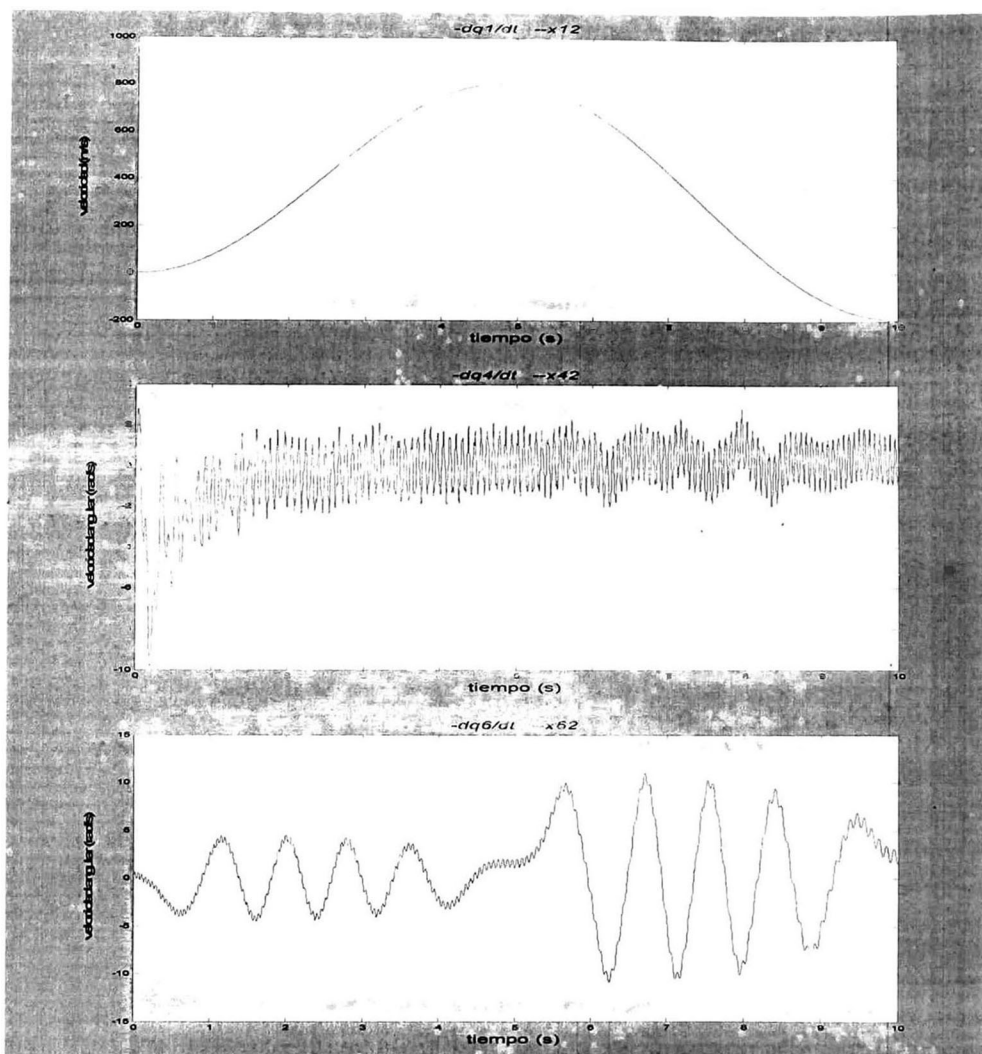


Figura 4.27. Identificación de  $dq_1/dt$ ,  $dq_4/dt$ ,  $dq_6/dt$ , con modelo (4.10).

En Figura 4.28 están contenidas las gráficas del error de identificación, para las tres articulaciones seleccionadas.

El comportamiento del error de identificación de la posición muestra un buen desempeño para cada uno de los subsistemas (articulaciones), y converge a una región de acotamiento pequeña en la vecindad de cero. Aunque aparentemente esto no se cumple para la primera articulación, se explica por la trayectoria de su posición, la cual evoluciona rápidamente hacia un valor muy alto. En condiciones reales las restricciones físicas impiden esto; en las pruebas ocurre por que se ensaya con el simulador una entrada de gran magnitud.

El error relativo, para todas las articulaciones, incluida la primera, con las particularidad ya

mencionadas, es menor a 0.2%. En términos absolutos, el error de identificación en las seis articulaciones giratorias oscila dentro de valores máximos de una, dos y cuatro milésimas de radián; lo cual, para cuestiones prácticas es muy conveniente.

Otra observación interesante es que el comportamiento del error de posición está muy correlacionado con el comportamiento de la velocidad de la articulación; de este modo, puede pronosticarse el desempeño del identificador en base a la complejidad del movimiento de las articulaciones. En el entrenamiento se emplean entradas que provocan movimientos muy rápidos en las articulaciones, en situaciones reales se esperan trayectorias de las articulaciones menos complejas.

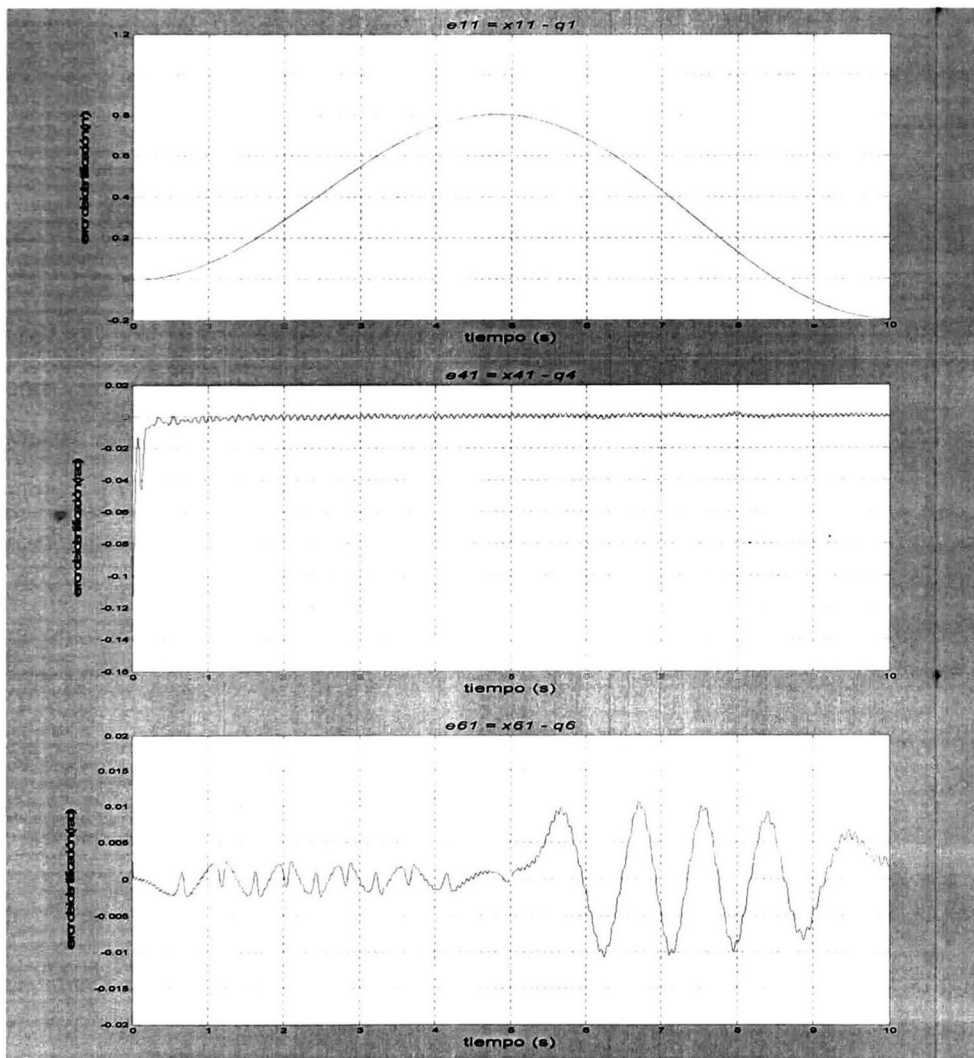


Figura 4.28. Error de identificación de  $q_1$ ,  $q_4$ ,  $q_6$ , con modelo (4.10).

En las siguientes gráficas se muestra el error de identificación de la variable de velocidad. Para la identificación de la velocidad resultan similares los modelos (4.5) y (4.9), y similares también entre sí el (4.6) y el (4.10); en Figura 4.29 se ven los resultados. Estas gráficas requieren una observación cuidadosa, dado que éstas tienen distintas escalas.

Modelos (4.5) y (4.9):

Modelos (4.6) y (4.10):

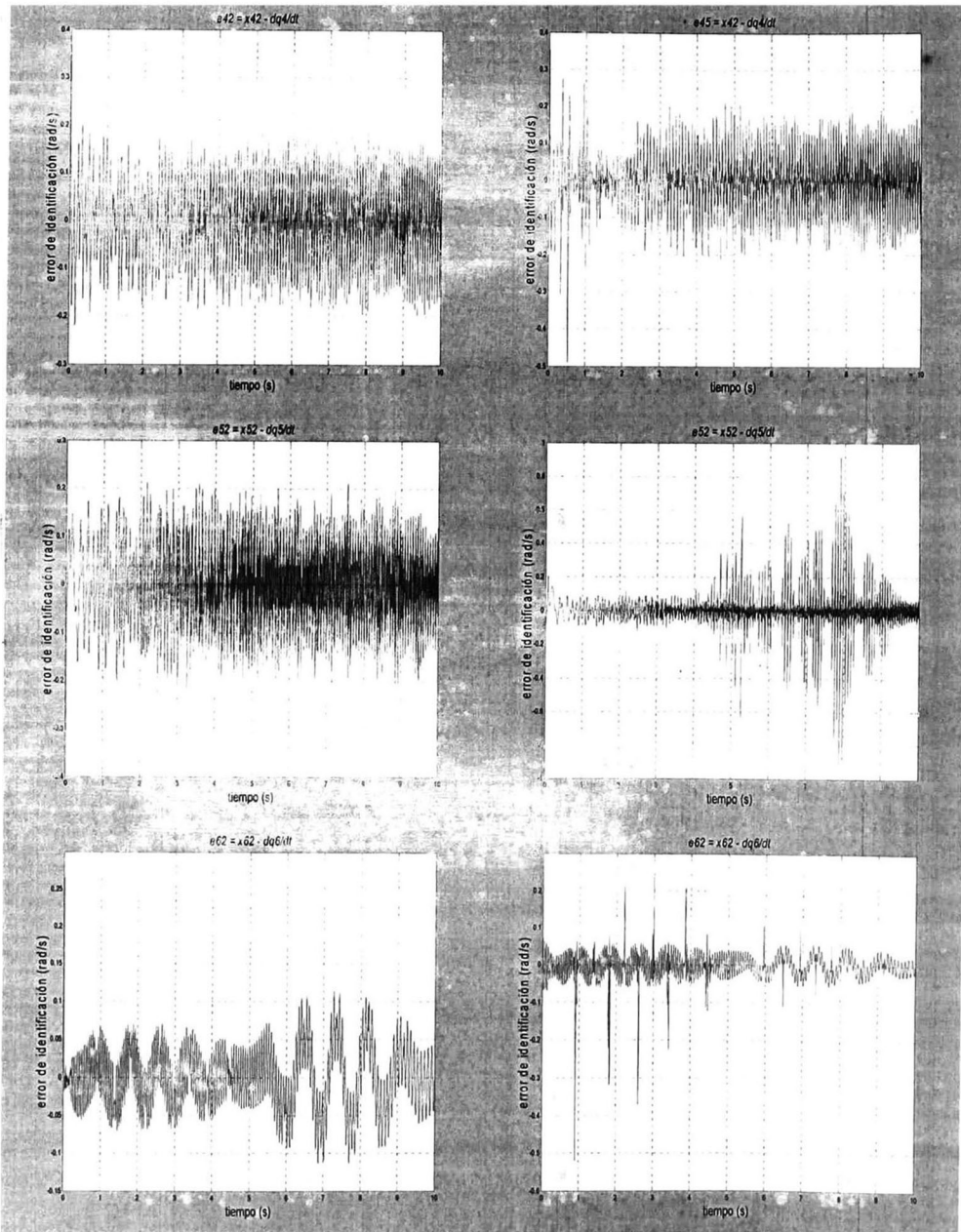


Figura 4.29. Error de identificación de  $dq_4/dt$ ,  $dq_5/dt$ ,  $dq_6/dt$ . Izq.: (4.5), (4.9). Der.: (4.6), (4.10).



Resulta, de manera coherente con los resultados en el robot de dos articulaciones, que cuando se enriquece la red neuronal con más interconexiones, puede alcanzarse una disminución del error. También puede verse que los cambios de signo en la posición o en la velocidad de la articulación que se identifica, pueden provocar la aparición de picos en la trayectoria del error; esto es atribuido a que en tales circunstancias, si alguna función de activación procesa uno o más de esos estados, la función sigmoïdal va abandonando la región de saturación y entra a una región crítica en la medida en que su argumento se va acercando al cero. En general, la evolución del error de identificación obedece fielmente a las variaciones en las trayectorias del sistema real.

La evolución de los pesos sinápticos puede observarse a partir de las gráficas de Apéndice A. Los pesos en (4.5) y (4.6) evolucionan igual que los pesos correspondientes en (4.9) y (4.10). Si se utiliza una interconexión para cada neurona se garantiza que los valores de los pesos no corran con el riesgo de evolucionar hacia el infinito. Lo anterior se debe a que sus trayectorias se modifican con los estados de la planta, y en este caso se tiene un sistema pasivo, con estados acotados ante entradas acotadas. Con más interconexiones se corre el riesgo de que los valores de los pesos puedan evolucionar hacia valores muy grandes; en un subconjunto de los pesos, éstos pueden contribuir en sentidos opuestos, compensándose entre sí. Esto hace necesario que en tales casos, por seguridad, se introduzca la modificación robusta en la ley de ajuste de pesos.



# Capítulo 5

## Control neuronal descentralizado de manipuladores robóticos

*En este capítulo se expone el esquema de control neuronal descentralizado propuesto para llevar a cabo tareas regulación y seguimiento en manipuladores robóticos, y se presentan los resultados de simulación para un manipulador robótico de dos eslabones y para un robot ANAT de siete eslabones, a través de los diferentes modelos de redes neuronales expuestos en el capítulo previo.*

### 5.1 Introducción

De acuerdo a lo expuesto en el capítulo anterior, los diferentes modelos que se proponen en esta tesis, para la identificación neuronal descentralizada, pueden ser representados por el siguiente modelo general:

$$\begin{aligned}\dot{x}_{i1} &= -a_{i1}x_{i1} + \mathbf{w}_{i1}^T \boldsymbol{\rho}_{i1}(\boldsymbol{\chi}_i) + x_{i2} \\ \dot{x}_{i2} &= -a_{i2}x_{i2} + \mathbf{w}_{i2}^T \boldsymbol{\rho}_{i2}(\boldsymbol{\chi}_i) + u_i\end{aligned}\tag{5.1}$$

donde  $i = 1, 2, \dots, N$ ,  $x_{i1}$  y  $x_{i2}$  son los estados (neuronas) de la  $i$ -ésima red neuronal, que identifican respectivamente a  $\chi_{i1}$  y  $\chi_{i2}$ , las cuales representan a la posición y a la velocidad de la  $i$ -ésima articulación;  $a_{i1}$ ,  $a_{i2}$  son constantes reales positivos;  $\boldsymbol{\rho}_i(\boldsymbol{\chi}_i)$  es el vector de

interconexiones preprocesadas por su función de activación sigmoïdal, posiblemente de alto orden, que entran a la  $i$ -ésima neurona, con su correspondiente vector de pesos sinápticos  $w_{ij}$ ; y  $u_i$  es la fuerza o torque aplicado a la  $i$ -ésima articulación.

Como se explica en el capítulo anterior, una estructura como la del modelo (5.1) es afín en el control. La variable a ser directamente controlada, a través de  $u_i$ , es  $x_{i2}$ , que a su vez actúa como una variable de control ficticia, a través de la cual  $x_{i1}$  es controlada.

Dada la estructura de (5.1), y el procedimiento propuesto para la síntesis del control, el objetivo del control se define a partir de la trayectoria deseada para la posición de cada articulación del manipulador robótico, considerando la primera y segunda derivadas respectivas. La trayectoria deseada para la posición de cada articulación se traduce en la señal deseada para la variable del modelo neuronal que identifica a dicha posición.

Este enfoque puede considerarse como un control adaptable indirecto; la señal de control se va ajustando de acuerdo al cambio en los valores de parámetros del modelo de red neuronal, y es un control diseñado sobre este modelo, para actuar a la vez sobre el manipulador robótico.

Dado que la estructura del modelo neuronal es afín en el control, es posible diseñar un control que lleve la trayectoria de sus estados a converger asintóticamente hacia la referencia.

El esquema de identificación permite que la diferencia, entre el vector de estados del manipulador robótica y el vector de estados del modelo neuronal descentralizado, resulte finalmente acotado; esto implica que si el error de control del modelo neuronal puede ser llevado a cero, entonces, para el sistema real, el error de control puede ser llevado hacia la pequeña región a la que converge el error de identificación.

En particular, para cada variable de posición de la articulación, del manipulador robótico, el valor absoluto del error de control es menor o igual a la suma del valor absoluto del error de identificación, más el valor absoluto del error de control del modelo identificador. El primer término de esta suma (error de identificación), converge a una región pequeña y el segundo (error de control del modelo neuronal), converge asintóticamente a cero.

Dada la señal de referencia  $\beta_{i1}$ , para cada articulación, se cumple esta desigualdad:

$$|x_{i1} - \beta_{i1}| \leq |x_{i1} - \chi_{i1}| + |x_{i1} - \beta_{i1}|.$$

## 5.2 Método general de diseño del control

En esta sección se desarrolla el esquema de control, para regulación y seguimiento de trayectoria, para el modelo neuronal descentralizado (5.1). El modelo describe la estructura, de forma general, para cualquiera de los casos propuestos en la tesis.

En la primera ecuación de (5.1)  $x_{i2}$  aparece como un control ficticio para  $x_{i1}$ , en la segunda ecuación de (5.1) aparece el control verdadero  $u_i$ ; en ambas ecuaciones se tiene una estructura afin en el control. Aprovechando lo anterior, a través de  $x_{i2}$  puede controlarse  $x_{i1}$ , para que siga la señal de referencia deseada, y  $x_{i2}$  se controlada a través de  $u_i$ .

Con este esquema de control se garantiza que el manipulador robótico, identificado por un enfoque descentralizado neuronal, presente un comportamiento deseado definido sobre las trayectorias de las posiciones de sus articulaciones.

Por cada subsistema, es decir, para cada articulación, se introduce una nueva variable que corresponde el error de control de cada red neuronal:

$$z_{i1} = x_{i1} - \beta_{i1},$$

donde  $\beta_{i1}$ , con su primera y segunda derivadas también conocidas, define la trayectoria deseada para la posición de la articulación. La nueva variable es forzada a ser gobernada por esta ecuación:

$$\dot{z}_{i1} = -k_{i1}z_{i1},$$

con lo cual se obliga al error de control del modelo neuronal a converger a cero asintóticamente, con un número real positivo  $k_{i1}$ . Para determinar la ecuación que debe gobernar el comportamiento de  $x_{i1}$ , se sustituye la derivada de  $z_{i1}$ :

$$\dot{x}_{i1} - \dot{\beta}_{i1} = -k_{i1}z_{i1},$$

se tiene entonces que:

$$\dot{x}_{i1} = -k_{i1}z_{i1} + \dot{\beta}_{i1}.$$

Se iguala la primera ecuación en (5.1) con la ecuación anterior:

$$-a_{i1}x_{i1} + \mathbf{w}_{i1}^T \boldsymbol{\rho}_{i1}(\boldsymbol{\chi}_i) + x_{i2} = -k_{i1}z_{i1} + \dot{\beta}_{i1}.$$

Despejando  $x_{i2}$  se obtiene la señal deseada para esta variable, y se distingue esta señal llamándola  $\beta_{i2}$ :

$$\beta_{i2} = -k_{i1}z_{i1} + \dot{\beta}_{i1} + a_{i1}x_{i1} - \mathbf{w}_{i1}^T \boldsymbol{\rho}_{i1}(\boldsymbol{\chi}_i).$$

Se introduce otra nueva variable para el error en la variable del control ficticio:

$$z_{i2} = x_{i2} - \beta_{i2}$$

Para obligar a la variable  $z_{i2}$  a converger a cero, asintóticamente, se debe cumplir que

$$\begin{aligned} \dot{z}_{i2} &= -k_{i2}z_{i2}, \\ \dot{x}_{i2} - \dot{\beta}_{i2} &= -k_{i2}z_{i2}, \end{aligned}$$

donde  $k_{i2}$  es un número real positivo. Despejando  $\dot{x}_{i2}$  resulta

$$\dot{x}_{i2} = -k_{i2}z_{i2} + \dot{\beta}_{i2}.$$

Se iguala con la segunda ecuación en (5.1):

$$-a_{i2}x_{i2} + \mathbf{w}_{i2}^T \boldsymbol{\rho}_{i2}(\boldsymbol{\chi}_i) + u_i = -k_{i2}z_{i2} + \dot{\beta}_{i2}.$$

Despejando  $u_i$  se obtiene la señal de control:

$$u_i = -k_{i2}z_{i2} + \dot{\beta}_{i2} + a_{i2}x_{i2} - \mathbf{W}_{i2}^T \boldsymbol{\rho}_{i2}(\boldsymbol{\chi}_i). \quad (5.2)$$



Se completa el conocimiento de todos los términos que aparecen en la ecuación de la ley de control (5.2) considerando las ecuaciones que corresponden a la ley aprendizaje y a las funciones sigmoideas, y derivando  $\beta_{i2}$ , de lo que se obtiene:

$$\dot{\beta}_{i2} = -k_{i1}\dot{z}_{i1} + \ddot{\beta}_{i1} + a_{i1}\dot{x}_{i1} - \dot{w}_{i1}^T \rho_{i1}(\chi_i) - w_{i1}^T \dot{\rho}_{i1}(\chi_i), \quad (5.3)$$

donde la derivada del vector de entradas de interconexión que aparece en el último término se obtiene a partir de

$$\dot{\rho}_{i1}(\chi_i) = \frac{\partial \rho_{i1}}{\partial \chi_i} \dot{\chi}_i.$$

Considerando que  $\chi_i = [q_i, \dot{q}_i]^T$  se tiene que  $\dot{\chi}_i = [\dot{q}_i, \ddot{q}_i]^T$ ; como se supone que la medición de  $\ddot{q}_i$  no está disponible, se introduce una restricción para las interconexiones en la primera neurona: en el argumento de las sigmoideas solamente puede emplearse la posición, y no la velocidad. La ecuación anterior se reescribe de esta manera:

$$\dot{\rho}_{i1}(q_i) = \frac{\partial \rho_{i1}}{\partial q_i} \dot{q}_i,$$

en la que la derivada parcial siempre es posible obtenerla, para cualquier orden en las interconexiones, a partir de las funciones sigmoideas seleccionadas. En el desarrollo de esta tesis se utiliza la función tangente hiperbólica, por lo que para es necesario conocer que:

$$\frac{d(\tan(q_i))}{dt} = \sec^2(q_i) \frac{dq_i}{dt}$$

Las ecuaciones anteriores resultan expresiones generales que pueden ser utilizadas para cualquier modelo de identificación descentralizada con redes neuronales de dos estados. Como se menciona antes, para la neurona-estado que identifica a la posición, sus interconexiones, que pueden ser de cualquier orden, sólo pueden incluir a la posición del manipulador robótico, obviamente preprocesada por la función de activación sigmoideal; en



la neurona-estado que identifica a la velocidad, sus interconexiones sí pueden incluir a la posición y a la velocidad.

Al igual que en el capítulo anterior, los modelos de redes neuronales empleados como identificadores del manipulador robótico, contruidos ahora con la finalidad de llevar a cabo el control, son expuestos de forma separada en dos enfoques, y dentro de cada uno de ellos se describen modelos que difieren entre sí por sus interconexiones. Se eligen para el esquema completo de identificación y control los modelos más simples.

### 5.2.1 Control en el esquema de identificación de velocidad

Para este esquema de control, en el modelo general (5.1),  $a_{i1} = 0$  y los vectores  $w_{ij}$  y  $\rho_{ij}(\chi_i)$  son nulos dado que no existen interconexiones de entrada a la primer neurona de cada red neuronal, para  $i = 1, 2, \dots, N$ . A partir de aquí, en lugar de  $\chi_i$  se prefiere usar  $q_i$  y  $\dot{q}_i$ , la posición y velocidad de la articulación  $i$ , respectivamente, considerando que  $\chi_i = [q_i, \dot{q}_i]^T$  para  $i = 1, 2, \dots, N$ .

En la síntesis del control, las ecuaciones de  $\beta_{i2}$  y  $\dot{\beta}_{i2}$  se modifican a expresiones más reducidos, a causa de los términos que se hacen nulos o cero en (5.1), como se explica antes. Resulta:

$$\begin{aligned}\beta_{i2} &= -k_{i1}z_{i1} + \dot{\beta}_{i1}, \\ \dot{\beta}_{i2} &= -k_{i1}\dot{z}_{i1} + \ddot{\beta}_{i1}.\end{aligned}$$

Haciendo las sustituciones necesarias en (5.2) se obtiene la ley de control:

$$u_i = -k_{i1}k_{i2}(x_{i1} - \beta_{i1}) - (k_{i1} + k_{i2})(x_{i2} - \dot{\beta}_{i1}) + \ddot{\beta}_{i1} + a_{i2}x_{i2} - w_{i2}^T \rho_{i2}(q_i, \dot{q}_i). \quad (5.4)$$

Este esquema de control es el que se aplica de manera más natural a los manipuladores robóticos, pues el primer estado, que es la posición de la articulación, es definido por la integral del segundo estado, que es la velocidad de la articulación. Además las condiciones iniciales se consideran conocidas al suponer que tanto la posición como la velocidad de

cada articulación son medibles en todo momento, por lo que los estados iniciales de las redes neuronales quedan establecidos con la lectura de los estados del manipulador robótico en el tiempo inicial  $t = 0$ .

### 5.2.1.1 Modelo con una interconexión de primer orden

Este modelo de red neuronal es el (4.5), expuesto en la sección 4.3.1; la neurona que identifica a la velocidad de la articulación está interconectada únicamente con la velocidad real preprocesada por la función sigmoideal. Es el modelo más simple y el control queda definido por la siguiente ecuación:

$$u_i = -k_{i1}k_{i2}(x_{i1} - \beta_{i1}) - (k_{i1} + k_{i2})(x_{i2} - \dot{\beta}_{i1}) + \ddot{\beta}_{i1} + a_{i2}x_{i2} - w_{i22}\rho_{i22}(\dot{q}_i). \quad (5.5)$$

### 5.2.1.2 Modelos con dos interconexiones de primer orden

Este modelo de red neuronal es el (4.6), expuesto en la sección 4.3.2; la neurona que identifica a la velocidad de la articulación está doblemente interconectada, a la posición y a la velocidad reales preprocesadas por su función de activación. El control queda definido por la siguiente ecuación:

$$u_i = -k_{i1}k_{i2}(x_{i1} - \beta_{i1}) - (k_{i1} + k_{i2})(x_{i2} - \dot{\beta}_{i1}) + \ddot{\beta}_{i1} + a_{i2}x_{i2} - w_{i21}\rho_{i21}(q_i) - w_{i22}\rho_{i22}(\dot{q}_i). \quad (5.6)$$

## 5.2.2 Control en el esquema de identificación de posición y velocidad

Para este esquema de control existen todos los términos del modelo general (5.1). Considerando la restricción que se tiene para las interconexiones de la primera neurona-estado de cada red neuronal, la ecuación de  $\dot{\beta}_{i2}$  resulta así:

$$\dot{\beta}_{i2} = -k_{i1}\dot{z}_{i1} + \ddot{\beta}_{i1} + a_{i1}\dot{x}_{i1} - \dot{w}_{i1}^T \rho_{i1}(q_i) - w_{i1}^T \frac{\partial \rho_{i1}(q_i)}{\partial q_i} \dot{q}_i,$$

y la ecuación de la ley de control (5.2) puede reescribirse de la forma:

$$u_i = -k_{i2}z_{i2} + \dot{\beta}_{i2} + a_{i2}x_{i2} - w_{i2}^T \rho_{i2}(q_i, \dot{q}_i). \quad (5.7)$$

Haciendo las sustituciones necesarias en (5.7) se obtiene esta nueva expresión para la ley de control:

$$\begin{aligned} u_i = & -k_{i1}k_{i2}(x_{i1} - \beta_{i1}) - (k_{i1} + k_{i2})(x_{i2} - \dot{\beta}_{i1}) + \ddot{\beta}_{i1} + \\ & a_{i1}\dot{x}_{i1} - \dot{w}_{i1}^T \rho_{i1}(q_i) - w_{i1}^T \frac{\partial \rho_{i1}(q_i)}{\partial q_i} \dot{q}_i + a_{i2}x_{i2} - w_{i2}^T \rho_{i2}(q_i, \dot{q}_i) \end{aligned} \quad (5.8)$$

### 5.2.2.1 Modelo con dos neuronas, caso 1

Este modelo de red neuronal es el (4.9), expuesto en la sección 4.4.1; la neurona que identifica a la velocidad de la articulación está interconectada únicamente con la velocidad real, preprocesada por la función sigmoideal. A diferencia del modelo utilizado en la sección 5.2.1.1, la posición de la articulación sí es aproximada por una neurona-estado. El control queda definido por la siguiente ecuación:

$$\begin{aligned} u_i = & -k_{i1}k_{i2}(x_{i1} - \beta_{i1}) - (k_{i1} + k_{i2})(x_{i2} - \dot{\beta}_{i1}) + \ddot{\beta}_{i1} + \\ & a_{i1}\dot{x}_{i1} - \dot{w}_{i11} \rho_{i11}(q_i) - w_{i11} \frac{\partial \rho_{i11}(q_i)}{\partial q_i} \dot{q}_i + a_{i2}x_{i2} - w_{i22} \rho_{i22}(\dot{q}_i) \end{aligned} \quad (5.9)$$

### 5.2.2.2 Modelo de dos neuronas, caso 2

Este modelo de red neuronal es el (4.10), expuesto en la sección 4.4.2; la neurona que identifica a la velocidad de la articulación está doblemente interconectada, a la posición y a

la velocidad, reales, preprocesadas por su función de activación. Además, como en el modelo previo, la posición es identificada por una neurona con una interconexión. El control queda definido por la siguiente ecuación:

$$u_i = -k_{i1}k_{i2}(x_{i1} - \beta_{i1}) - (k_{i1} + k_{i2})(x_{i2} - \dot{\beta}_{i1}) + \ddot{\beta}_{i1} + a_{i1}\dot{x}_{i1} - \dot{w}_{i11}\rho_{i11}(q_i) - w_{i11}\frac{\partial\rho_{i11}(q_i)}{\partial q_i}\dot{q}_i + a_{i2}x_{i2} - w_{i21}\rho_{i21}(q_i) - w_{i22}\rho_{i22}(\dot{q}_i) \quad (5.10)$$

## 5.3 Control del manipulador robótico de dos articulaciones

Los modelos neuronales propuestos en este capítulo, para el control descentralizado de manipuladores robóticos, se prueban, a través de simulaciones, para regulación y seguimiento de trayectorias del sistema (3.2), un manipulador robótico de dos uniones revolutivas.

Las pruebas consisten en simular computacionalmente el comportamiento en lazo cerrado del manipulador robótico y del modelo neuronal, en el esquema de identificación y control descentralizado.

### 5.3.1 Evaluación experimental

Para llevar a cabo la evaluación experimental las simulaciones son implementadas en MATLAB<sup>®</sup> y Simulink<sup>®</sup> (marcas registradas de MathWorks, Inc.). Los resultados de las pruebas son evaluados mediante gráficas de comportamiento del error de control y de la señal de control.

El modelo matemático (3.2), del manipulador robótico es implementado a través de un programa en MATLAB<sup>®</sup> cada modelo neuronal es implementado en Simulink<sup>®</sup>, el esquema completo se ejecuta en Simulink<sup>®</sup>. Los valores de los parámetros del manipulador robótico son  $m_1 = 1$  Kg,  $m_2 = 1$  Kg,  $l = 1$  m. Para las redes neuronales los valores de  $a_{ij}$ , y  $\gamma_{ijk}$  son previamente seleccionados de acuerdo al rango de frecuencia de la señal de referencia.



Primero se evalúa el esquema de control neuronal descentralizado para manipuladores robóticos, con los modelos (4.5), (4.6), (4.9) y (4.10); el propósito es probar la efectividad del control basado en cada uno de los modelos, y compararlos entre sí. Posteriormente un control neuronal descentralizado es comparado con un control tipo PD (realimentación con ganancia proporcional y derivativa del error); primero en tareas de regulación; luego, de seguimiento de trayectorias; y finalmente, implementando perturbaciones, tanto en regulación como en seguimiento de trayectorias.

### 5.3.1.1 Descripción de los experimentos

En el primer experimento, se simula la evolución del comportamiento del manipulador robótico, controlado por la señal generada a través de la ley de control que corresponde al modelo neuronal descentralizado, en turno. Primero, se prueba el modelo (4.5) con la ley de control (5.5); segundo, el (4.6) con la ley de control (5.6); tercero, el (4.9) con la ley de control (5.9); y cuarto, el (4.10) con la ley de control (5.10). Se efectúan las simulaciones con los cuatro modelos controlando al manipulador robótico para que cada una de sus articulaciones siga la señal:  $r = -\cos(4t)$ . Los parámetros fijos en las redes neuronales tienen el mismo valor, excepto los que, por la estructura del modelo, deban ser nulos. Se observa y compara el desempeño de los controladores, a través de gráficas del error de control y de la señal de referencia.

El segundo experimento, es diseñado para comparar el desempeño del control neuronal descentralizado con el de un control PD, se realizan el experimento en tres fases: la de regulación, la de seguimiento y la del sistema con perturbaciones.

Las perturbaciones son implementadas mediante cambios súbitos en los parámetros físicos y geométricos del manipulador robótico; se pretende simular cambios en la carga variando la masa del segundo eslabón, y otras perturbaciones variando la masa del primer eslabón y la longitud de los dos eslabones; todas las perturbaciones se introducen por un tiempo de dos décimas de segundo.

En el primer evento de la fase de perturbaciones se introduce un cambio en  $m_2$  de 1 kg a 10 kg, en  $t = 10$  s, mientras se ejecuta una tarea de regulación; la misma perturbación se introduce en el tercero y cuarto evento, con tareas de seguimiento de trayectoria. En el



segundo evento, transcurre una tarea de regulación, cuando múltiples perturbaciones son simuladas, los cambios instantáneos son los siguientes: en  $t = 7$  s,  $m_2 = 10$  kg; en  $t = 8$  s,  $m_1 = 10$  kg; en  $t = 9$  s,  $l = 5$  m; unos segundos después se invierte el orden de éstas perturbaciones: en  $t = 12$  s,  $l = 5$  m; en  $t = 13$  s,  $m_1 = 10$  kg; en  $t = 14$  s,  $m_2 = 10$  kg.

Las ganancias en el control PD, y los parámetros fijos en las redes neuronales, son previamente seleccionados; el criterio para éstos es aproximar el acotamiento del error que se alcanza con uno y otro controlador; en las simulaciones que se muestran, el error en estado estacionario no es mayor a dos centésimas de radián.

Las señales de referencia son, para cada evento del segundo experimento, de acuerdo a la siguiente tabla:

Figuras en Apéndice A:	Señal de Referencia:	Tarea de Control:
Figura A.3.17, Figura A.3.18	[0; 0]	Regulación
Figura A.3.19, Figura A.3.20	[-1; -1]	Regulación
Figura A.3.21, Figura A.3.22	[0; -1]	Regulación
Figura A.3.23, Figura A.3.24	[-1; 0]	Regulación
Figura A.3.25, Figura A.3.26	$[-\cos(t); -\cos(t)]$	Seguimiento
Figura A.3.27, Figura A.3.28	$[-\cos(4t); -\cos(4t)]$	Seguimiento
Figura A.3.29, Figura A.3.30	$[-\cos(t); -\cos(4t)]$	Seguimiento
Figura A.3.31, Figura A.3.32	$[0; -\cos(4t)]$ , $[-\cos(4t); 0]$	Seguimiento
Figura A.3.33, Figura A.3.34	[-1; -1]	Regulación c/Perturbación
Figura A.3.35, Figura A.3.36	[0; 0]	Regulación c/Perturbación
Figura A.3.37, Figura A.3.38	$[-\cos(t); -\cos(t)]$	Seguimiento c/Perturbación
Figura A.3.39, Figura A.3.40	$[-\cos(4t); -\cos(4t)]$	Seguimiento c/Perturbación

Tabla 5.1 Referencias empleadas para tareas de regulación y seguimiento.

### 5.3.1.2 Interpretación de resultados

En las gráficas de Figura 5.1 se muestra el error de control, con cada uno de los cuatro controles neuronales que se prueban en el primer experimento.

La rapidez con que converge el error a la región de acotamiento resulta la misma para los cuatro modelos; el tamaño de dicha región es un poco menor con el modelo (4.5), no más

de un 10% de reducción con respecto a los otros; las oscilaciones de la trayectoria del error sí difieren notablemente, la explicación se reserva para el análisis de las señales de control.

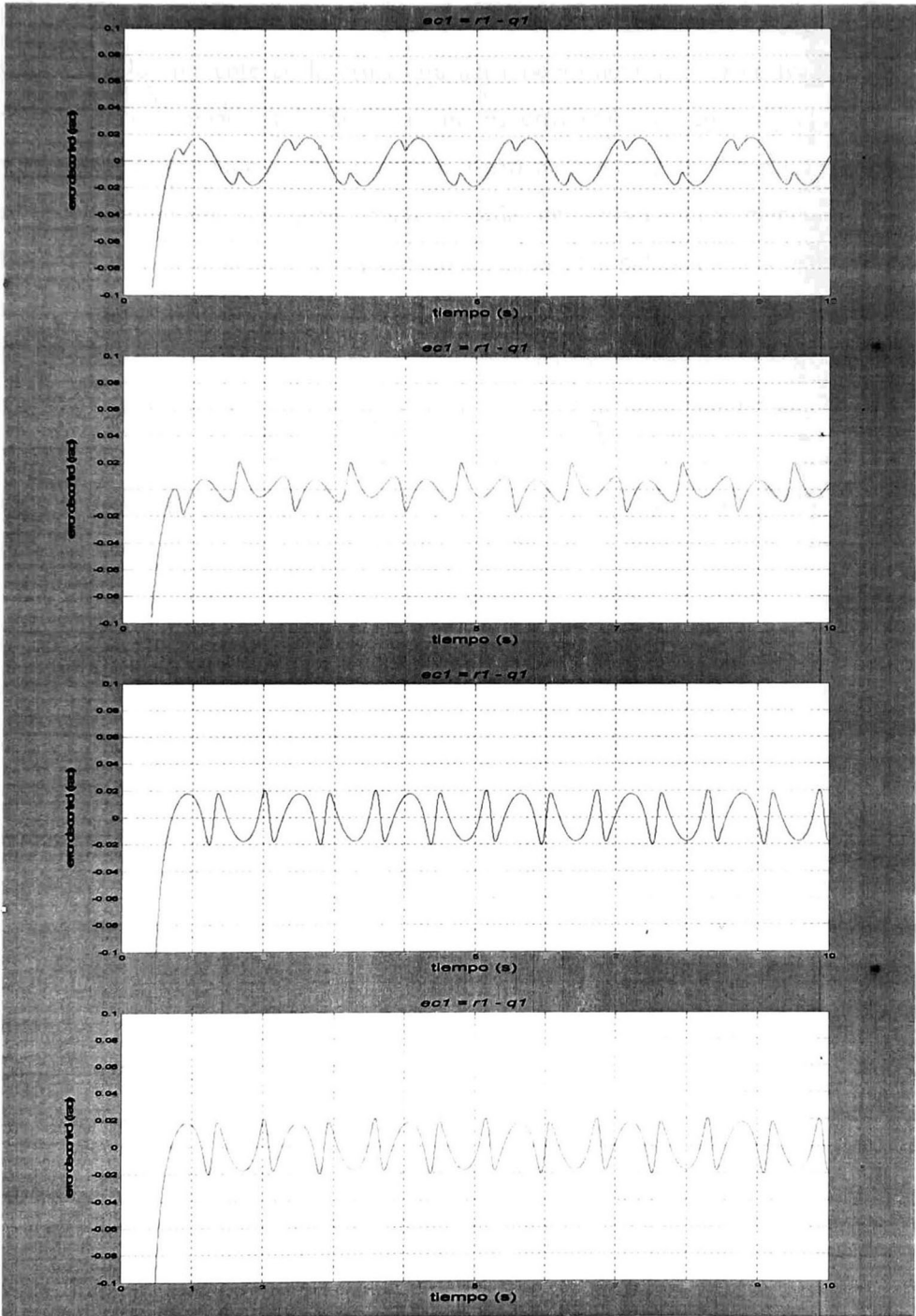


Figura 5.1. Error de control de  $q_1$ , con (4.5), (4.6), (4.9), (4.10).

Desde la observación del error, el uso de dos interconexiones en la segunda neurona no representa ventaja; se comparan los modelos así: (4.5) contra (4.6); y (4.9) contra (4.10).

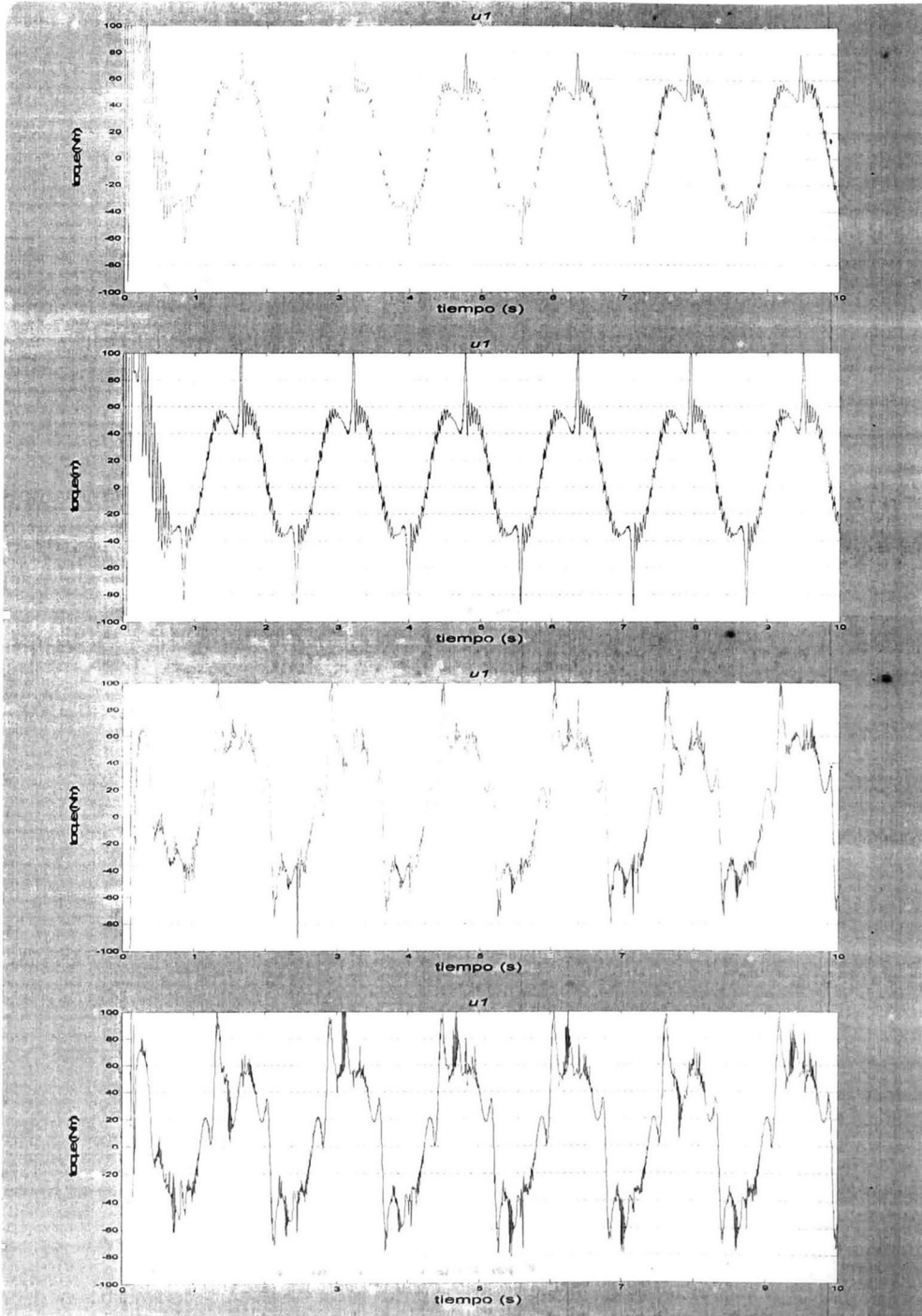


Figura 5.2. Control  $u_1$ , con (4.5), (4.6), (4.9), (4.10).

Las gráficas de Figura 5.2 muestran las señales de control que resultan con cada uno de los cuatro modelos en este experimento.

Con los modelos (4.9) y (4.10), que pertenecen al segundo enfoque de identificación, en el que la primer neurona aproxima a la posición de la articulación a través de una ecuación propia, resulta lo siguiente en la estructura del modelo neuronal: la derivada del estado-neurona, que corresponde a la posición, no es estrictamente igual al segundo estado-neurona, que corresponde a la velocidad. Permanentemente la dinámica adicional es forzada a ser eliminada, en consecuencia, la señal de control pierde mucho en suavidad. Puede observarse lo anterior al comparar las gráficas de señal de control de (4.9) y (4.10) contra las de (4.5) y (4.6), en Figura 5.2. La implementación del control en tiempo real puede complicarse, por la necesidad de producir señales de control, en este caso torques, con trayectorias menos suaves, con altas aceleraciones y desaceleraciones en la amplitud, y con mayor frecuencia cambios de signo en su derivada.

También, a partir de la observación de la señal de control, el uso de dos interconexiones en la segunda neurona no representa ventaja: la diferencia en magnitud, de mínimos y máximos locales próximos entre sí, es mayor con esos modelos. Lo anterior implica que el torque debe variar con mayor celeridad y a la vez el sistema debe responder con la misma celeridad. La comparación de los modelos, para este último análisis, es a través de las gráficas en Figura 5.2, así: (4.5) contra (4.6); y (4.9) contra (4.10).

En Apéndice A se encuentran las gráficas correspondientes a la segunda articulación, del análisis de estas resultan las mismas conclusiones que se obtienen para la primera articulación.

Los resultados siguientes son del segundo experimento, que se efectúa para comparar el desempeño del control neuronal descentralizado con el de un control PD. En base a los resultados del primer experimento se opta por trabajar con el modelo neuronal más simple, el (4.5), al que le corresponde el control (5.5).

Se muestran en Figura 5.3 uno de los resultados de la fase de tareas de regulación, en particular cuando la referencia para la primera articulación es de menos un radián, y cero para la segunda articulación; el total de las gráficas se encuentran en Apéndice A. Las trayectorias en línea continua pertenecen al uso del control neuronal (RN); y las de línea segmentada al control PD.



En todos los casos, en esta fase del experimento, la región de acotamiento del error se alcanza más rápido con el control tipo PD. Implica esto un costo en la señal de control; con el control PD el valor de amplitud máximo es, en general, de mayor magnitud. Las dos diferencias señaladas son consecuencia de las altas ganancias del control PD, en contraste con las pequeñas ganancias del error y de la derivada del error implícitas en el método de síntesis del control neuronal.

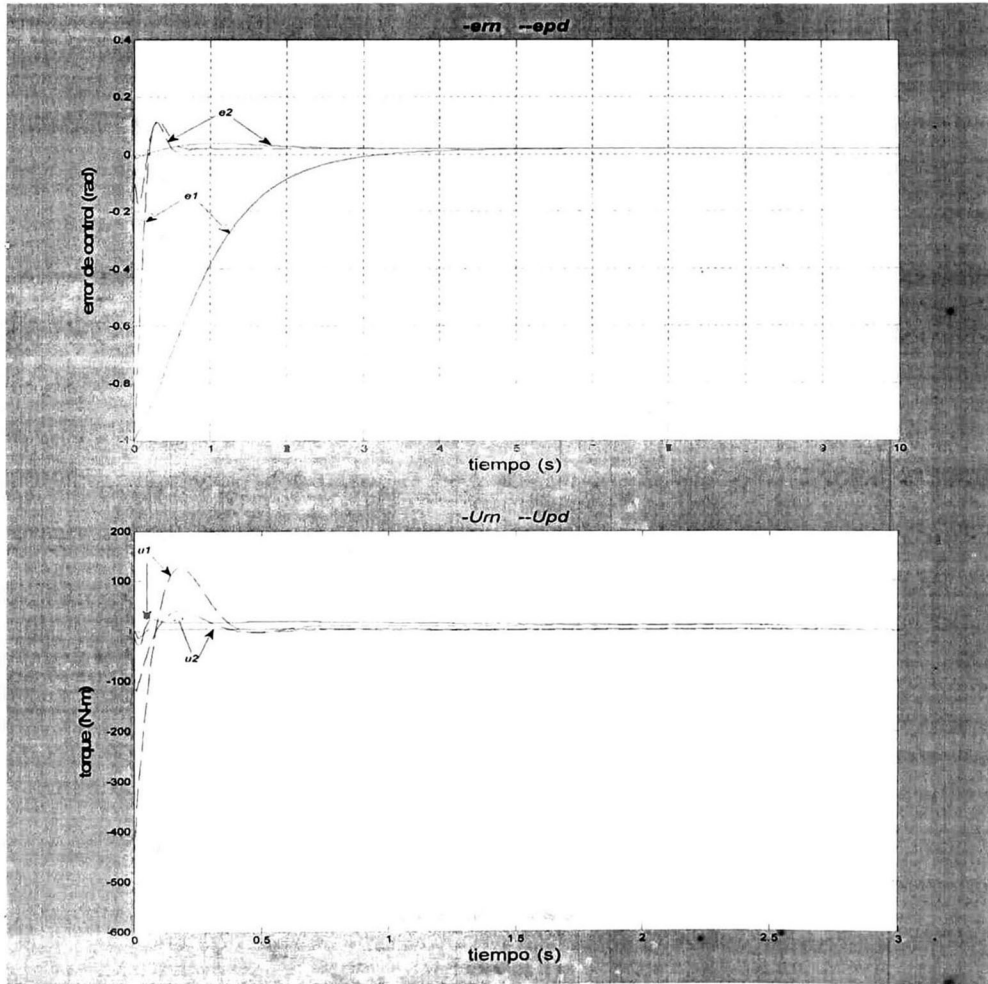


Figura 5.3. Error de control (RN vs. PD), y Señal de Control (RN vs. PD).

La segunda fase de este experimento, para tareas de seguimiento de trayectorias, muestra un resultado en Figura 5.4. Las diferencias encontradas en la fase de regulación, también se observan en seguimiento de trayectorias. La mayor magnitud, en el valor de la amplitud



máxima, con el control tipo PD es más significativa, aunque no se visualiza de forma completa, debido a la escala en las gráficas. Por ejemplo, para el caso que se presenta en la gráfica el sobretiro del control PD para la primera articulación es de aproximadamente 2,500 N-m, mientras que en el control neuronal no rebasa 100 N-m.

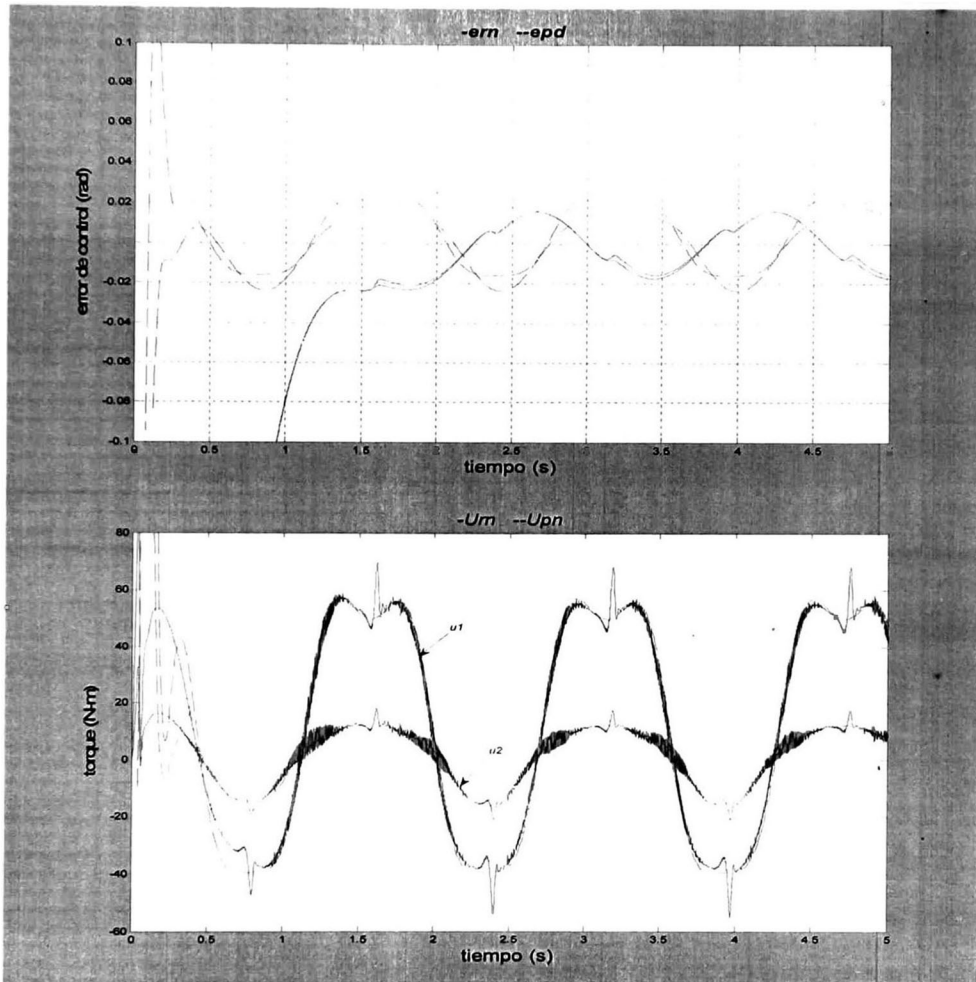


Figura 5.4. Error de control (RN vs. PD), y Señal de Control (RN vs. PD).

En el último evento de esta fase del experimento, las referencias se intercambian a la mitad del tiempo en que transcurre la simulación, una de las referencias es constante mientras la otra es una señal senoidal. Se observa, en Figura 5.5, que el error en estado estacionario es menos sensible al cambio de referencia cuando se usa el control neuronal. Finalmente, es notable que las señales de control del PD; y del RN; en estado estacionario, se aproximan.

La diferencia fundamental es la pérdida de suavidad visible en la señal del control neuronal, en tareas de seguimiento de trayectorias, como consecuencia de la rápida y obligada adaptación de los pesos de la red neuronal de forma continua.

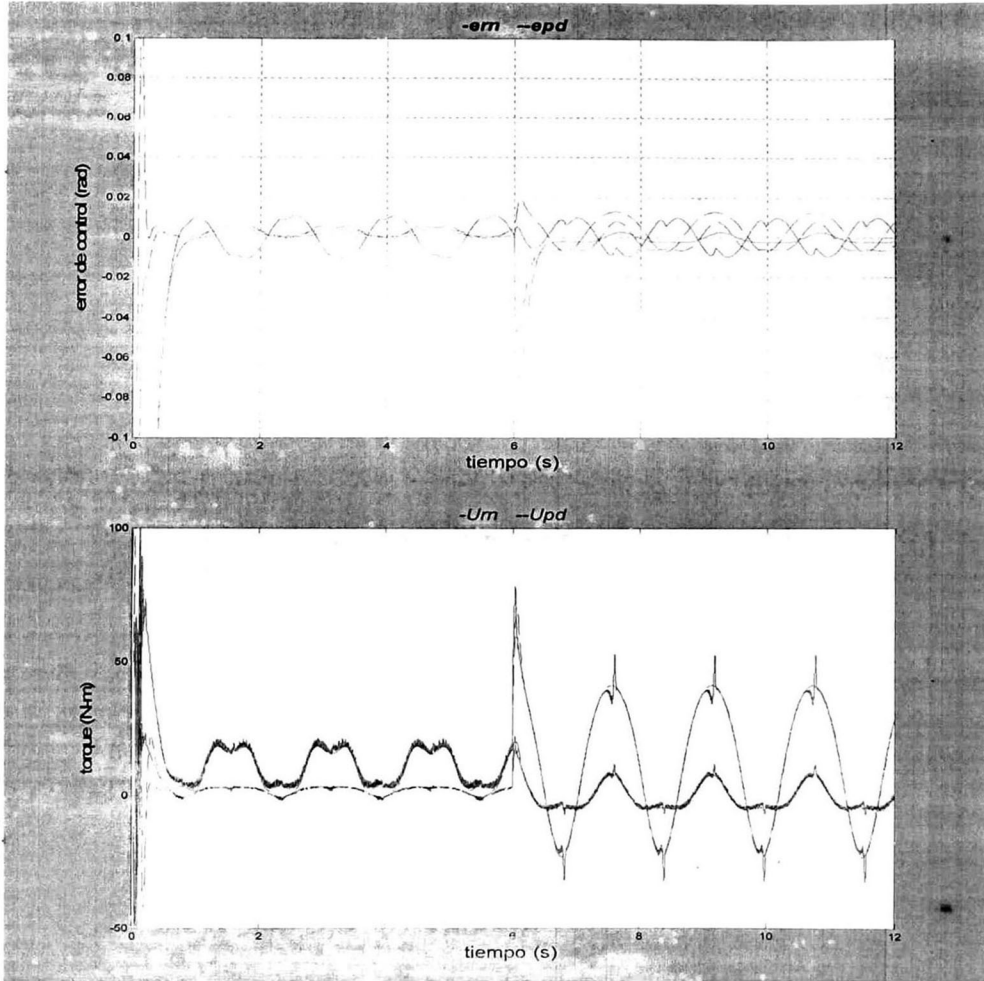


Figura 5.5. Error de control (RN vs. PD), Señal de control (RN vs. PD).

Los resultados de la última fase de este experimento se muestran a continuación. Las gráficas para regulación con perturbación se muestran en Figura 5.6. Se observa que con el control neuronal el crecimiento del error es mucho menor y se reestablece más lentamente; también se observa que el control neuronal ejerce una adaptación a las perturbaciones y el error de control crece en menor medida cuando se vuelven a presentar las mismas

perturbaciones; ante la primera aparición de cada perturbación crece el rango de variación de los valores de los pesos sinápticos.

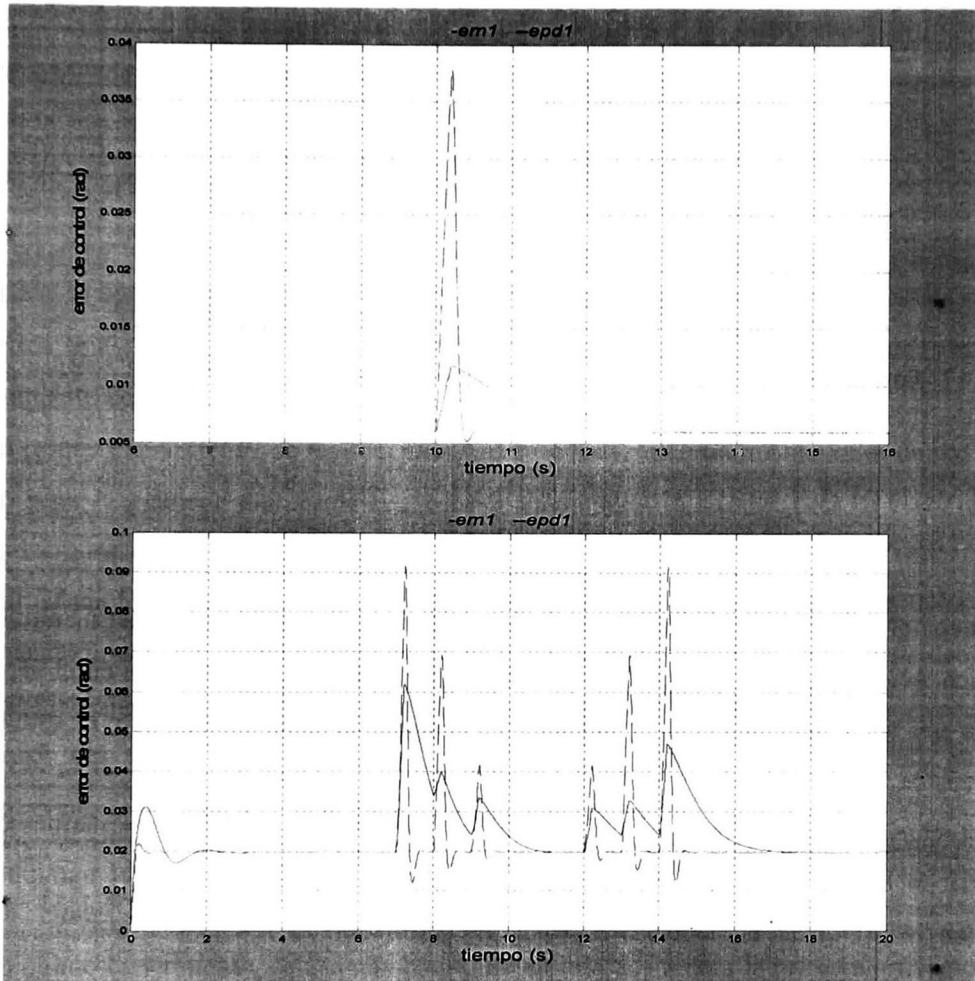


Figura 5.6. Error de control (RN vs. PD), en tareas de regulación con perturbaciones.

En las gráficas de Figura 5.7, se observa que en tareas de seguimiento, el control neuronal tiene mucha mayor capacidad de adaptación a las perturbaciones. El error presenta una variación considerablemente pequeña y la recuperación del error tras la perturbación es muy rápida. Lo anterior significa una ventaja importante del control neuronal.

En Apéndice A, se exhiben gráficas de resultados para cada caso de experimentación de Tabla 5.1, donde se incluyen diferentes tareas de regulación y de seguimiento de trayectoria. Los resultados que se muestran en este capítulo son los más representativos.

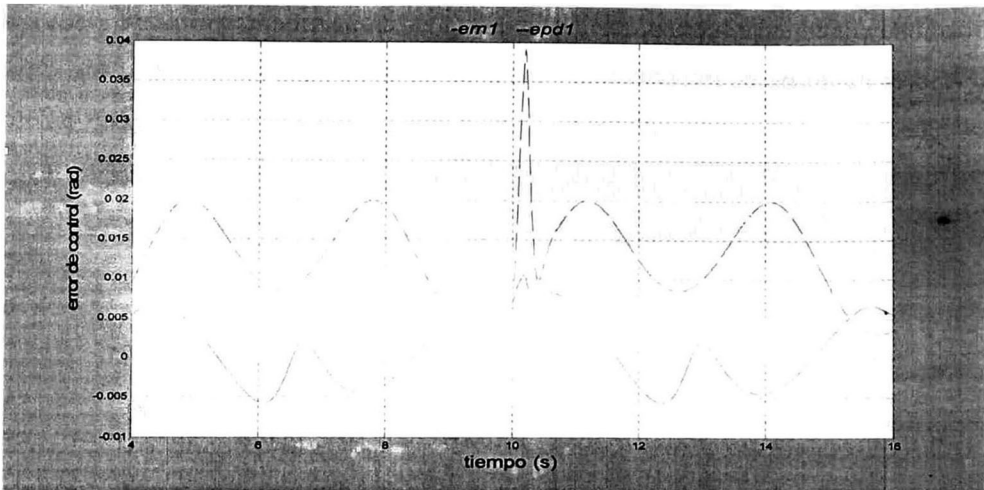


Figura 5.7. Error de control (RN vs. PD), en tarea de seguimiento con perturbaciones.

## 5.4 Control del manipulador robótico tipo ANAT

Se prueba el esquema de control neuronal descentralizado, con el modelo del manipulador robótico ANAT, de siete articulaciones; la descripción de éste se encuentra en el capítulo tres de esta tesis. Se examina el desempeño del esquema de control, propuesto en esta tesis, en tareas de regulación y de seguimiento de trayectorias.

Las pruebas se desarrollan mediante simulación computacional, del comportamiento del sistema en lazo cerrado. Se integra el esquema completo, que incluye: el manipulador robótico, la referencia, el modelo neuronal, y el controlador.

### 5.4.1 Evaluación experimental

Todos los componentes del sistema en lazo cerrado se integran en un programa que se ejecuta en la plataforma de Simulink®. El manipulador robótico es considerado con las siguientes parámetros físicos y geométricos: cada uno de sus eslabones tiene una masa de 0.5 Kg; las longitudes nominales de los mismos son, en el orden usual con respecto a la base: 0.57 m, 0.16641 m, 0.12278 m, 0.0845 m, 0.1756 m, 0.25107 m y 0.02343 m; los coeficientes de fricción cinética, en el mismo orden: 0.1, 0.2, 0.1, 0.2, 0.1, 0.2, y 0.1.



El propósito principal de estos experimentos es probar la efectividad del esquema de control neuronal descentralizado con un modelo que contiene mayor número de interacciones dinámicas, con mayor número de subsistemas, todos interactuando entre sí.

La efectividad del esquema propuesto es examinada a partir del comportamiento del error de control y de la señal de control.

### 5.4.1.1 Descripción de los experimentos

Para probar la efectividad del control neuronal descentralizados con el manipulador robótico tipo ANAT, se realizan siete experimentos; incluyen tareas de regulación y de seguimiento de trayectorias.

En el primer experimento, se regulan todas las articulaciones a un mismo valor de posición; -1 m para la primera articulación, que es prismática, y de -1 rad para el resto de las articulaciones, que son giratorias.

En el segundo experimento, se efectúa regulación, con valores diferentes para cada articulación; se lleva la articulación prismática hacia el cero; para las otras articulación la referencias constantes son: -3 rad, -2 rad, -1 rad, 1 rad, 2 rad, 3 rad.

A partir del tercer experimento, la tarea es de seguimiento de trayectorias; en este experimento se emplea la misma señal de referencia en todas las articulaciones:  $r = -\cos(t)$ .

El cuarto experimento, también con una misma señal de referencia para todas las articulaciones:  $r = -\cos(4t)$ .

En el quinto experimento, se usa una constante de referencia para la primera articulación y las dos señales senoidales ya mencionadas se alternan para el resto de las uniones.

El sexto experimento, un caso se más interesante, consiste en usar señales de referencia senoidales que difieren entre sí para una y otra articulación, en frecuencia, en amplitud y/o en fase.

Finalmente, en el séptimo experimento, las articulaciones siguen trayectorias particulares llevando al efector final a seguir una trayectoria rectangular específica en un lapso de 12 segundos. La generación de las referencias es realizada por el programa utilizado por Le Boudec y colaboradores [23], que se emplea con la autorización de los autores. Los

resultados del control neuronal descentralizado, se comparan con los que ofrecen dichos investigadores, de un control adaptable basado en la teoría de Lyapunov.

Para los siete experimentos se grafican las trayectorias de posición de las articulaciones y del error de control.

### 5.4.1.2 Interpretación de resultados

Se observa el buen desempeño del control neuronal descentralizado en tarea de regulación. En los dos primeros experimentos, con el manipulador robótico de siete articulaciones, el control es tan efectivo como cuando se aplica al manipulador robótico de dos articulaciones giratorias; no cambia la efectividad del control ante un mayor número de articulaciones y ante la unión prismática.

Para tareas de regulación, la convergencia del error de control, hacia una región de acotamiento muy pequeña, en la vecindad de cero; se presentan algunos de los resultados en las gráficas de Figura 5.8. Aparecen las trayectorias para las siete uniones.

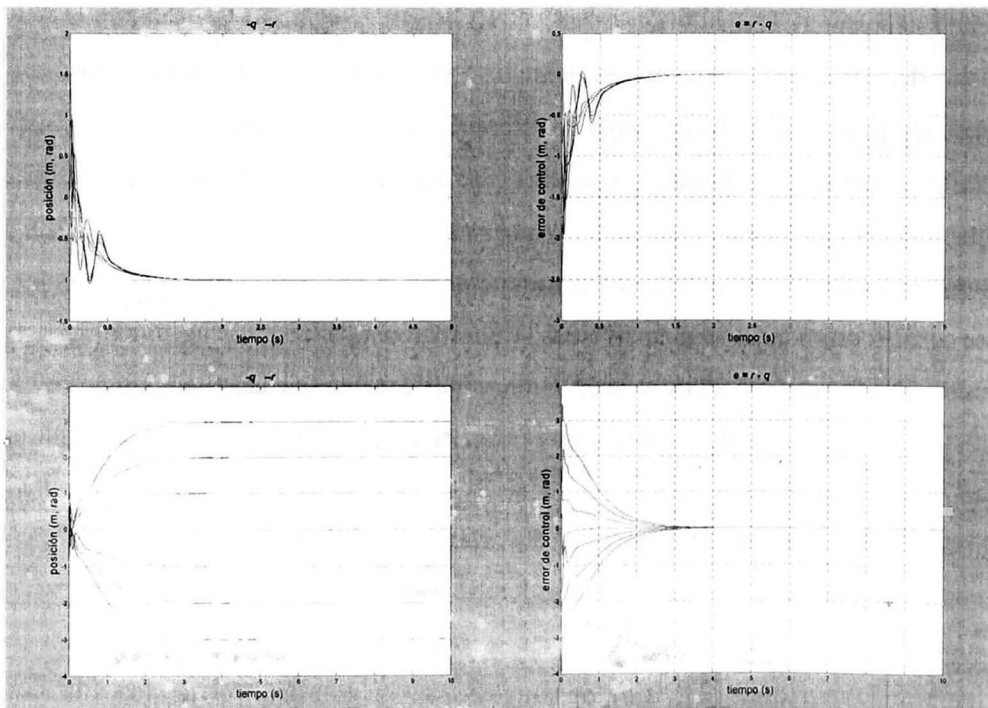


Figura 5.8. Regulación, error de regulación; manipulador robótico ANAT.

La convergencia del error de control, hacia una pequeña región de acotamiento, se observa también en los resultados de seguimiento de trayectorias, del tercero al sexto experimento. Las gráficas de los experimentos dos y tres, se muestran a continuación; en la parte inferior las que corresponden al seguimiento de una referencia de más alta frecuencia. En ambos casos se tiene la misma referencia para las siete articulaciones.

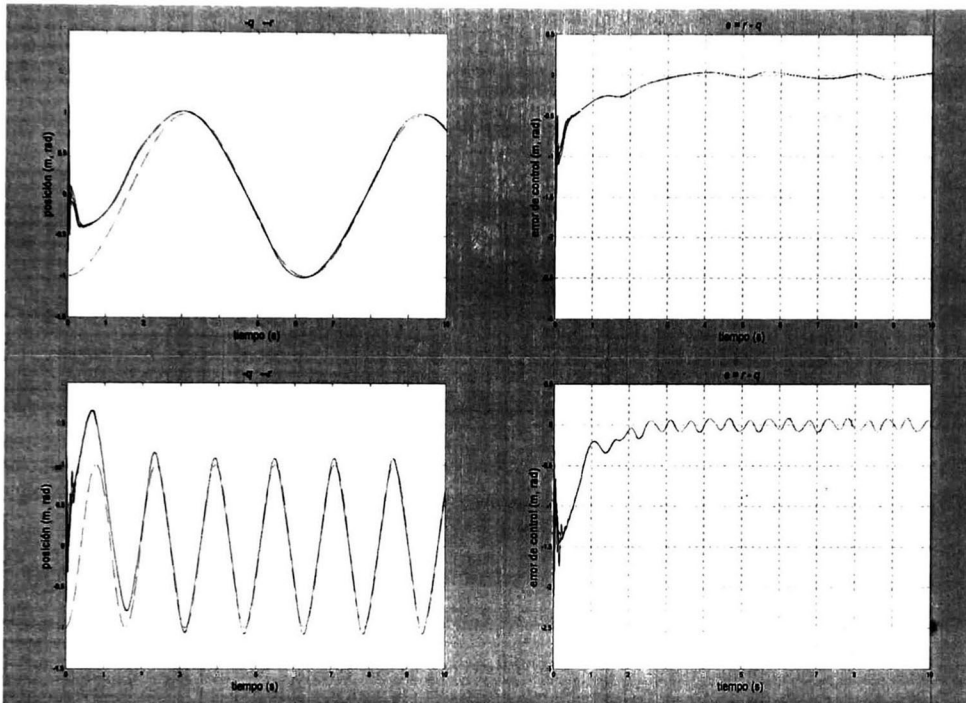


Figura 5.9. Seguimiento de trayectoria, error de seguimiento; manipulador robótico ANAT.

Del análisis de las gráficas en Figura 5.9, particularmente de las que corresponden al error de seguimiento, se establece lo siguiente: cuando las articulaciones siguen señales de referencia con frecuencias mayores, hay un incremento en el tamaño de la región de acotamiento del error; esto sucede si se mantienen los mismos valores para los parámetros fijos de las redes neuronales (coeficientes en las ecuaciones de estado e índices de aprendizaje para los pesos sinápticos). Sin embargo, si desea reducirse el error puede optarse por aumentar los valores de dichos parámetros.

El control neuronal descentralizado es capaz de lograr que las uniones sigan cualquier trayectoria suave y acotada.



En las siguientes gráficas se muestra el error de seguimiento en experimentos con distintas referencias para las trayectorias de la posición de las uniones. En la parte superior, de Figura 5.10, cuando se usa una referencia constante para la primera unión y otras dos referencias sinusoidales para el resto de las uniones; en la parte inferior, para cada unión la referencia es distinta, en amplitud y en frecuencia. Las gráficas de trayectoria de las uniones y de las neuronas correspondientes pueden observarse en Apéndice A.

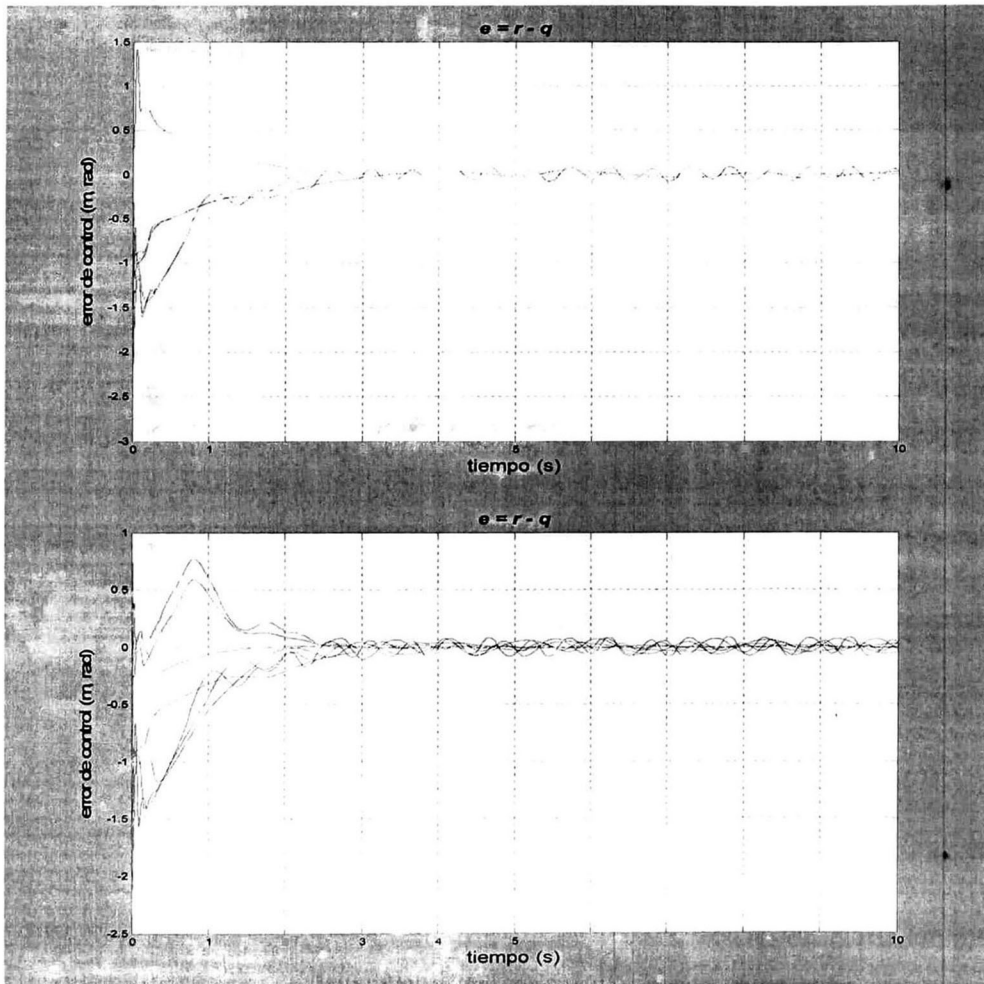


Figura 5.10. Error de seguimiento; manipulador robótico ANAT.

Para algunas uniones del modelo del manipulador robótico, el error de seguimiento oscila dentro de una región un poco mayor, esto es consistente con la mención anterior, respecto al efecto que tiene la alta frecuencia de la referencia.



Por último, para el séptimo experimento, las uniones deben seguir determinadas trayectorias (ver Figura 5.11), para que en el espacio de trabajo resulte una trayectoria rectangular. Se selecciona este experimento para probar el control neuronal descentralizado ante trayectorias con nuevas complicaciones.

La trayectoria en el espacio de trabajo, que se muestra en Figura 3.4, es una trayectoria rectangular que inicia en la esquina inferior derecha, y se inicia con el trazo del lado derecho; cada lado se recorre en un tiempo de tres segundos, resultando de doce segundos el tiempo de desarrollo de la tarea completa. Las trayectorias en el espacio de las uniones son calculadas a través de un programa diseñado por Le Boudec y colaboradores [23], este programa proporciona el modelo neuronal la referencia para cada una de las uniones.

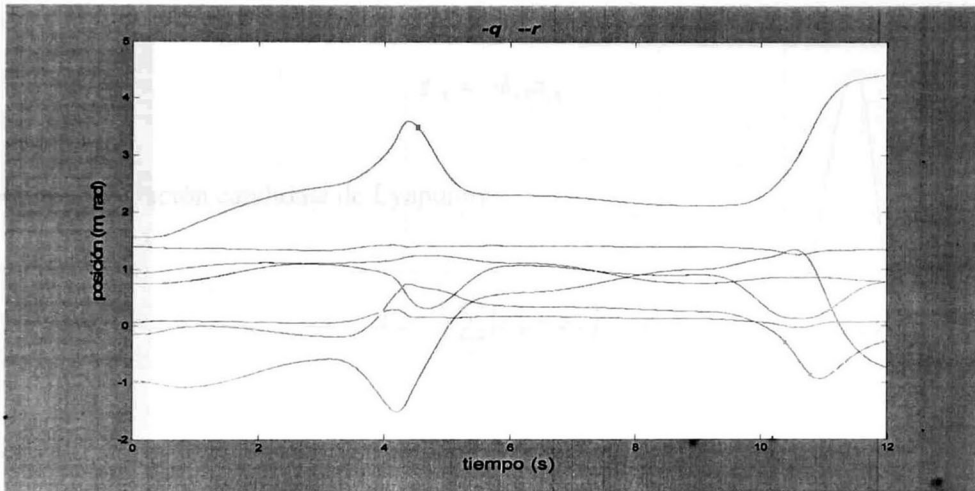


Figura 5.11. Trayectorias de las uniones, para una trayectoria rectangular en el espacio de trabajo.

La planeación de trayectorias debe lograr que las articulaciones se muevan de modo que el efector final trace el rectángulo y que el manipulador no colisione con los obstáculos, ni consigo mismo. Por ejemplo, la posición de la segunda articulación, entre tres y seis segundos, mientras se traza el lado izquierdo del rectángulo, pudiera adoptar valores mayores a  $\pi/2$ , en el procedimiento de planeación de trayectoria esto queda restringido, debido a que pudiera colisionar el manipulador con el obstáculo cilíndrico a la izquierda. Además, entre seis y doce segundos, periodo en el que se dibuja el lado derecho, también pudieran moverse hacia valores mayores a  $\pi/2$ , colisionando con el obstáculo de la derecha;

esto también se evita, a través de costos en la función objetivo y de la redundancia del manipulador. La articulación número cuatro tiene la más alta probabilidad de colisionar; entre tres y seis segundos, ocurriría si se mueve más allá de 0.8 rad; esto se evita de igual manera que con la articulación número dos; igualmente, entre los nueve y los doce segundos, pudiera colisionar si se mueve más bajo de -1 rad, lo cual también se logra evitar. Las gráficas de error de seguimiento están en Figura 5.12; las correspondientes al control neuronal descentralizado (CND) en la parte superior, y las del controlador central adaptable (CCA), en la parte inferior. Se establece que el control neuronal descentralizado, propuesto en esta tesis, es capaz de lograr un buen desempeño, ante la tarea de que el efector final trace una trayectoria no necesariamente suave.

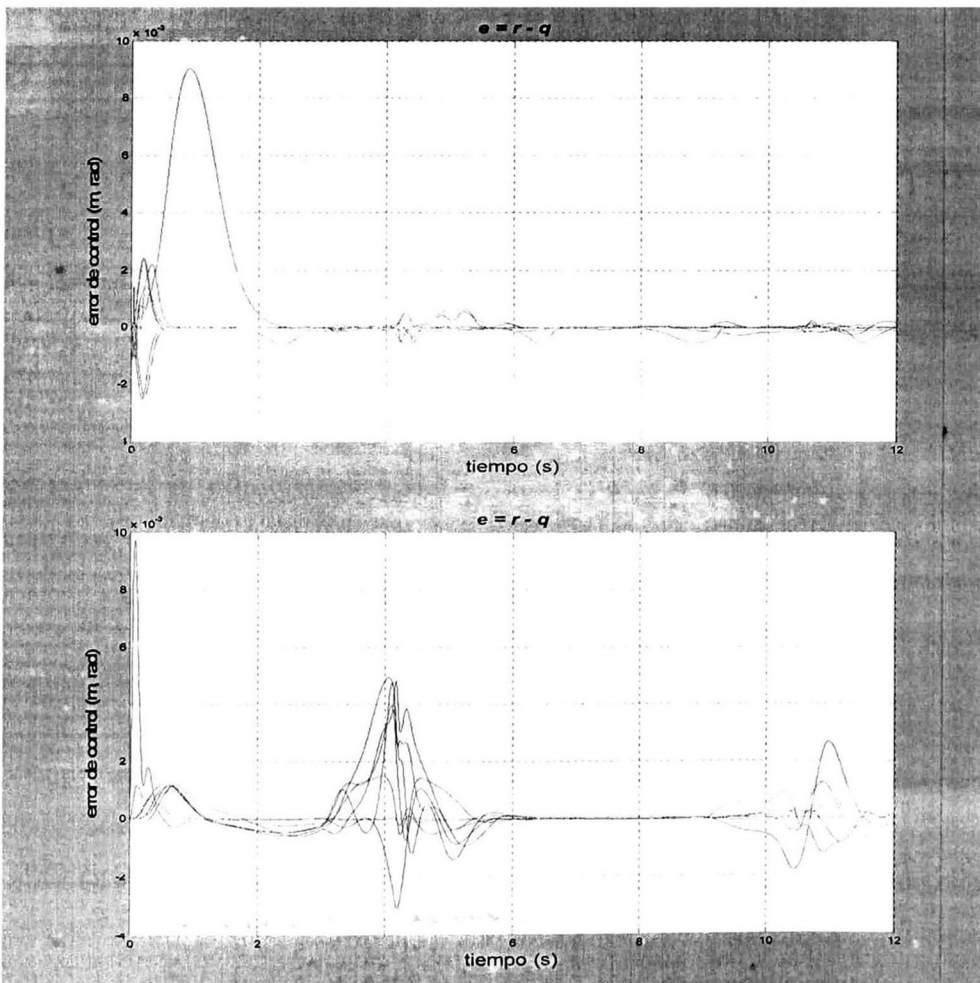


Figura 5.12. Error de seguimiento, con CND y con CCA.

## 5.5 Análisis de estabilidad

En la sección 5.1 se establece que el error de control del manipulador robótico cumple la siguiente desigualdad:

$$|\chi_{i1} - \beta_{i1}| \leq |x_{i1} - \chi_{i1}| + |x_{i1} - \beta_{i1}|.$$

Cada controlador del modelo neuronal es diseñado, tal como se explica en la sección 5.2, para que la señal de error  $z_{i1}$ , para la primera neurona, y la señal de error  $z_{i2}$ , para la segunda neurona, se comporten de la siguiente manera en lazo cerrado:

$$\begin{aligned}\dot{z}_{i1} &= -k_{i1}z_{i1}, \\ \dot{z}_{i2} &= -k_{i2}z_{i2}.\end{aligned}$$

Se define una función candidata de Lyapunov:

$$V = \frac{1}{2} \sum_{i=1}^N (z_{i1}^2 + z_{i2}^2).$$

La derivada en la trayectoria de las variables de error resulta:

$$\dot{V} = \sum_{i=1}^N (-k_{i1}z_{i1}^2 - k_{i2}z_{i2}^2).$$

Esto garantiza la estabilidad global del modelo neuronal. Por la otra parte, considerando la estabilidad de la identificación, cuya demostración se encuentra en el capítulo uno, se garantiza la estabilidad del sistema completo en lazo cerrado. En el caso ideal, sin error de modelado, el error de control del manipulador converge a cero; con existencia de error de modelado, que es el caso real, el error de control del sistema converge a una región acotada.

# Capítulo 6

## Conclusiones y trabajo futuro

### 6.1 Conclusiones

Los resultados obtenidos en el desarrollo de este trabajo de tesis permiten concluir que los sistemas de manipuladores robóticos, completamente actuados, pueden ser controlados, en el espacio de la tarea, bajo un enfoque neuronal descentralizado.

La dinámica de cada articulación del manipulador robótico puede ser aproximada, convenientemente para el esquema de control propuesto, con una red neuronal recurrente de primer orden, local y con una sola interconexión (en la topología serie-paralelo), si se ejecuta el algoritmo de aprendizaje permanentemente en línea.

Con el modelo neuronal para identificación descentralizada, que se propone en esta tesis, se consigue un error de identificación finalmente acotado; con una región de acotamiento en la vecindad de cero.

El modelo neuronal que se propone en esta tesis posee una estructura afín en el control; el control es diseñado sobre éste modelo; se garantiza así un error de control que converge asintóticamente a una pequeña región de acotamiento, la misma que se obtiene para el error de identificación.

Como resultado del ajuste en línea de parámetros del modelo neuronal se obtiene un control adaptable robusto ante incertidumbres, perturbaciones y variaciones en el tiempo del modelo del manipulador robótico.

El enfoque propuesto resulta atractivo para ser implementado en tiempo real por el reducido costo de tiempo computacional con la operación en paralelo de dispositivos controladores locales.

## **6.2 Trabajo futuro**

A futuro, este proyecto de investigación ha de continuar con la implementación en tiempo real del esquema de control propuesto, particularmente para el manipulador robótico tipo ANAT, de siete articulaciones, cuyo modelo es utilizado en el desarrollo del presente trabajo de tesis.

Los resultados obtenidos con este trabajo sugieren la idea de completar un esquema de control descentralizado para manipuladores robóticos, en el espacio de la tarea; para ello es necesario integrar el esquema propuesto en esta tesis, con una solución al problema de la cinemática inversa, mediante redes neuronales u otra técnica de inteligencia computacional.



# Bibliografía

- [1] M. N. Ahmad, and J. H. S. Osman, "Application of proportional-integral sliding mode tracking controller to robot manipulators", *Proceedings of the 2003 IEEE Conference on Control Applications*, vol. 1, pp. 87-92, 2003.
- [2] P. Baldi, "Neural networks, orientations of the hypercube, and algebraic threshold functions", *IEEE Transactions on Information Theory*, vol. IT-34, pp. 523-530, 1988.
- [3] V. H. Benitez, "Control neuronal a bloques: aplicación a un motor de inducción de desplazamiento lineal" Tesis de Maestría, CINVESTAV del IPN, Unidad Guadalajara, México, Noviembre 2002.
- [4] V. H., Benitez, A. G. Loukianov, and E. N. Sanchez, "Neural Identification and Control of a Linear Induction Motor Using an  $\alpha$ - $\beta$  Model", *Proceedings of the American Control Conference*, Denver, Colorado, USA, pp. 4041-4046, June 2003.
- [5] M. A. Cohen, and S. Grossberg, "Absolute stability of global pattern formation and parallel memory storage by competitive neural networks", *IEEE Transactions on Systems, Man, and Cybernetics*, vol. SMC-13, pp 815-826, 1983.
- [6] G. Cybenko, "Approximations by superpositions of a sigmoidal function", *Math. Contr., Signals, Sys.*, vol. 2, pp. 303-314, 1989.
- [7] A. Dembo, O. Farotimi, and T. Kailath, "High order absolutely stable neural networks". *IEEE Transactions on Circuits and Systems*, vol. 38, no.1, pp 57-65, 1991.
- [8] L.-Ch. Fu, "Robust adaptive decentralized control of robot manipulators" *IEEE Transactions on Automatic Control*, vol. 37, no. 1, pp. 106-110, 1992.
- [9] K. S. Fu, R. C. González, y C. S. G. Lee. *Robótica: Control, Detección, Visión e Inteligencia*, 1ª ed., McGraw-Hill, México, 1993.
- [10] K.-I. Funahashi, "On the approximate realization of continuous mappings by neural networks", *IEEE Transactions on Neural Networks*, vol. 2, pp. 183-192, 1989.
- [11] R. Gourdeau, "Object-oriented programming for robotic manipulator simulation", *IEEE Robotics and Automation*, vol. 4, no. 3, pp. 21-29, 1997.

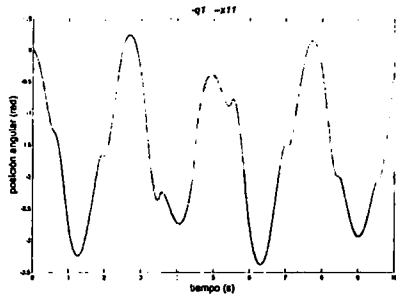
- [12] J. J. Hopfield, "Neurons with graded responses have collective computational properties like those of two state neurons", *Proceeding of the National Academy of Science*, vol. 81, pp 3088-3092, 1984.
- [13] S-H. Hsu, and L.-Ch. Fu, "Globally adaptive decentralized control of time-varying robot manipulators", *Proceedings of the 2003 IEEE International Conference on Robotics and Automation*, pp. 1458-1462, 2003.
- [14] S.-H. Hsu, and L.-Ch. Fu, "Globally fully adaptive decentralized control of robot manipulators", *IEEE Trans. on Decision and Control*, vol. 2, no. 1, pp. 1733-1738, 2002.
- [15] S. Huang, K. K. Tan, and T. H. Lee, "Decentralized control design for large-scale systems with strong interconnections using neural networks", *IEEE Transactions on Automatic Control*, vol. 48, no. 5, pp. 805-810, 2003.
- [16] P. A. Ioannou, and A. Datta, "Robust adaptive control: a unified approach" *Proceeding of the IEEE*, vol. 77, no. 12, pp 1736-1768, December 1991.
- [17] K. M. Lynch, L. E. Kavrachi, H. M. Choset, S. Hutchinson, G. Kantor, W. Burgard, S. Thrun, *Principles of Robot Motion*, 1<sup>a</sup> ed., MIT Press, USA, 2005.
- [18] Z.-P. Jiang, "New results in decentralized adaptive nonlinear control with output-feedback", *Proceedings of the IEEE 38<sup>th</sup> Conference on Decision & Control*, Phoenix, Arizona, USA, pp. 4772-4777, December 1999.
- [19] Y. Jin, "Decentralized adaptive fuzzy control of robot manipulators", *IEEE Transactions on Systems, Man, and Cybernetics-Part B: Cybernetics*, vol. 28, no. 1, pp. 47-57, February 1998.
- [20] A. Karakasoglu, S. I. Sudharsanan, and M K. Sundareshan, "Identification and decentralized adaptive control using dynamical neural networks with application to robotic manipulators" *IEEE Transactions on Neural Networks*, vol. 4, no. 6, pp. 919-930, 1993.
- [21] E. B. Kosmatopoulos, M. A. Christodoulou, and P. A. Ioannou, "Dynamical neural networks that ensure exponential identification error convergence", *IEEE Neural Networks*, vol. 10, no. 2, pp. 299-314, 1997.
- [22] E. B. Kosmatopoulos, M. M. Polycarpou, M. A. Christodoulou, and P. A. Ioannou, "High-order neural networks structures for identification of dynamical systems", *IEEE Transactions on Neural Networks*, vol. 6, no. 2, pp. 422-431, 1995.
- [23] B. Le Boudec, M. Saad, and V. Nerguizian, "Adaptive control of redundant robots" *8<sup>th</sup> International Conference on Modeling and Simulation of Electric Machines, Convertes and Systems*, ELECTRIMACS 2005, Hammamet, Tunisia, April 2005.

- [24] M. Liu, "Decentralized control of robot manipulators: nonlinear and adaptive approaches", *IEEE Transactions on Automatic Control*, vol. 44, no. 2, pp. 357-363, 1999.
- [25] M.-L. Ni, and M. J. Er, "Decentralized control of robot manipulators with coupling and uncertainties", *Proceedings of the American Control Conference*, Chicago, Illinois, pp. 3326-3330, June 2000.
- [26] P. Peretto, and J. J. Niez, "Long term memory storage capacity of multiconnected neural networks", *Biol. Cybernetics*, vol. 54, pp 53-63, 1986.
- [27] M. M. Polycarpou, and P. A. Ioannou, "On the existence and uniqueness of solutions in adaptive control systems" *IEEE Transactions on Automatic Control*, vol. 38, no. 3, pp 474-479, 1993.
- [28] L. J. Ricalde, "Control neuronal adaptable óptimo inverso con restricción en las entradas", Tesis de Doctorado, CINVESTAV del IPN, Unidad Guadalajara, México, Julio 2005.
- [29] G. A. Rovithakis, and M. A. Christodoulou, "Adaptive control of unknown plants using dynamical neural networks", *IEEE Transactions on Systems, Man, and Cybernetics*, vol. 24, no. 3, pp 400-412, 1994.
- [30] G. A. Rovithakis, and M. A. Christodoulou, *Adaptive Control with Recurrent High Order Neural Networks*, Springer-Verlag, New York, USA, 2000.
- [31] R. Safaric, and J. Rodic, "Decentralized neural-network sliding-mode robot controller" *Proceedings of the 26<sup>th</sup> Annual Conference of the IEEE Industrial Electronics Society*, vol. 2, no. 1, pp. 906-911, 2000.
- [32] R. H. Taylor, and D. Stoianovici "Medical robotics in computer-integrated surgery", *IEEE Robotics and Automation*, vol. 19, no. 5, pp. 765-781, 2003.
- [33] R.-J., Wai, "Tracking control based on neural network strategy for robot manipulators", *Neurocomputing*, 51, pp. 425-445, 2003.

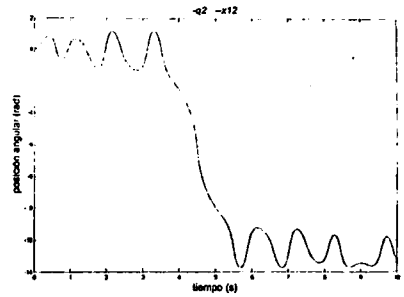
## **APÉNDICE A.**

### **Gráficas de resultados**

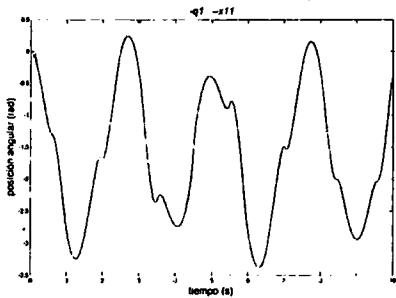
## A.1 Gráficas de identificación del manipulador robótico de dos uniones



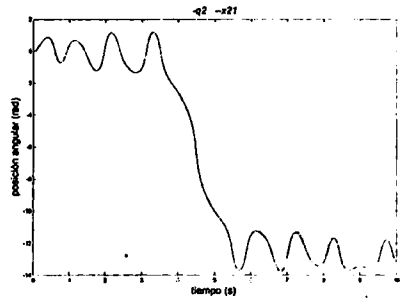
A.1.1. Identificación de  $q_1$ , con modelo (4.5).



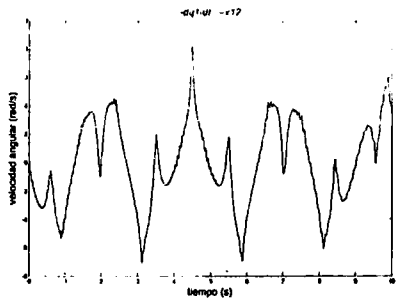
A.1.2. Identificación de  $q_2$ , con modelo (4.5).



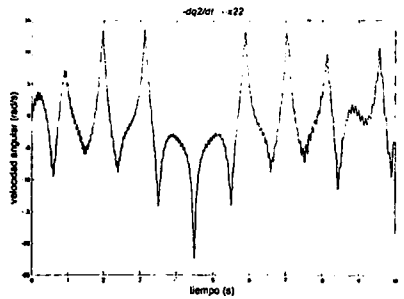
A.1.3. Identificación de  $q_1$ , con modelo (4.9).



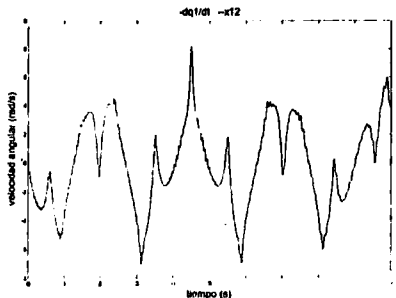
A.1.4. Identificación de  $q_2$ , con modelo (4.9).



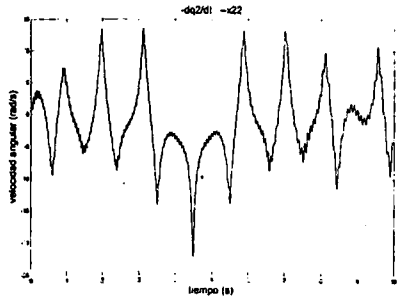
A.1.5. Identificación de  $dq_1/dt$ , con modelo (4.5).



A.1.6. Identificación de  $dq_2/dt$ , con modelo (4.5).

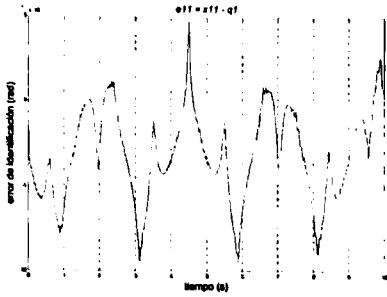


A.1.7. Identificación de  $dq_1/dt$ , con modelo (4.9).

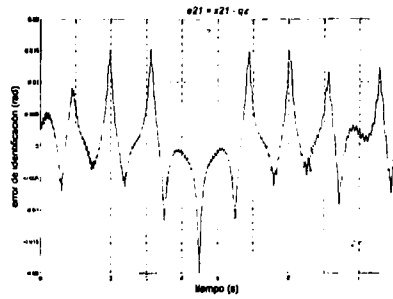


A.1.8. Identificación de  $dq_2/dt$ , con modelo (4.9).

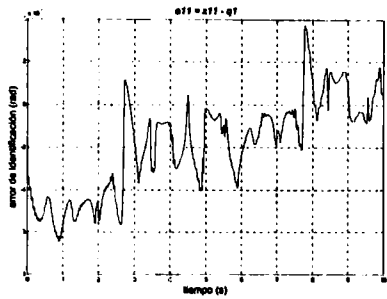




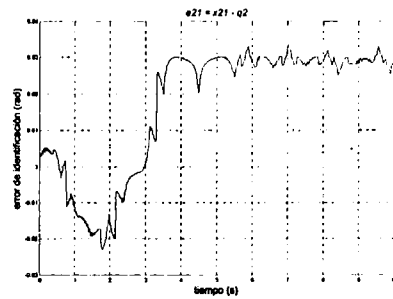
A.1.9. Error de identificación de  $q_1$ , con (4.5).



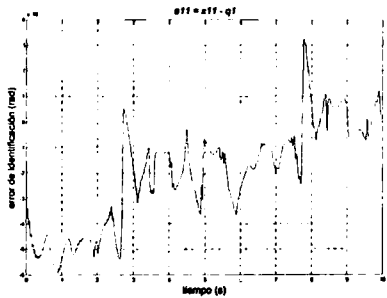
A.1.10. Error de identificación de  $q_2$ , con (4.5).



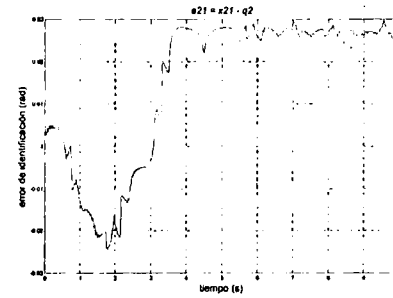
A.1.11. Error de identificación de  $q_1$ , con (4.6).



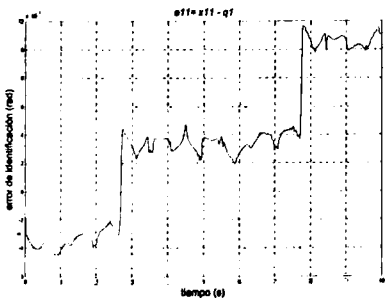
A.1.12. Error de identificación de  $q_2$ , con (4.6).



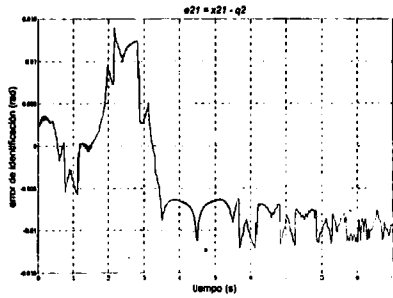
A.1.13. Error de identificación de  $q_1$ , con (4.7).



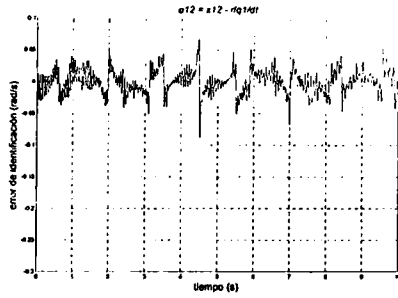
A.1.14. Error de identificación de  $q_2$ , con (4.7).



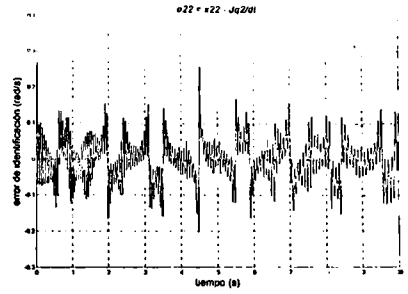
A.1.15. Error de identificación de  $q_1$ , con (4.8).



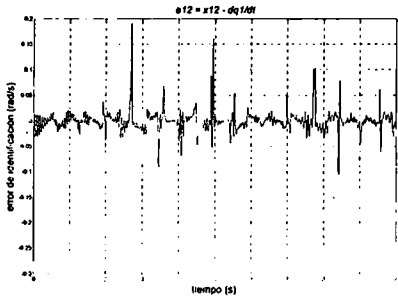
A.1.16. Error de identificación de  $q_2$ , con (4.8).



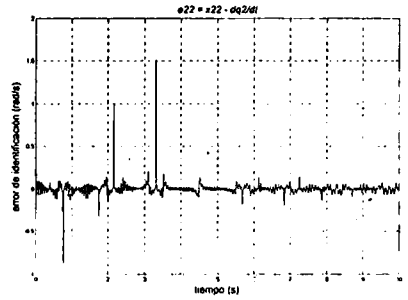
A.1.17. Error de identificación de  $dq_1/dt$ , (4.5).



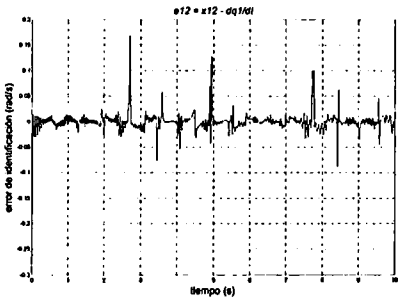
A.1.18. Error de identificación de  $dq_2/dt$ , (4.5).



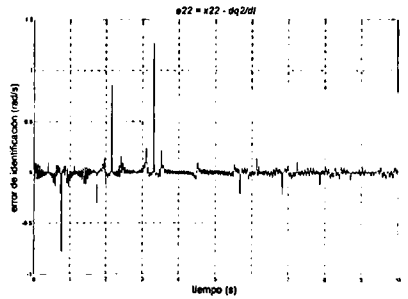
A.1.19. Error de identificación de  $dq_1/dt$ , (4.6).



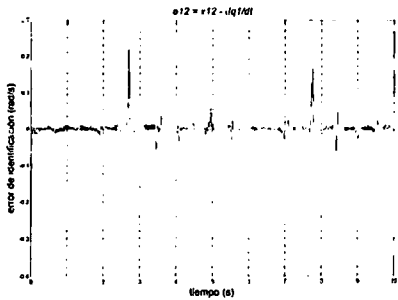
A.1.20. Error de identificación de  $dq_2/dt$ , (4.6).



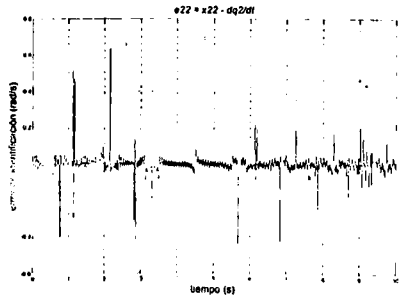
A.1.21. Error de identificación de  $dq_1/dt$ , (4.7).



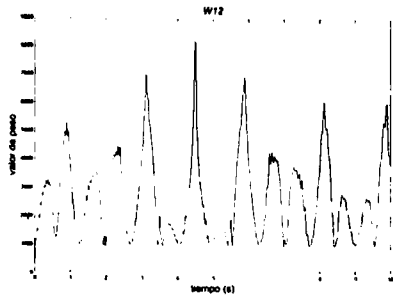
A.1.22. Error de identificación de  $dq_2/dt$ , (4.7).



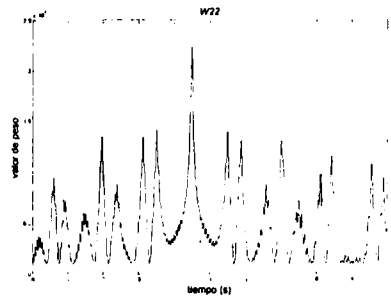
A.1.23. Error de identificación de  $dq_1/dt$ , (4.8).



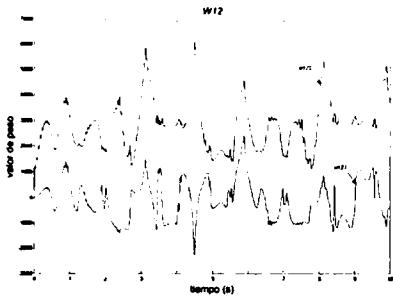
A.1.24. Error de identificación de  $dq_2/dt$ , (4.8).



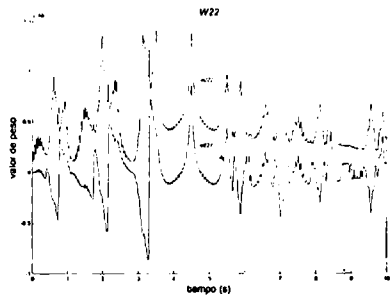
A.1.25. Pesos sinápticos de red neuronal 1, (4.5).



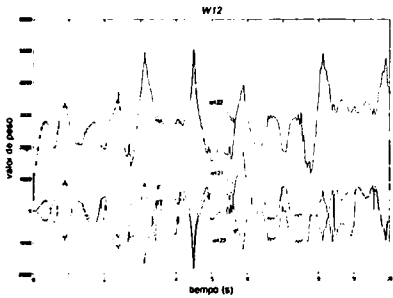
A.1.26. Pesos sinápticos de red neuronal 2, (4.5).



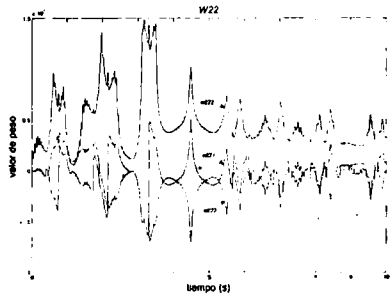
A.1.27. Pesos sinápticos de red neuronal 1, (4.6).



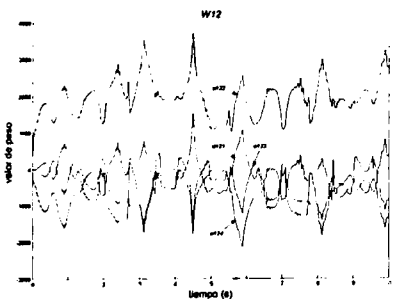
A.1.28. Pesos sinápticos de red neuronal 2, (4.6).



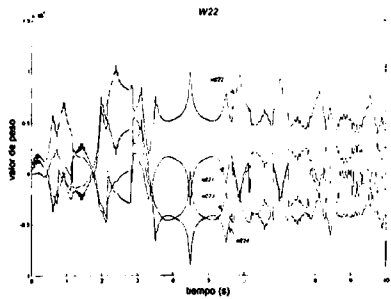
A.1.29. Pesos sinápticos de red neuronal 1, (4.7).



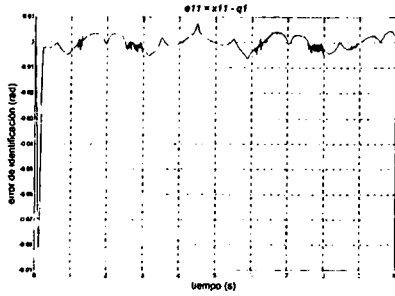
A.1.30. Pesos sinápticos de red neuronal 2, (4.7).



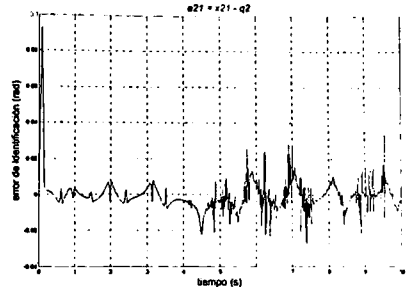
A.1.31. Pesos sinápticos de red neuronal 1, (4.8).



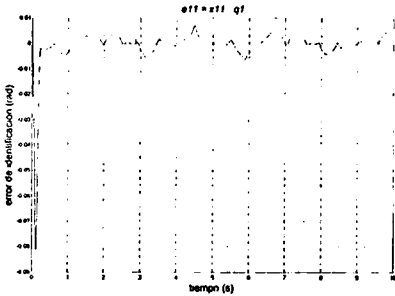
A.1.32. Pesos sinápticos de red neuronal 2, (4.8).



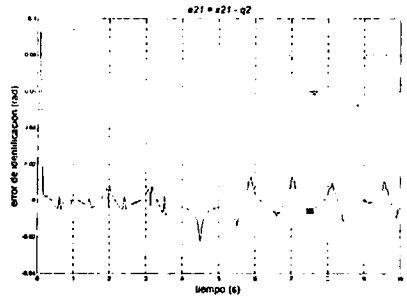
A.1.33. Error de identificación de  $q_1$ , con (4.9).



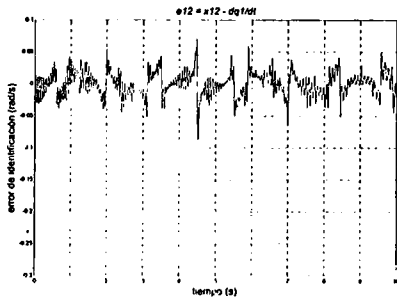
A.1.34. Error de identificación de  $q_2$ , con (4.9).



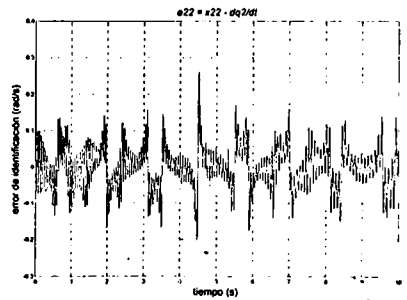
A.1.35. Error de identificación de  $q_1$ , con (4.10).



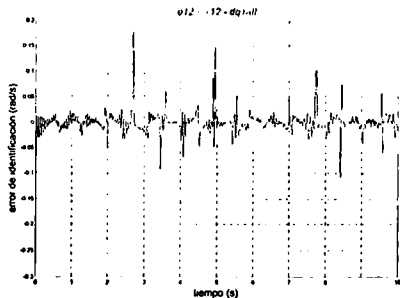
A.1.36. Error de identificación de  $q_2$ , con (4.10).



A.1.37. Error de identificación de  $dq_1/dt$ , (4.9).



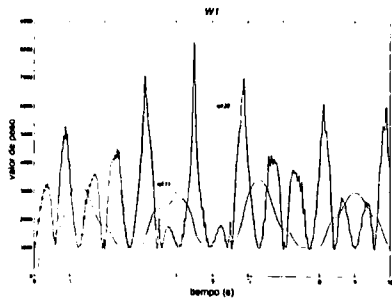
A.1.38. Error de identificación de  $dq_2/dt$ , (4.9).



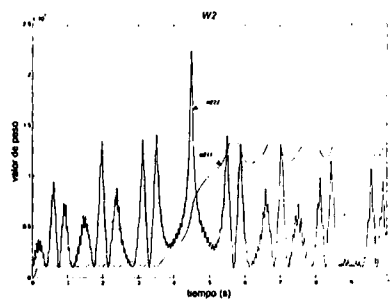
A.1.39. Error de identificación de  $dq_1/dt$ , (4.10).



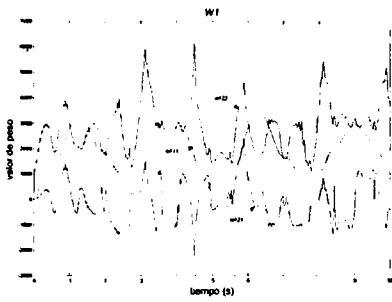
A.1.40. Error de identificación de  $dq_2/dt$ , (4.10).



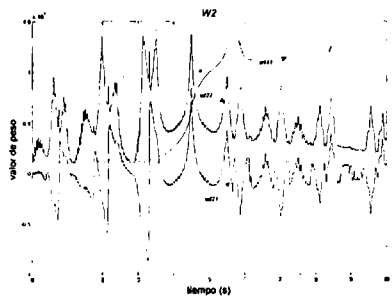
A.1.41. Pesos sinápticos de red neuronal 1, (4.9).



A.1.42. Pesos sinápticos de red neuronal 2, (4.9).



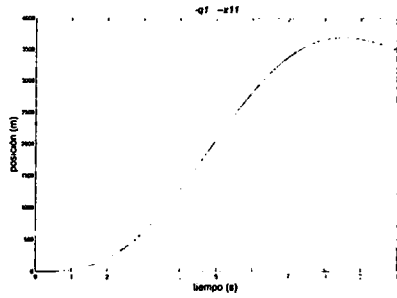
A.1.43. Pesos sinápticos de red neuronal 1, (4.10).



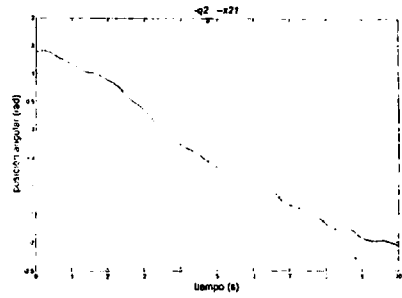
A.1.44. Pesos sinápticos de red neuronal 2, (4.10).



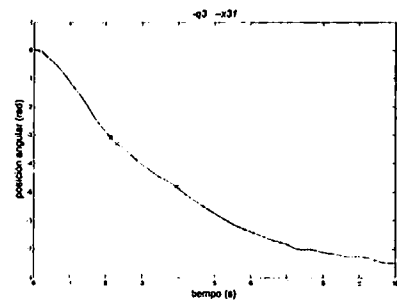
## A.2 Gráficas de identificación del manipulador robótico ANAT de siete uniones



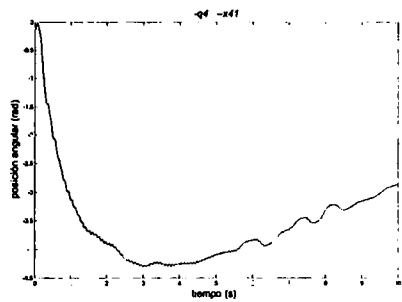
A.2.1. Identificación de  $q_1$ .



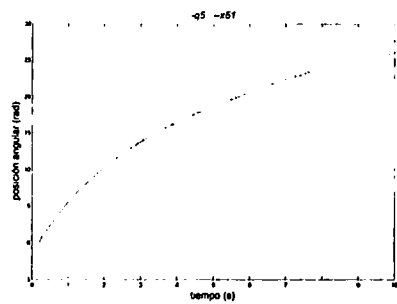
A.2.2. Identificación de  $q_2$ .



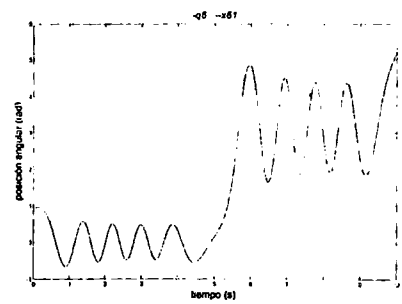
A.2.3. Identificación de  $q_3$ .



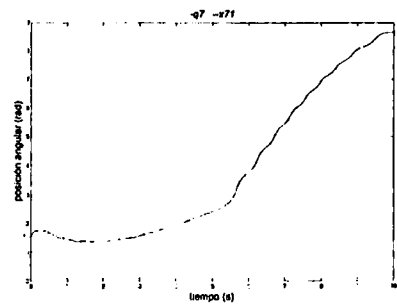
A.2.4. Identificación de  $q_4$ .



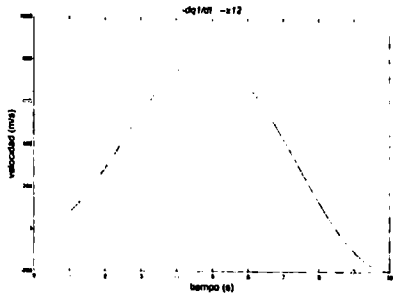
A.2.5. Identificación de  $q_5$ .



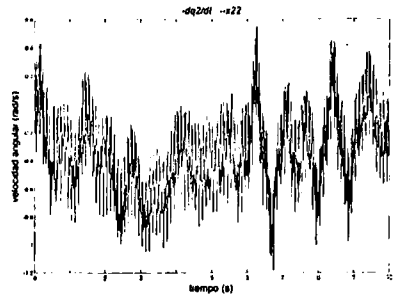
A.2.6. Identificación de  $q_6$ .



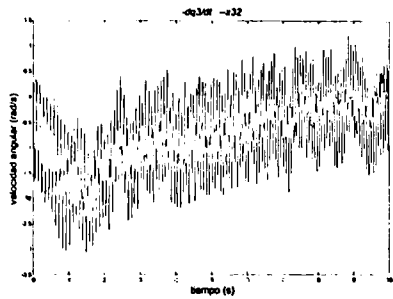
A.2.7. Identificación de  $q_7$ .



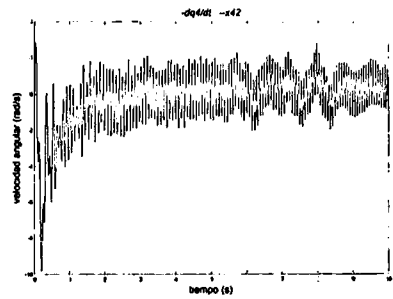
A.2.8. Identificación de  $dq_1/dt$ .



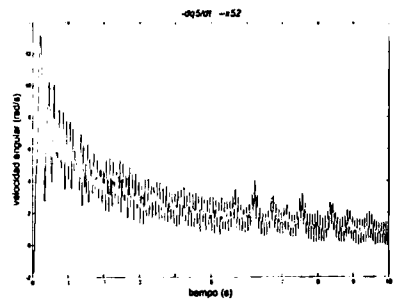
A.2.9. Identificación de  $dq_2/dt$ .



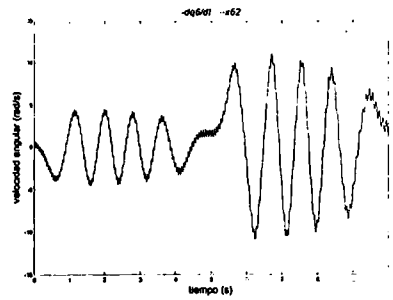
A.2.10. Identificación de  $dq_3/dt$ .



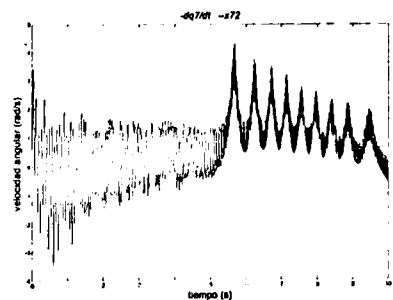
A.2.11. Identificación de  $dq_4/dt$ .



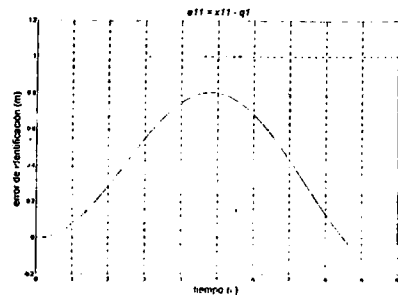
A.2.12. Identificación de  $dq_5/dt$ .



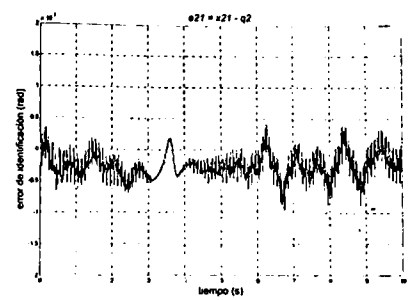
A.2.13. Identificación de  $dq_6/dt$ .



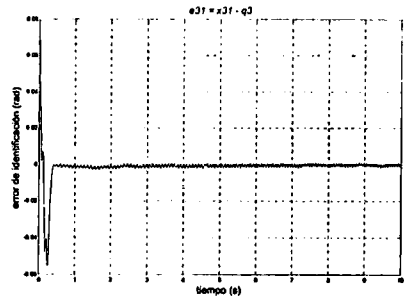
A.2.14. Identificación de  $dq_7/dt$ .



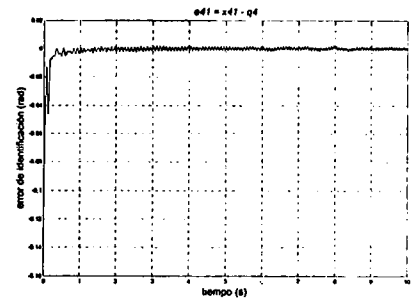
A.2.15. Error de identificación de  $q_1$ .



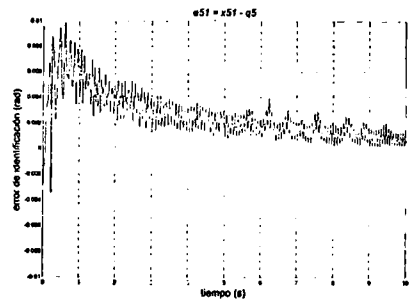
A.2.16. Error de identificación de  $q_2$ .



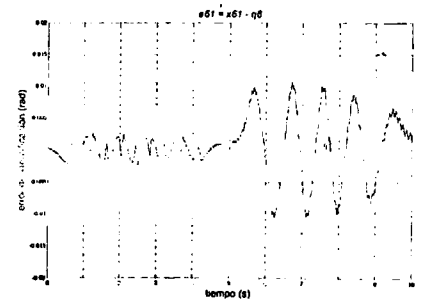
A.2.17. Error de identificación de  $q_3$ .



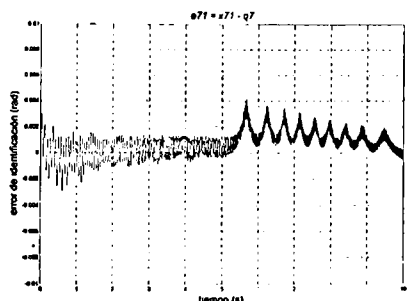
A.2.18. Error de identificación de  $q_4$ .



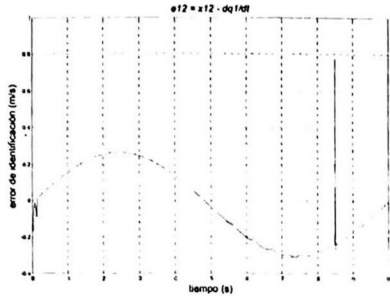
A.2.19. Error de identificación de  $q_5$ .



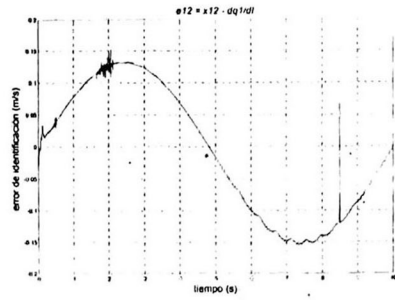
A.2.20. Error de identificación de  $q_6$ .



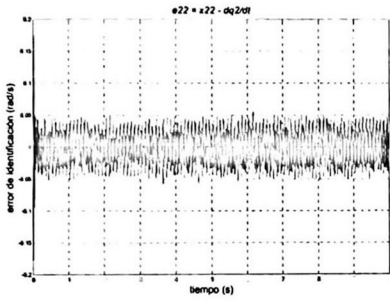
A.2.21. Error de identificación de  $q_7$ .



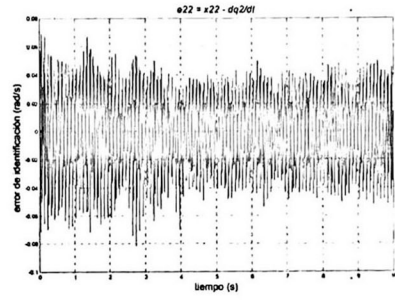
A.2.22. Error de identificación de  $dq_1/dt$ .



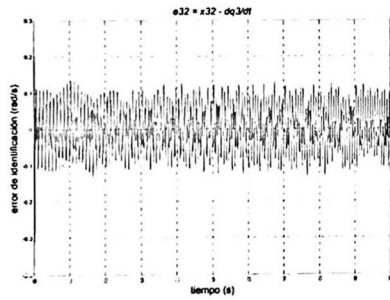
A.2.23. Error de identificación de  $dq_1/dt$ .



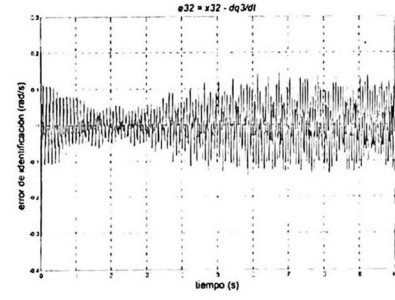
A.2.24. Error de identificación de  $dq_2/dt$ .



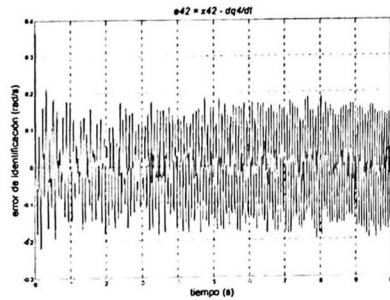
A.2.25. Error de identificación de  $dq_2/dt$ .



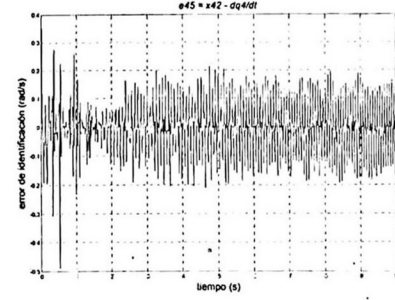
A.2.26. Error de identificación de  $dq_3/dt$ .



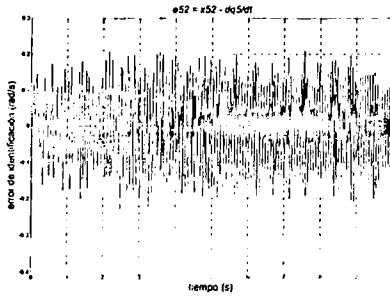
A.2.27. Error de identificación de  $dq_3/dt$ .



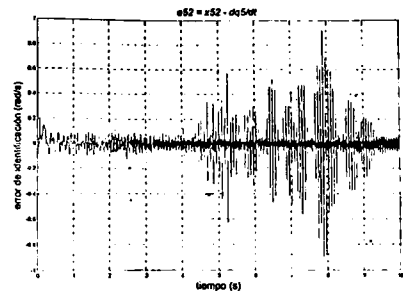
A.2.28. Error de identificación de  $dq_4/dt$ .



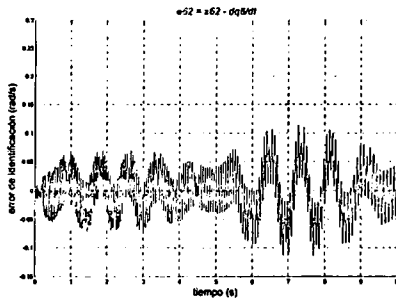
A.2.29. Error de identificación de  $dq_4/dt$ .



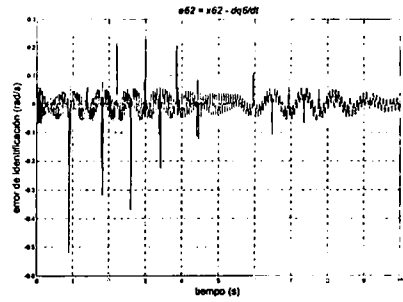
A.2.30. Error de identificación de  $dq_s/dt$ .



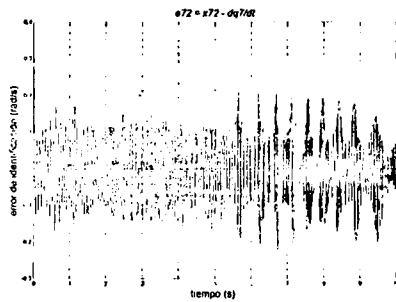
A.2.31. Error de identificación de  $dq_s/dt$ .



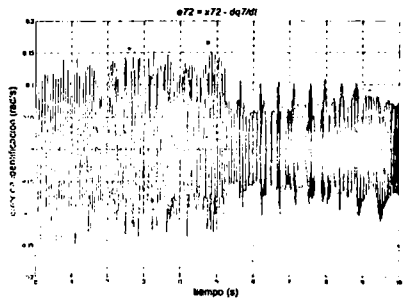
A.2.32. Error de identificación de  $dq_s/dt$ .



A.2.33. Error de identificación de  $dq_s/dt$ .

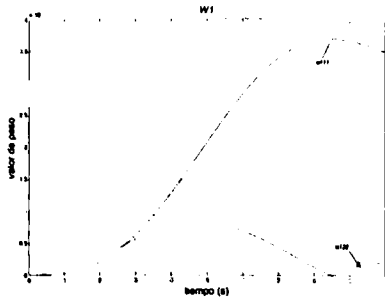


A.2.34. Error de identificación de  $dq_7/dt$ .

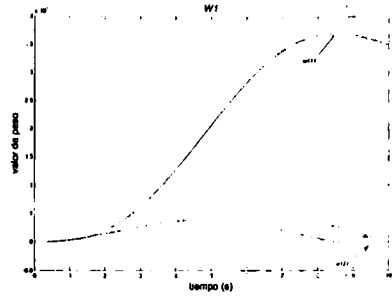


A.2.35. Error de identificación de  $dq_7/dt$ .

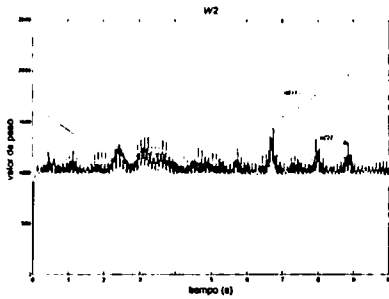




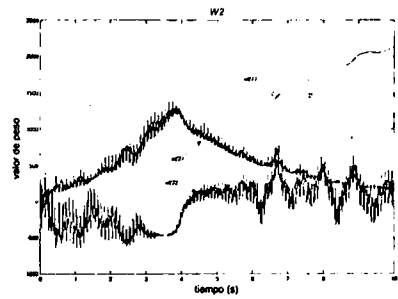
A.2.36. Evolución de pesos red neuronal 1.



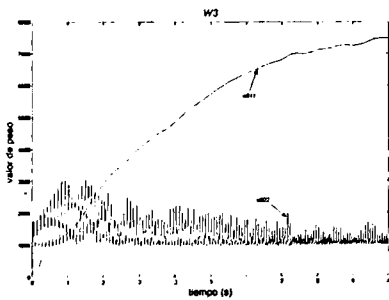
A.2.42. Evolución de pesos red neuronal 1.



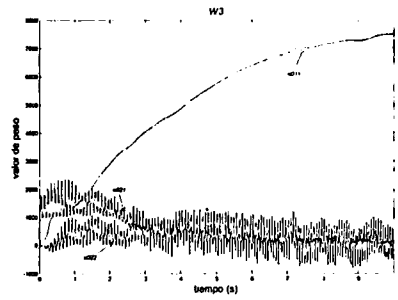
A.2.37. Evolución de pesos red neuronal 2.



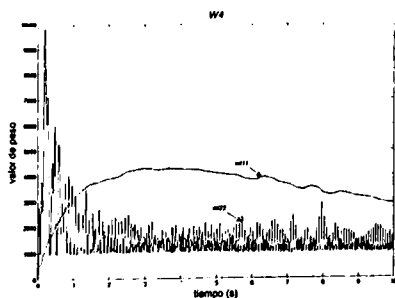
A.2.44. Evolución de pesos red neuronal 2.



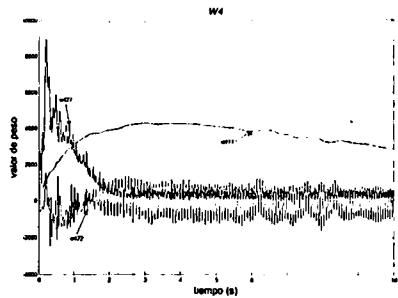
A.2.38. Evolución de pesos red neuronal 3.



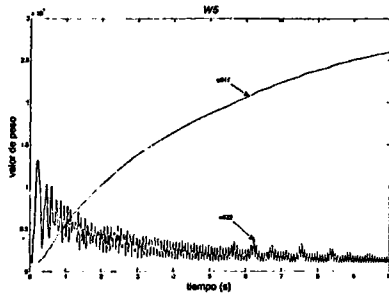
A.2.39. Evolución de pesos red neuronal 3.



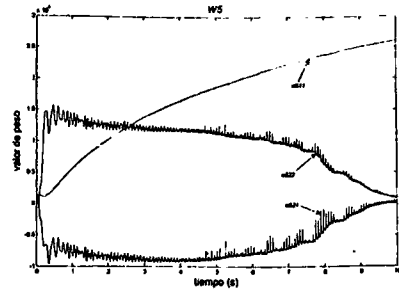
A.2.40. Evolución de pesos red neuronal 4.



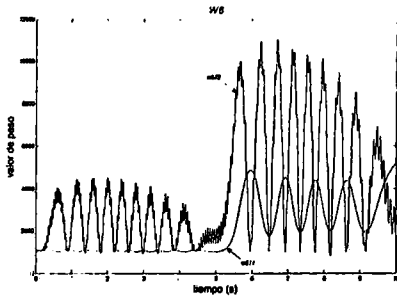
A.2.41. Evolución de pesos red neuronal 4.



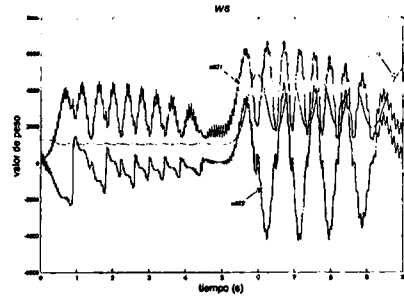
A.2.43. Evolución de pesos red neuronal 5.



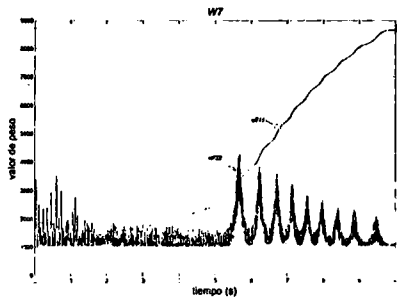
A.2.48. Evolución de pesos red neuronal 5.



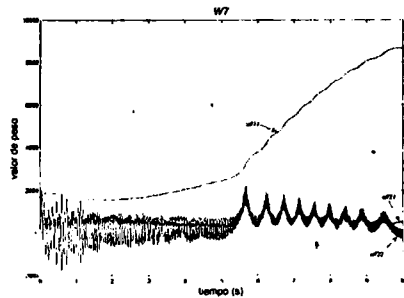
A.2.45. Evolución de pesos red neuronal 6.



A.2.47. Evolución de pesos red neuronal 6.

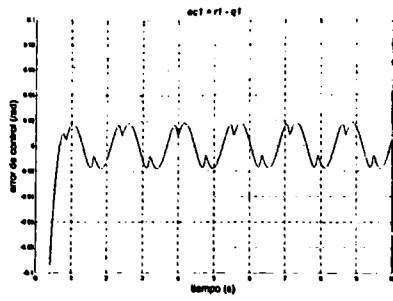


A.2.46. Evolución de pesos red neuronal 7.

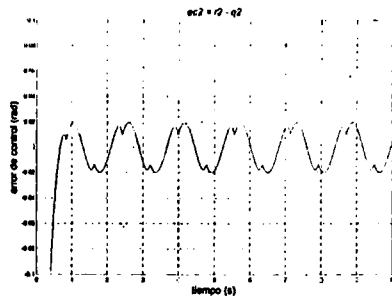


A.2.49. Evolución de pesos red neuronal 7.

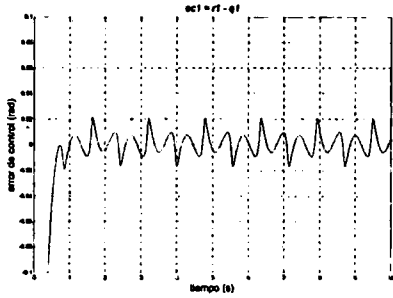
### A.3 Gráficas de control del manipulador robótico de dos uniones



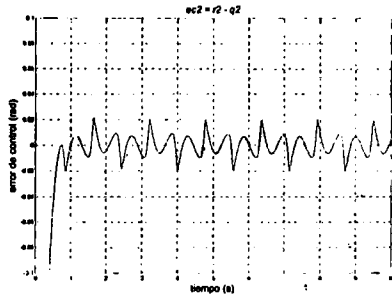
A.3.1. Error de control de  $q_1$ , con (4.5).



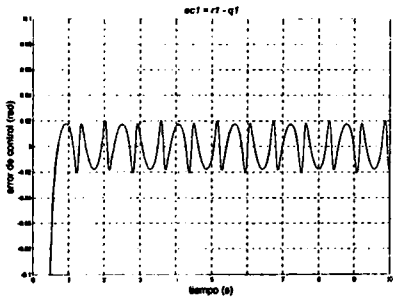
A.3.2. Error de control de  $q_2$ , con (4.5).



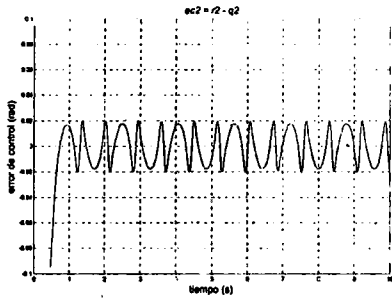
A.3.3. Error de control de  $q_1$ , con (4.6).



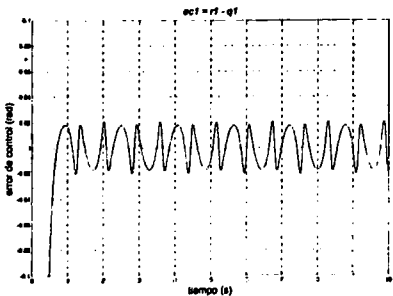
A.3.4. Error de control de  $q_2$ , con (4.6).



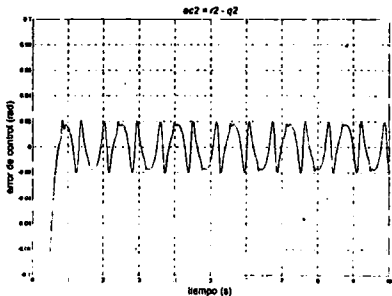
A.3.5. Error de control de  $q_1$ , con (4.9).



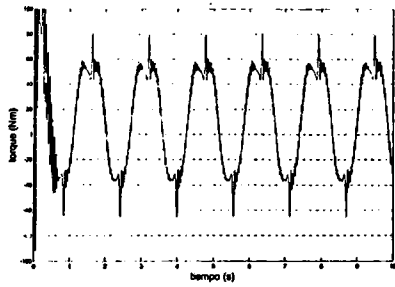
A.3.6. Error de control de  $q_2$ , con (4.9).



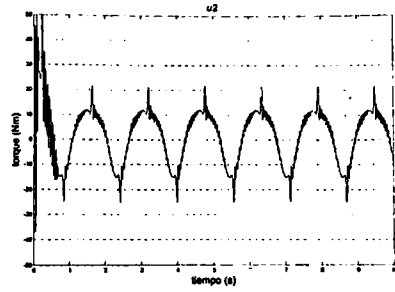
A.3.7. Error de control de  $q_1$ , con (4.10).



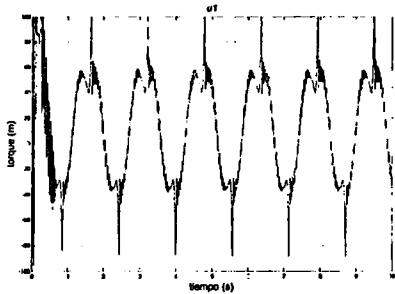
A.3.8. Error de control de  $q_2$ , con (4.10).



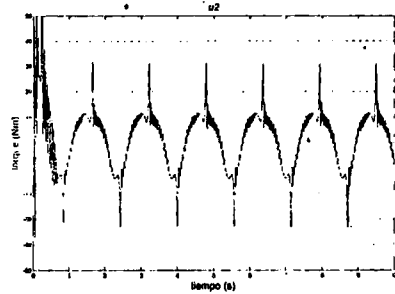
A.3.9. Control  $u_1$ , con (4.5).



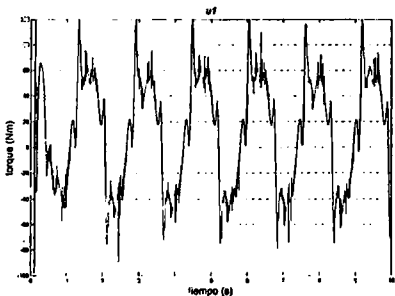
A.3.10. Control  $u_2$ , con (4.5).



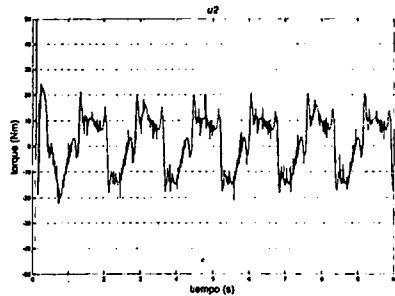
A.3.11. Control  $u_1$ , con (4.6).



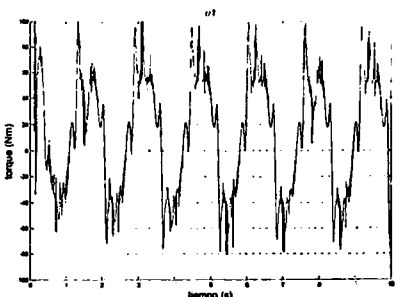
A.3.12. Control  $u_2$ , con (4.6).



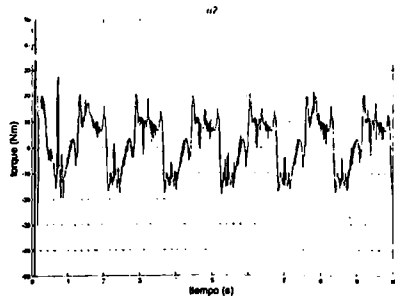
A.3.13. Control  $u_1$ , con (4.9).



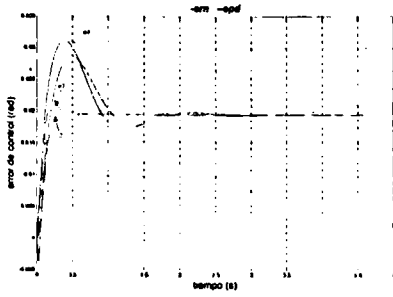
A.3.14. Control  $u_2$ , con (4.9).



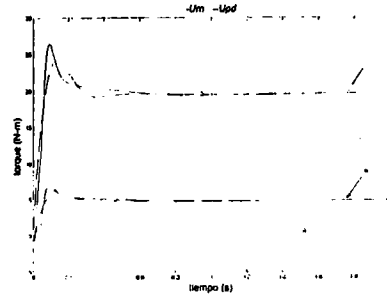
A.3.15. Control  $u_1$ , con (4.10).



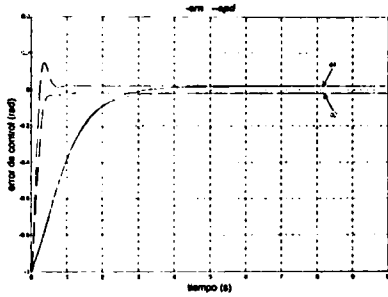
A.3.16. Control  $u_2$ , con (4.10).



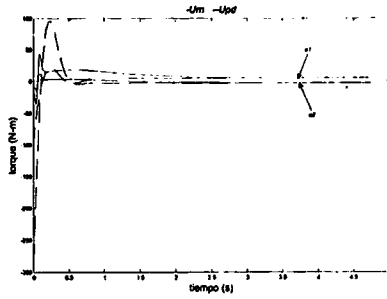
A.3.17. Error de control (RN vs. PD).



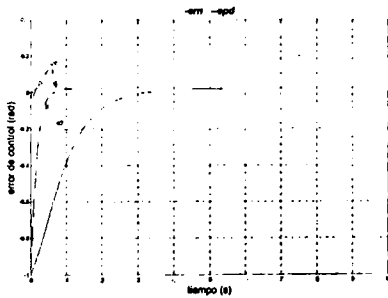
A.3.18. Control (RN vs. PD).



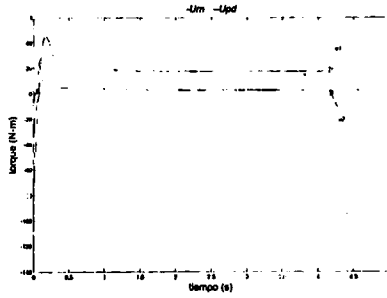
A.3.19. Error de control (RN vs. PD).



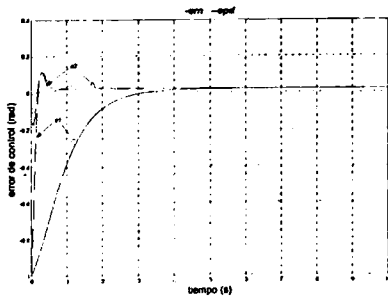
A.3.20. Control (RN vs. PD).



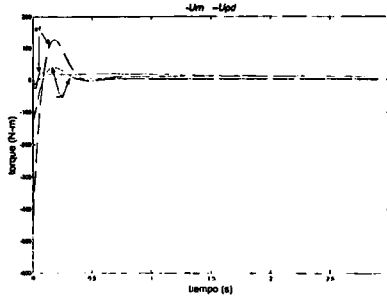
A.3.21. Error de control (RN vs. PD).



A.3.22. Control (RN vs. PD).

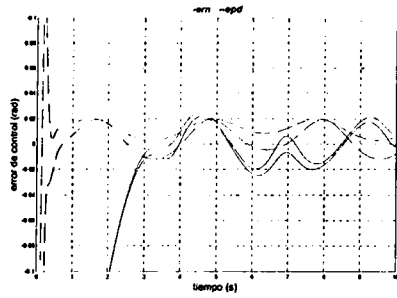


A.3.23. Error de control (RN vs. PD).

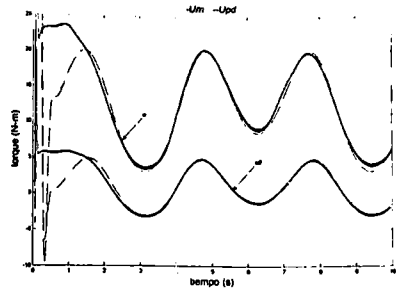


A.3.24. Control (RN vs. PD).

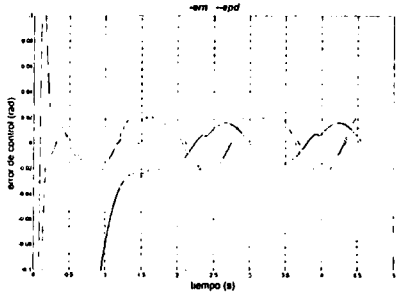




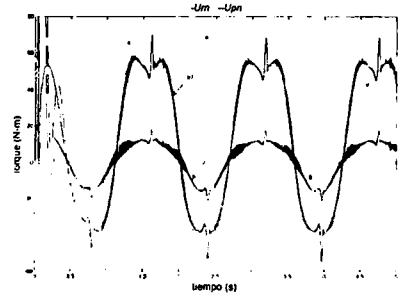
A.3.25. Error de control (RN vs. PD).



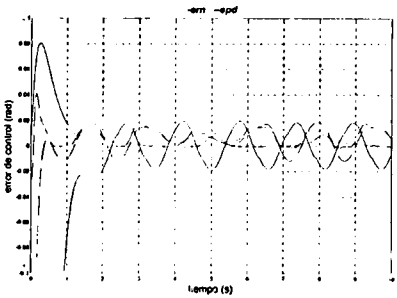
A.3.26. Control (RN vs. PD).



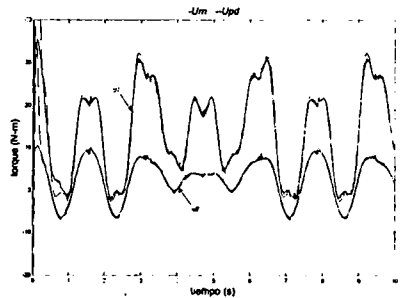
A.3.27. Error de control (RN vs. PD).



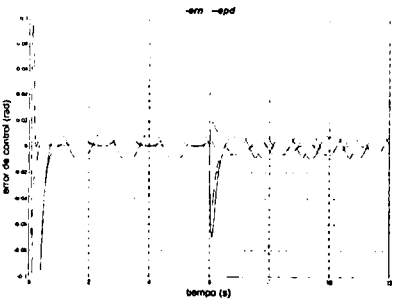
A.3.28. Control (RN vs. PD).



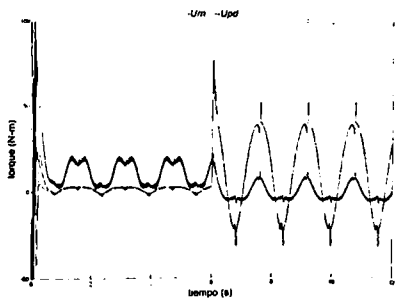
A.3.29. Error de control (RN vs. PD).



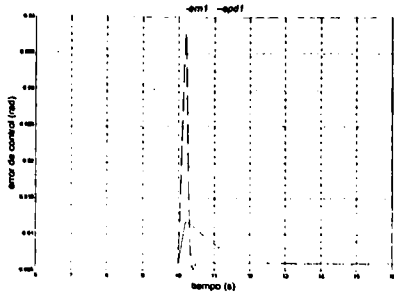
A.3.30. Control (RN vs. PD).



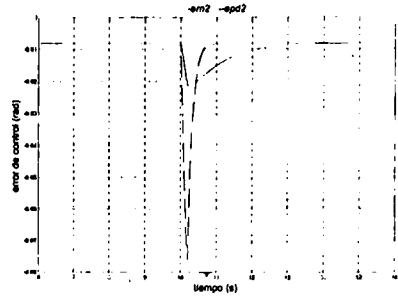
A.3.31. Error de control (RN vs. PD).



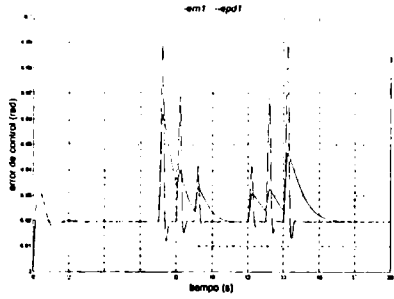
A.3.32. Control (RN vs. PD).



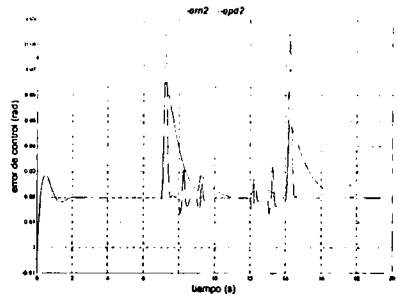
A.3.33. Error de control de  $q_1$  (RN vs. PD).



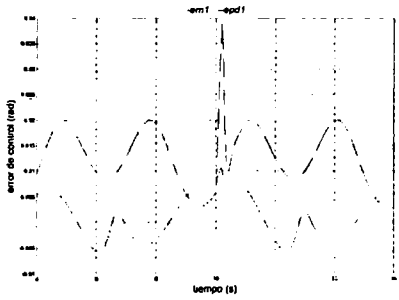
A.3.34. Error de control de  $q_2$  (RN vs. PD).



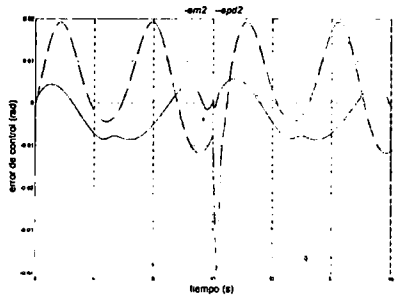
A.3.35. Error de control de  $q_1$  (RN vs. PD).



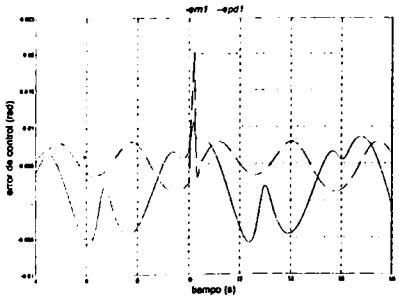
A.3.36. Error de control de  $q_2$  (RN vs. PD).



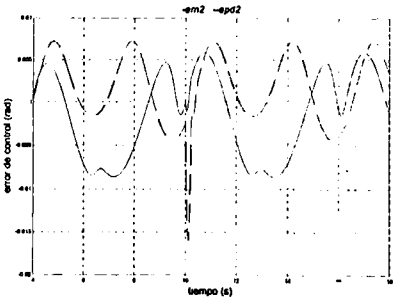
A.3.37. Error de control de  $q_1$  (RN vs. PD).



A.3.38. Error de control de  $q_2$  (RN vs. PD).

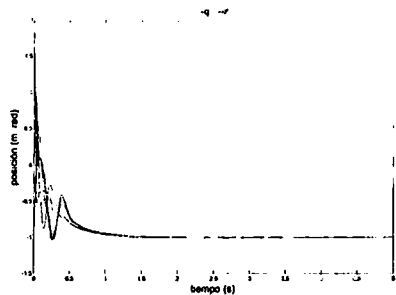


A.3.39. Error de control de  $q_1$  (RN vs. PD).

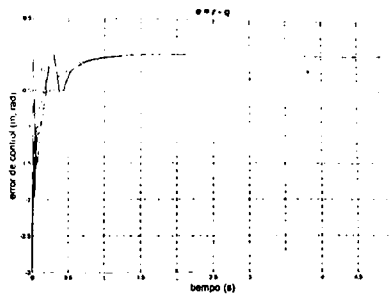


A.3.40. Error de control de  $q_2$  (RN vs. PD).

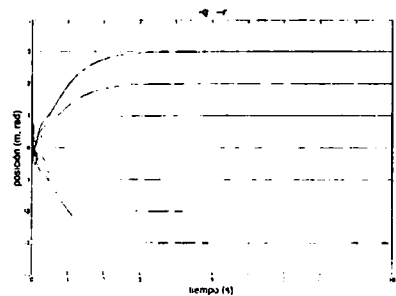
## A.4 Gráficas de control del manipulador robótico ANAT de siete uniones



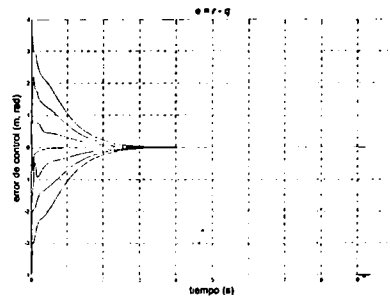
A.4.1. Regulación.



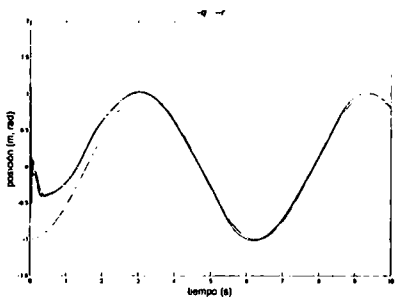
A.4.2. Error de regulación.



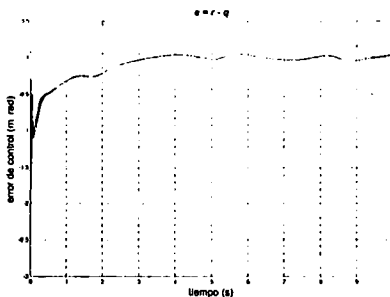
A.4.3. Regulación.



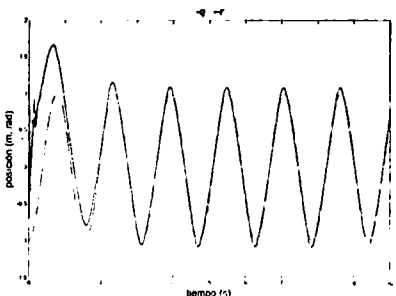
A.4.4. Error de regulación.



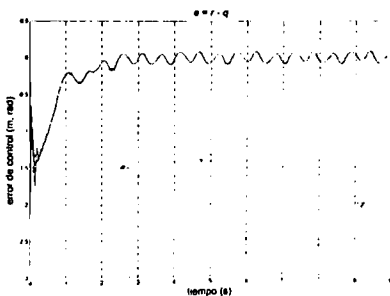
A.4.5. Seguimiento de trayectoria.



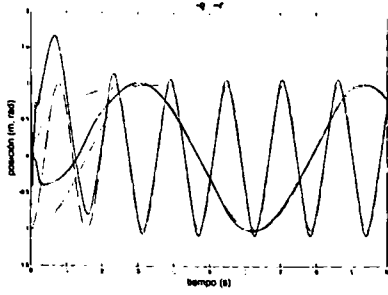
A.4.6. Error de seguimiento.



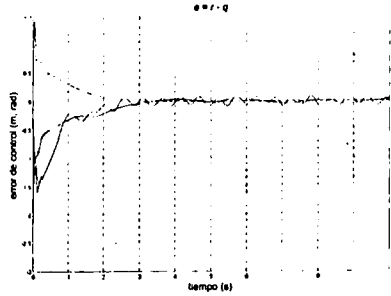
A.4.7. Seguimiento de trayectoria.



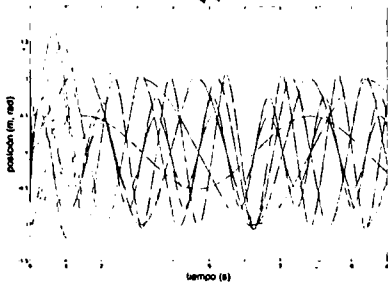
A.4.8. Error de seguimiento.



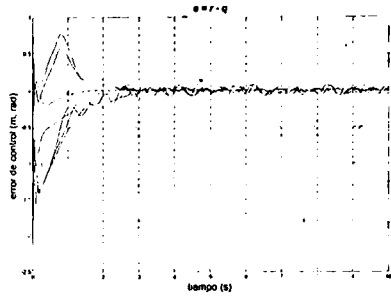
A.4.9. Seguimiento de trayectoria.



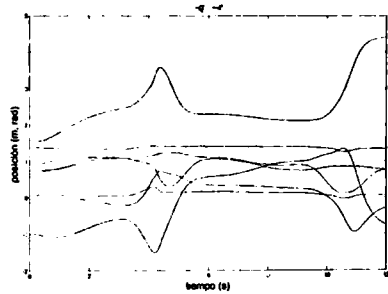
A.4.10. Error de seguimiento.



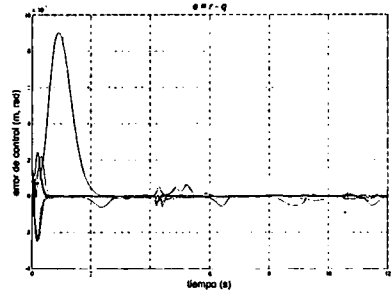
A.4.11. Seguimiento de trayectoria.



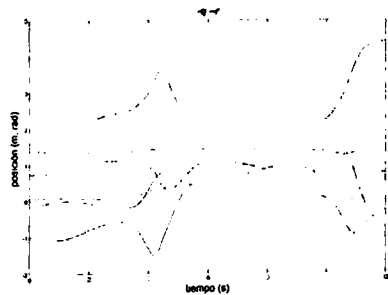
A.4.12. Error de seguimiento.



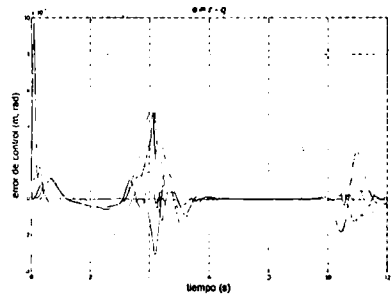
A.4.13. Seguimiento con CND.



A.4.14. Error de seguimiento, con CND.



A.4.15. Seguimiento con CCA.



A.4.16. Error de seguimiento, con CCA.

## **APÉNDICE B.**

**Artículo presentado en “2006 IEEE International Symposium  
Intelligent Control”**



# Decentralized Neural Identification and Control for Robotics Manipulators

Edgar N. Sanchez\*, Armando Gaytan\*, and Maarouf Saad\*\*

\*CINVESTAV, Unidad Guadalajara,  
Apartado Postal 31-438, Plaza La Luna,  
Guadalajara, Jalisco C.P. 45091, México,  
email: [sanchez@gdl.cinvestav.mx](mailto:sanchez@gdl.cinvestav.mx).

\*\*École de Technologie Supérieure,  
1100 Notre Dame West,  
Montreal, Quebec, Canada, H3C 1K3

**Abstract** — This paper presents a decentralized control scheme, based on a recurrent neural identifier with a block control structure, and its application to robotics manipulators. A local joint controller is proposed for each joint, using only local angular position and velocity measurements. These very simple local joint controllers allow trajectory tracking, with reduced computations. The applicability of the proposed scheme is illustrated, via simulations, first by the applications to a two degree of freedom robotic manipulator and then to a seven degree of freedom one.

## I. INTRODUCTION

Robotics manipulator control is still a very interesting research subject. The preferred structure is the centralized one, which has become very efficient thanks to technological advancements [1]. However, due to manipulator dynamics, which is highly nonlinear, the strong interconnection between the components, and other effects, the corresponding mathematical model is quite complex, requiring very elaborated centralized control schemes [1]-[3], [15].

Since early 60's of the 20th century, the decentralized control approach, an important component of control system methodologies, has been developed. In this approach, the whole system is seen as the interconnection of multiple local sub-systems; the control synthesis is done for each local sub-system, considering only the local variables [4], [5]. This approach was created in order to solve the control of large-scale systems; for these systems, a centralized controller exchanges a large quantity of information with the local sub-systems, easily overloading the computation capabilities existing today [4]-[6].

Decentralized control has been applied in robotics, mainly to cooperative multiple mobile robots and cooperative

manipulators, where it is natural to consider each mobile robot or each manipulator as a sub-system of the whole system. However, recently, research is being developed related to decentralized stand alone robotics manipulators; in this approach, each joint is considered as a sub-system in order to develop local controllers, which consider only local angular position and angular velocities measurements, and compensate the interconnection effects, usually assumed as disturbances. The resulting local controllers are easily implemented for real-time applications. [3], [7], [8].

In this paper, the authors propose a new decentralized control structure of stand alone robotics manipulators, on the basis of recurrent neural networks. This structure is motivated by the well known nonlinear approximation capabilities of such neural networks [9], [10]. The proposed scheme is composed by a decentralized recurrent neural identifier, and a decentralized controller, developed based on the block control methodology. The applicability of the proposed scheme is illustrated, via simulation, by its application to the identification and control of a two degree of freedom robotic manipulator, and of an Articulated Nimble Adaptable Trunk (ANAT) one, with seven degree of freedom.

## II. NEURAL IDENTIFIER

### A. Mathematical Model

The robotics manipulator dynamics are modeled as follows [1], [2], [7], [11]:

$$\ddot{q} = M(q, \dot{q})^{-1} [\tau - V(q, \dot{q}) - F_f(\dot{q}) - F_g(q)] \quad (1)$$

where  $q, \dot{q}, \ddot{q} \in \mathcal{R}^N$  are, respectively, angular position, angular speed, and angular acceleration of the  $N$  joints,

$M(\cdot) \in \mathfrak{R}^{N \times N}$  is the inertia matrix,  $\tau \in \mathfrak{R}^N$  is the applied torque to the  $N$  joints, and  $V(\cdot)$ ,  $F_f(\cdot)$ ,  $F_g(\cdot) \in \mathfrak{R}^N$  are the terms corresponding to centrifugal and Coriolis effects, friction and gravity, respectively.

This model does not completely represent the manipulator due to the inexact knowledge of the friction effects, the omission of some dynamics, and parameter uncertainties and variations [2], [3], [15].

(1) can be partitioned in  $N$  sub-systems, each one formulated as:

$$\ddot{q}_i = f_i(q_i, \dot{q}_i, \tau_i) + \Gamma_i(q_1, q_2, \dots, q_N, \dot{q}_1, \dot{q}_2, \dots, \dot{q}_N, \tau_1, \tau_2, \dots, \tau_N) \quad (2)$$

where  $i=1,2,\dots,N$ ,  $f_i(\cdot)$  depends only on the local variables, and  $\Gamma_i(\cdot)$  represents the interconnection effects.

For (1) and (2), it is assumed that [2], [7]:

$$\begin{aligned} M(q)\ddot{q} &\leq c_m \|\ddot{q}\| \\ V(q, \dot{q}) &\leq c_v \|\dot{q}\|^2 \\ F_f(\dot{q}) &\leq c_f \|\dot{q}\| \\ F_g(q) &\leq c_g \end{aligned} \quad (3)$$

where  $c_m$ ,  $c_v$ ,  $c_f$  and  $c_g$  are constants.

### B. Decentralized Neural Identifier

In order to identify (2), we propose the following recurrent neural network model

$$\begin{aligned} \dot{x}_{i1} &= -a_{i1}x_{i1} + w_{i11}\rho_{i11}(q_i) + x_{i2} \\ \dot{x}_{i2} &= -a_{i2}x_{i2} + w_{i21}\rho_{i21}(q_i) + w_{i22}\rho_{i22}(\dot{q}_i) + \tau_i \end{aligned} \quad (4)$$

where  $i=1,2,\dots,N$ ,  $x_{i1}$  and  $x_{i2}$  are the states (neurons) of the  $i$ -th neural network, representing  $q_i$  and  $\dot{q}_i$  (the  $i$ -th angular position and velocity);  $a_{i1}$ ,  $a_{i2}$  are positive real constants,  $w_{ij}$  is the respective synaptic weight of the  $\rho_{ijk}(\cdot)$  activation function, and  $\tau_i$  is the torque applied to the  $i$ -th joint.

It is worth to note that (3), as defined, constitutes a series-parallel identifier and fulfills the conditions of the block controllable form (BCF) [12], [13], with two blocks of dimension one. This is also a particular case of the more general one discussed in [6].

### C. Neural Network Learning Law

In order to on-line update the synaptic weights, we follow [10]. Defining the identification error, between the neural identifier and the joint variables for each  $i$ -th joint, by:

$$\begin{aligned} e_{i1} &= x_{i1} - q_i \\ e_{i2} &= x_{i2} - \dot{q}_i \end{aligned} \quad (5)$$

Then the learning law is determined as

$$\dot{w}_{ijk} = -\gamma_{ijk} e_{ij} \rho_{ijk} \quad (6)$$

where  $k \in \{1,2\}$ , each  $\gamma_{ijk}$  is positive real constant. For each weight, we obtain:

$$\begin{aligned} \dot{w}_{i11} &= -\gamma_{i11} e_{i1} \rho_{i11}(q_i) \\ \dot{w}_{i21} &= -\gamma_{i21} e_{i2} \rho_{i21}(q_i) \\ \dot{w}_{i22} &= -\gamma_{i22} e_{i2} \rho_{i22}(\dot{q}_i) \end{aligned} \quad (7)$$

with the activation functions defined as:

$$\begin{aligned} \rho_{i11} &= \tanh(q_i) \\ \rho_{i21} &= \tanh(q_i) \\ \rho_{i22} &= \tanh(\dot{q}_i) \end{aligned} \quad (8)$$

(4) represents the local model without considering interconnection effects. In [9], [10], it is explained how to modify (6) and (7) to do them robust in presence of these disturbances and errors.

## III. NEURAL CONTROL

(4) is a very simple case of BCF. Then, it is possible to synthesize a control structure by a recursive method [12], [13]. First, let define for each  $i$ -th joint, the desired trajectory  $\beta_{i1}$ ,  $\beta_{i2}$  respectively for each local variable  $q_i$ ,  $\dot{q}_i$ . Then for each local variable, we define the tracking error as  $q_i - \beta_{i1}$ ,  $\dot{q}_i - \beta_{i2}$ .

It is easy to see that the following inequalities are satisfied:

$$\begin{aligned} |q_i - \beta_{i1}| &\leq |q_i - x_{i1}| + |x_{i1} - \beta_{i1}| \\ |\dot{q}_i - \beta_{i2}| &\leq |\dot{q}_i - x_{i2}| + |x_{i2} - \beta_{i2}| \\ i &= 1 \dots N \end{aligned}$$

where  $| \cdot |$  stands for the absolute value.

Given that  $|q_i - x_{i1}| = |x_{i1} - q_i|$  and  $|\dot{q}_i - x_{i2}| = |x_{i2} - \dot{q}_i|$ , then the first right hand term of these inequalities is minimized by the neural identifier already discussed. Hence, the control goal is to minimize the second right hand term. It is clear that the simultaneous minimization of both right hand terms ensures the minimization of the robotics manipulator trajectory tracking error.

In order to minimize the tracking error between the neuron states and the desired trajectories (second right hand term), we introduce the following coordinate transformation.

$$\begin{aligned} z_{i1} &= x_{i1} - \beta_{i1} \\ \dot{z}_{i1} &= \dot{x}_{i1} - \dot{\beta}_{i1} \\ \dot{z}_{i1} &= -k_{i1}z_{i1} \end{aligned} \quad (9)$$

$$\begin{aligned} z_{i2} &= x_{i2} - \beta_{i2} \\ \dot{z}_{i2} &= \dot{x}_{i2} - \dot{\beta}_{i2} \\ \dot{z}_{i2} &= -k_{i2}z_{i2} \end{aligned} \quad (10)$$

Substituting the first equation of (4) in the second one of (9) and equalizing the result with the third one of (9), we obtain

$$\begin{aligned} \dot{z}_{i1} &= -a_{i1}x_{i1} + w_{i11}\rho_{i11}(q_i) + x_{i2} - \dot{\beta}_{i1} \\ -k_{i1}z_{i1} &= -a_{i1}x_{i1} + w_{i11}\rho_{i11}(q_i) + x_{i2} - \dot{\beta}_{i1} \\ -k_{i1}z_{i1} + a_{i1}x_{i1} - w_{i11}\rho_{i11}(q_i) + \dot{\beta}_{i1} &= x_{i2} \\ \beta_{i2} &= -k_{i1}z_{i1} + a_{i1}x_{i1} - w_{i11}\rho_{i11}(q_i) + \dot{\beta}_{i1} \end{aligned} \quad (11)$$

Following a similar procedure, we substitute the second equation of (4) in the second one of (10) and equalize the result with the third one of (10), in order to obtain the formula for  $\tau_i$ :

$$\begin{aligned} \dot{z}_{i2} &= -a_{i2}x_{i2} + w_{i21}\rho_{i21}(q_i) + w_{i22}\rho_{i22}(\dot{q}_i) + \tau_i - \dot{\beta}_{i2} \\ -k_{i2}z_{i2} &= -a_{i2}x_{i2} + w_{i21}\rho_{i21}(q_i) + w_{i22}\rho_{i22}(\dot{q}_i) + \tau_i - \dot{\beta}_{i2} \\ \tau_i &= -k_{i2}z_{i2} + a_{i2}x_{i2} - w_{i21}\rho_{i21}(q_i) - w_{i22}\rho_{i22}(\dot{q}_i) + \dot{\beta}_{i2} \end{aligned} \quad (12)$$

with

$$\dot{\beta}_{i2} = -k_{i1}\dot{z}_{i1} + a_{i1}\dot{x}_{i1} - \dot{w}_{i11}\rho_{i11}(q_i) - w_{i11}\dot{\rho}_{i11}(q_i) + \ddot{\beta}_{i1} \quad (13)$$

and

$$\dot{\rho}_{i11}(q_i) = \sec^2(q_i)\dot{q}_i \quad (14)$$

Hence, by means of (12), (13) and (14), it is possible to implement the local controllers for each joint.

#### IV. SIMULATIONS

In order to illustrate the applicability of the proposed identification and control procedure, we test it, via simulation using MATLAB (a trade mark of the MathWorks Inc.). The whole scheme is applied to a two degree of freedom manipulator and the ANAT manipulator. In order to simulate the 2 DOF manipulator, we use the following mathematical model [11]:

$$\begin{aligned} \tau_1 &= ((2m_1 + 8m_2 + 6m_2 \cos(q_2))\ddot{q}_1 + (2m_2 + 3m_2 \cos(q_2))\ddot{q}_2)^2 / 6 \\ &\quad - (m_2 l^2 / 2) \sin(q_2) (\dot{q}_2^2 + 2\dot{q}_1 \dot{q}_2) \\ &\quad + (gl / 2) (m_1 \cos(q_1) + m_2 \cos(q_1 + q_2) + 2m_2 \cos(q_1)) \\ \tau_2 &= ((2m_2 + 3m_2 \cos(q_2))\ddot{q}_1 + (2m_2)\ddot{q}_2)^2 / 6 \\ &\quad + (m_2 l^2 / 2) \sin(q_2) \dot{q}_1^2 + (gl / 2) m_2 \cos(q_1 + q_2) \end{aligned} \quad (15)$$

The parameter values are:  $l=0.5\text{m}$  (link length),  $m_1=0.75\text{kg}$ ,  $m_2=0.75\text{kg}$  (link mass) and  $g=9.806\text{m/s}^2$ . For ANAT, we use the simulation program mentioned in [14].

For identification, we use adequate chirp signals. Fig. 1 to Fig. 6 present the identification results for the 2 DOF manipulator. The corresponding results for the ANAT manipulator identification are presented in Fig. 7 and Fig. 8.

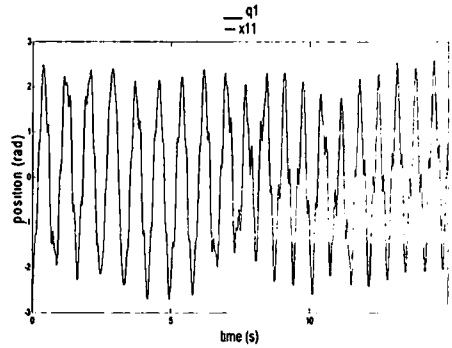


Fig. 1.  $q_1$  identification for the 2 DOF manipulator.

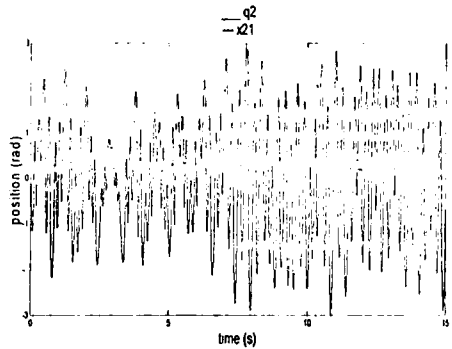


Fig. 2.  $q_2$  identification for the 2 DOF manipulator.

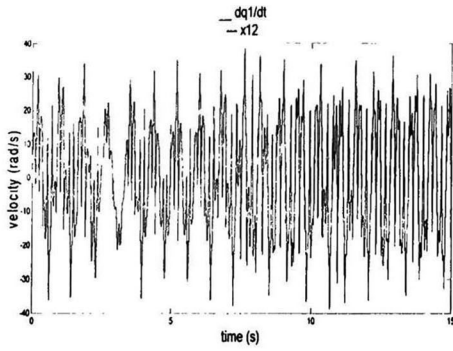


Fig. 3.  $\dot{q}_1$  identification for the 2 DOF manipulator.

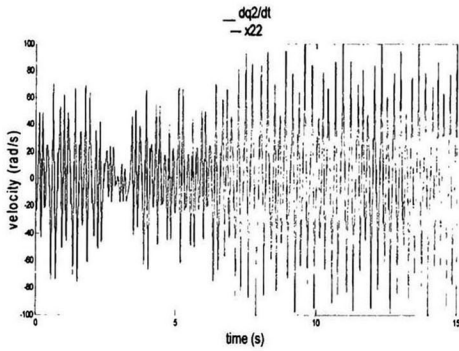


Fig. 4.  $\dot{q}_2$  identification for the 2 DOF manipulator.

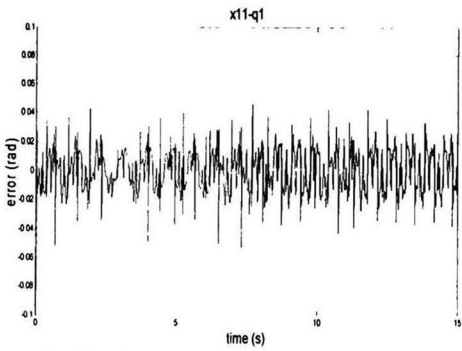


Fig. 5.  $q_1$  identification error for the 2 DOF manipulator.

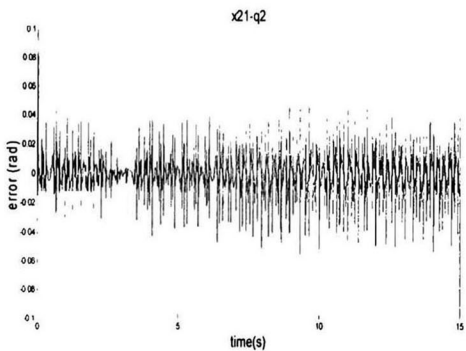


Fig. 6.  $q_2$  identification error for the 2 DOF manipulator.

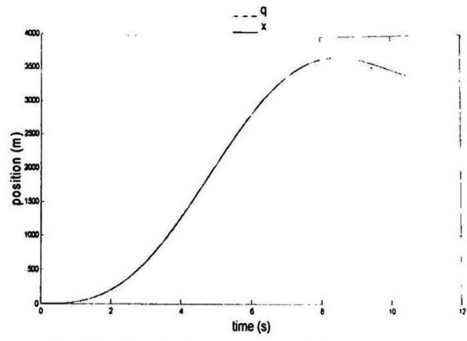


Fig. 7.  $q_1$  identification for the ANAT manipulator.

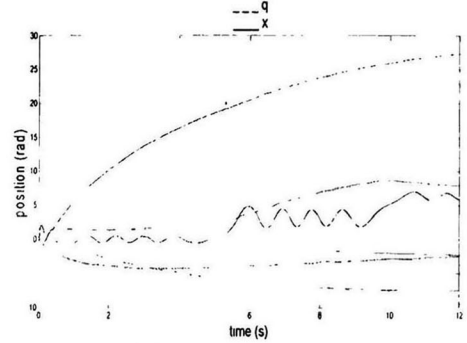


Fig. 8.  $q_2, \dots, q_7$  identification for the ANAT manipulator.

The obtained results are very satisfactory for both manipulators. Once, the identification component is validated, we proceed to test the whole structure (Identification and trajectory tracking control) for both manipulators. The corresponding results are displayed in Fig. 9 to Fig. 20.

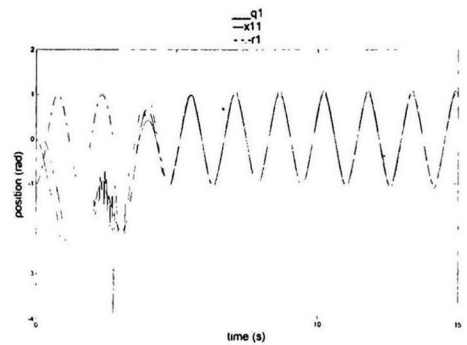


Fig. 9.  $q_1$  identification and tracking for the 2 DOF manipulator.

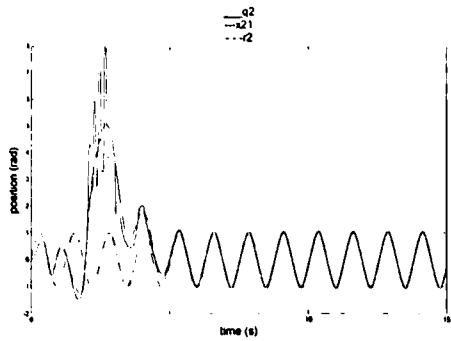


Fig. 10.  $q_2$  identification and tracking for the 2 DOF manipulator.

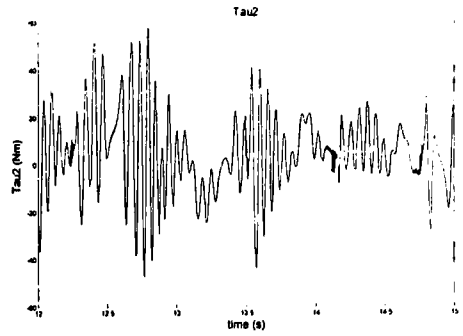


Fig. 14.  $\tau_2$  control for the 2 DOF manipulator.

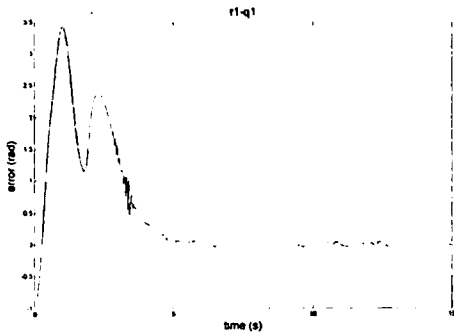


Fig. 11.  $q_1$  tracking error for the 2 DOF manipulator.

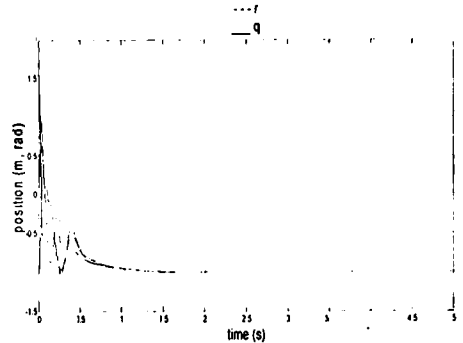


Fig. 15.  $q$  regulation for the ANAT manipulator.

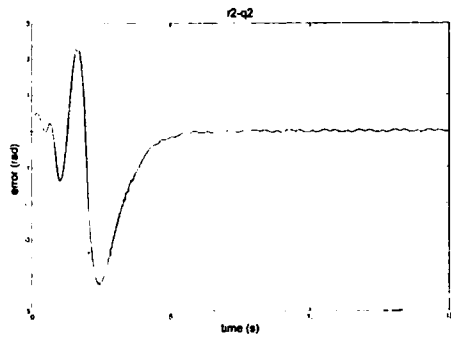


Fig. 12.  $q_2$  tracking error for the 2 DOF manipulator.

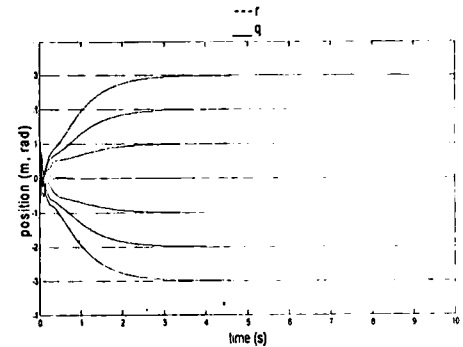


Fig. 16.  $q$  regulation for the ANAT manipulator.

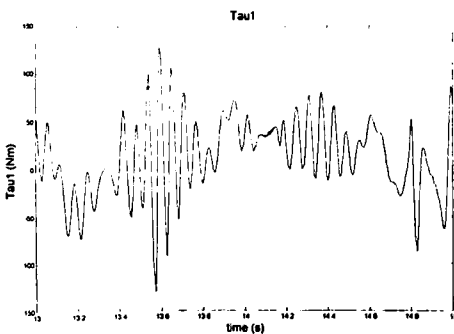


Fig. 13.  $\tau_1$  control for the 2 DOF manipulator.

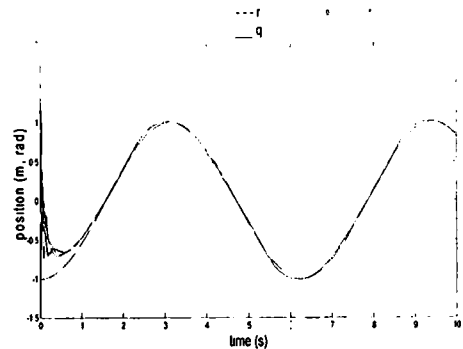


Fig. 17.  $q$  tracking for the ANAT manipulator.



research proceeds to test in real-time the whole scheme for the ANAT case.

## REFERENCES

- [1] R. Gourdeau, "Object-oriented programming for robotic manipulator simulation", *IEEE Robotics & Automation*, vol. 4, no. 3, pp. 21-29, 1997.
- [2] L. Fu, "Robust adaptive decentralized control of robot manipulators", *IEEE Trans. on Automatic Control*, vol. 37, no. 1, pp. 106-110, 1992.
- [3] A. Karakasoglu, S. I. Sudharsanan, and M. K. Sundareshan, "Identification and decentralized adaptive control using dynamical neural networks with application to robotic manipulators", *IEEE Trans. on Neural Networks*, vol. 4, no. 6, pp. 919-930, 1993.
- [4] S. Huang, K. K. Tan, and T. H. Lee, "Decentralized control design for large-scale systems with strong interconnections using neural networks", *IEEE Trans. on Automatic Control*, vol. 48, no. 5, pp. 805-810, 2003.
- [5] Z. P. Jiang, "New results in decentralized adaptive nonlinear control with output-feedback", *Proceedings of the 38<sup>th</sup> Conference on Decision & Control*, Phoenix, Arizona, USA, December 1999.
- [6] V. H. Benitez, A. G. Loukianov, and E. N. Sanchez, "Neural Identification and Control of a Linear Induction Motor Using an  $\alpha$ - $\beta$  Model", *American Control Conference*, Denver, Colorado, USA, June 2003.
- [7] M. Liu, "Decentralized control of robot manipulators: nonlinear and adaptive approaches", *IEEE Trans. on Automatic Control*, vol. 44, no. 2, pp. 357-363, 1999.
- [8] Y. Jin, "Decentralized adaptive fuzzy control of robot manipulators", *IEEE Transactions on Systems, Man, and Cybernetics-Part B: Cybernetics*, vol. 28, no. 1, February 1998.
- [9] E. B. Kosmatopoulos, M. A. Christodoulou, and P. A. Ioannou, "Dynamical neural networks that ensure exponential identification error convergence", *IEEE Neural Networks*, vol. 10, no. 2, pp. 299-314, 1997.
- [10] E. B. Kosmatopoulos, M. M. Polycarpou, M. A. Christodoulou, and P. A. Ioannou, "High-order neural networks structures for identification of dynamical systems", *IEEE Trans. on Neural Networks*, vol. 6, no. 2, pp. 422-431, 1995.
- [11] K. S. Fu, R. C. González, y C. S. G. Lee, *Robótica: Control, detección, visión e inteligencia*, 1<sup>a</sup> ed., pp. 85-106, McGraw-Hill, México, 1993.
- [12] A. G. Loukianov, "Robust Block Decomposition Sliding Mode Control Design", *Mathematical Problems in Engineering*, vol. 8(4-5) pp. 349-365, 2002.
- [13] V. Utkin, "Block Control Principle for Mechanical Systems", *Journal of Dynamic Systems, Measurement, and Control*, vol. 122, pp. 1-10, 2000.
- [14] B. Le Boudec, M. Saad, and V. Nerguizian, "Adaptive control of redundant robots", *8<sup>th</sup> International Conference on Modeling and Simulation of Electric Machines, Converters and Systems*, Hammamet, Tunisia, April 2005.
- [15] R. -J. Wai, "Tracking control based on neural network strategy for robot manipulators", *Neurocomputing*, 51, pp. 425-445, 2003.

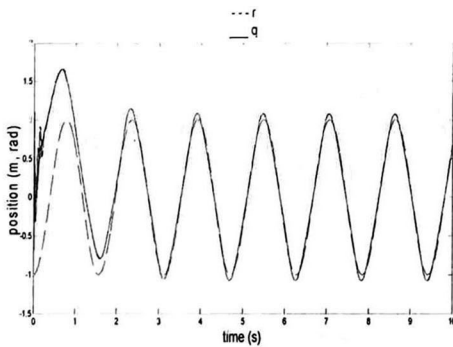


Fig. 18.  $q$  tracking for the ANAT manipulator.

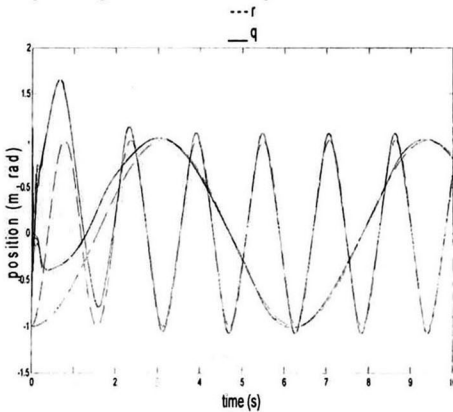


Fig. 19.  $q$  tracking for the ANAT manipulator.

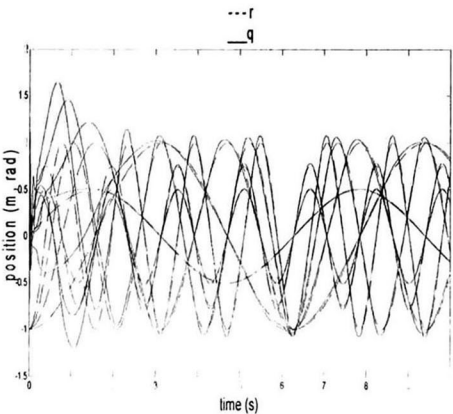


Fig. 20.  $q$  tracking for the ANAT manipulator

Again, the obtained results are very satisfactory

## V. CONCLUSIONS

A decentralized neural identification and control scheme is proposed for trajectory tracking of robotic manipulators. It is tested via simulations, using a 2 DOF manipulator and an ANAT one. The obtained results are very encouraging. The



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

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

Control Neuronal Descentralizado de Manipuladores Robóticos

del (la) C.

Armando GAYTÁN GODÍNEZ

el día 18 de Diciembre de 2006.

Dr. Bernardino Castillo Toledo  
Investigador CINVESTAV 3C  
CINVESTAV Unidad Guadalajara

Dr. Eduardo José Bayro Corrochano  
Investigador CINVESTAV 3C  
CINVESTAV Unidad Guadalajara

Dr. Alexander Georgievich Loukianov  
Investigador CINVESTAV 3C  
CINVESTAV Unidad Guadalajara





CINVESTAV  
BIBLIOTECA CENTRAL



SSIT000006598