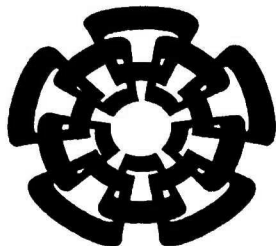




x(101587 1)





# CINVESTAV

Centro de Investigación de Estudios Avanzados del IPN  
Unidad Guadalajara

---

---

## CONTROL NEURONAL ADAPTABLE PARA SEGUIMIENTO DE TRAYECTORIAS



TESIS QUE PRESENTA  
**LUIS JOSUÉ RICALDE CASTELLANOS**

PARA OBTENER EL GRADO DE  
**MAESTRO EN CIENCIAS**

EN LA ESPECIALIDAD DE  
**INGENIERÍA ELÉCTRICA**

Guadalajara, Jal. Diciembre de 2001.

CLASIF.	.....
ADQUIS.	tesis - 2002
FECHA:	6 agosto 02
PROCED.	Serv. Bibli.

# **CONTROL NEURONAL ADAPTABLE PARA SEGUIMIENTO DE TRAYECTORIAS**

Tesis de Maestría en Ciencias  
Ingeniería Eléctrica

Por:

**Luis Josué Ricalde Castellanos**

Ingeniero Mecánico  
Instituto Tecnológico de Mérida

Becario de CONACYT, expediente No. **143768**

Directores de Tesis:  
**Dr. Edgar Nelson Sánchez Camperos**

CINVESTAV del IPN Unidad Guadalajara, Diciembre de 2001

*Dedicado a:*

*Luz María, Luis Manuel*

*Ana Laura y Mauricio*

*gracias por su apoyo*



# *Agradecimientos*

---

*Al Dr. Edgar Sánchez por todo el tiempo y apoyo ofrecido; por motivarme a lo largo de estos dos años.*

*A Bernardo Rincón y Víctor Flores por ser un modelo para mí.*

*A Italia Jiménez por haberme brindado su amistad y apoyo en los momentos difíciles.*

*A Omar Espinosa y Jorge Orozco por su amistad y apoyo incondicional.*

*A Clelia De la Peña, por haberme motivado a iniciar este proyecto, por toda la comprensión y sacrificio.*

*A Beatriz Salazar por todo su apoyo y amistad a lo largo de mi vida.*

*A mis compañeros de la generación 1999 de Control Automático, Rocío, Salvador, Juan J., Carlos, Juan C., Iván, Jaime, José L. y Cecilia, por ser un grupo tan unido; he aprendido mucho de ustedes.*

*A Eric, Javier, Iván, Alberto y Miguel, por ser mis amigos de toda la vida.*

*A todos ellos, muchas gracias.*

---

# Índice General

<b>1</b>	<b>Introducción</b>	<b>5</b>
<b>2</b>	<b>Fundamentos</b>	<b>7</b>
2.1	Sistemas no lineales y caos.	7
2.1.1	Introducción.	7
2.1.2	Aspectos teóricos.	7
2.1.3	Caos	9
2.1.4	Ejemplos de sistemas caóticos continuos, trayectorias y caracterización.	13
2.2	Conceptos de estabilidad de sistemas no lineales.	19
2.3	Redes Neuronales dinámicas de alto orden.	25
2.3.1	Redes neuronales en control.	25
<b>3</b>	<b>Control neuronal para seguimiento de trayectorias.</b>	<b>28</b>
3.1	Introducción.	28
3.2	Método de estabilización cuadrática.	29
3.2.1	Modelado de la planta	29
3.2.2	Análisis de seguimiento de trayectorias	29
3.2.3	Estabilización del error de seguimiento de trayectorias.	30
3.3	Método del control óptimo inverso.	32
3.3.1	Modelado de la planta	32
3.3.2	Análisis de seguimiento de trayectorias	32
3.3.3	Estabilización de la dinámica del error	33
3.3.4	Funcional a Optimizar	35
<b>4</b>	<b>Control de sistemas caóticos</b>	<b>37</b>
4.1	Introducción	37
4.2	Estabilización de sistemas caóticos.	38
4.2.1	Estabilización del sistema de Roessler.	38
4.2.2	Estabilización del sistema de Chua.	40
4.3	Producción de caos.	41
4.3.1	Seguimiento del atractor caótico de Chua.	42
4.3.2	Seguimiento del atractor caótico de Chen.	44
4.4	Sincronización de caos.	45

<b>5</b>	<b>Control de robots completamente actuados.</b>	<b>53</b>
5.1	Introducción.	53
5.2	Control de un manipulador completamente actuado.	54
5.2.1	Descripción del algoritmo.	54
5.2.2	Modelado del manipulador	57
5.2.3	Resultados en simulación utilizando Matlab.	61
<b>6</b>	<b>Conclusiones y trabajo futuro</b>	<b>66</b>
6.1	Conclusiones	66
6.2	Trabajo Futuro	67
	<b>APENDICE 1. Códigos en Matlab y diagramas en Simulink</b>	<b>71</b>
	<b>APENDICE 2. Artículos enviados</b>	<b>78</b>

# Índice de Figuras

2.1	Trayectorias caóticas, sistemas de Chen y Chua	9
2.2	Orbitas del Oscilador de Duffing, $w = 1.0$ , $\gamma = 0.30$ , $\delta = 0.15$ .	14
2.3	Mapa de Poincaré del oscilador de Duffing	14
2.4	Atractor caótico de Lorenz	15
2.5	Exponentes de Lyapunov del sistema de Lorenz	15
2.6	Atractor caótico de Chen	16
2.7	Exponentes de Lyapunov del sistema de Chen	16
2.8	Circuito de Chua	17
2.9	Función lineal a tramos del circuito de Chua	18
2.10	Atractor caótico de Chua con no linealidad dada por (2.27)	19
2.11	Atractor caótico de Chua con no linealidad dada por (2.29)	19
2.12	Exponentes de Lyapunov para el sistema de Chua	20
4.1	Esquema de control neuronal adaptable	38
4.2	Diagrama de fase del oscilador de Roessler	39
4.3	Estabilización del sistema de Roessler, evolución en el tiempo para $x(0) = (0, -10, 0)$ .	40
4.4	Espacio de fase de la estabilización del sistema de Roessler	40
4.5	Estabilización del sistema de Roessler, evolución en el tiempo de los pesos de la red neuronal.	41
4.6	Estabilización del sistema de Chua usando funciones cuadráticas de Lyapunov	41
4.7	Plano de fase del sistema de Chua estabilizado mediante funciones cuadráticas de Lyapunov	42
4.8	Evolución en el tiempo de los pesos de la red neuronal para la estabilización del sistema de Chua	42
4.9	Seguimiento del atractor de Chua usando estabilización cuadrática. Evolución en el tiempo de los estados.	43
4.10	Seguimiento del sistema de Chua usando estabilización cuadrática. Acción de control	44
4.11	Seguimiento del sistema de Chua usando estabilización cuadrática: diagrama de fase de la referencia y planta	45
4.12	Seguimiento del sistema de Chua usando el método del control óptimo inverso. Evolución de los estados.	45
4.13	Seguimiento del sistema de Chua usando el método del control óptimo inverso. Respuesta de control.	46
4.14	Seguimiento del sistema de Chua usando el método del control óptimo inverso. Diagramas de fase de referencia y planta	46



4.15	Seguimiento del sistema de Chua dado por (2.26) y (2.27) usando el método del control óptimo inverso. Evolución de los estados.	47
4.16	Seguimiento del sistema de Chua dado por (2.26) y (2.27) usando el método del control óptimo inverso. Acción de control obtenida con el método de estabilización cuadrática	47
4.17	Planos de fase de referencia y planta para seguimiento del sistema de Chua dado por (2.26) y (2.27) usando control óptimo inverso	48
4.18	Seguimiento del sistema de Chen usando estabilización cuadrática	48
4.19	Diagrama de fase de referencia y planta para seguimiento del sistema de Chen usando estabilización cuadrática	49
4.20	Evolución de los estados para seguimiento del sistema de Chen usando control óptimo inverso.	49
4.21	Diagrama de fase de referencia y planta para seguimiento del sistema de Chen usando control óptimo inverso.	50
4.22	Sincronización de caos mediante estabilización cuadrática: evolución en el tiempo de los estados.	50
4.23	Sincronización de caos mediante estabilización cuadrática: diagramas de fase de referencia y planta	50
4.24	Sincronización de caos mediante control óptimo inverso: evolución en el tiempo del estado 1.	51
4.25	Sincronización de caos mediante control óptimo inverso: evolución en el tiempo del estado 2	51
4.26	Sincronización de caos mediante control óptimo inverso: evolución en el tiempo del estado 3	51
4.27	Sincronización de caos mediante control óptimo inverso: diagrama de fase de la referencia (Atractor de Lorenz)	52
4.28	Sincronización de caos mediante control óptimo inverso: diagrama de fase de la planta (Atractor de Chen)	52
5.1	Manipulador planar de dos grados de libertad	58
5.2	Evolución en el tiempo para la posición del eslabón 1	63
5.3	Evolución en el tiempo para la posición del eslabón 2	63
5.4	Torque aplicado al eslabón 1	63
5.5	Torque aplicado al eslabón 2	64
5.6	Perturbación de torque aplicada al primer eslabón	64
5.7	Posición angular de los eslabones 1,2 y perturbación de par de tipo senoidal con frecuencia variable para el manipulador completamente actuado.	64
5.8	Torques aplicados cuando se tiene una perturbación senoidal de frecuencia variable.	65
6.1	Esquema de simulación en Simulink para el control neuronal adaptable para el manipulador robótico	75
6.2	Diagrama de bloques en Simulink de la red neuronal .	76
6.3	Diagrama de bloques en Simulink del modelo del manipulador robótico	76
6.4	Ley de adaptación de pesos (a)	77
6.5	Ley de adaptación de pesos (b)	77

# Capítulo 1

## Introducción

A partir de la publicación [19], ha habido un continuo incremento en el interés por aplicar redes neuronales en identificación y control de sistemas no lineales. La mayoría de estas aplicaciones usan estructuras estáticas [7],[9]. Últimamente se ha desarrollado el uso de redes neuronales dinámicas en control; permiten un modelado más eficaz de los sistemas dinámicos mencionados [21]. Dos libros recientes [22], [23] revisan la aplicación de redes neuronales dinámicas para identificación y control. En [22], en particular, se utiliza un aprendizaje fuera de línea, mientras que en [23] se analiza la identificación adaptable y el control por medio de aprendizaje en línea; la estabilidad en lazo cerrado se establece basándose en métodos de función de Lyapunov. En [23] el problema de seguimiento de trayectorias es reducido a un problema de seguimiento lineal, con aplicación a motores de corriente directa. En el presente trabajo se presenta un algoritmo de control para seguimiento de trayectorias en sistemas no lineales basado en un aprendizaje en línea para crear un modelo y funciones cuadráticas de Lyapunov para la obtención de la ley de control..

Por otro lado, los métodos de control aplicados a sistemas no lineales en general han sido extensamente desarrollados, por ejemplo, basándose en la teoría de geometría diferencial [10]. Recientemente, el enfoque de pasividad ha generado gran interés para sintetizar nuevas leyes de control de sistemas no lineales [8]. Un problema importante usando este enfoque es cómo lograr control no lineal robusto en la presencia de dinámicas no modeladas y perturbaciones externas. Esto motivó la aparición del control no lineal  $H_\infty$  [13], [1]. Una de las mayores dificultades de este método, además de la posible inestabilidad estructural del sistema, es el resolver el sistema de ecuaciones diferenciales parciales asociado. Con el propósito de reducir este problema computacional, el llamado método del control óptimo inverso fue recientemente desarrollado con base en el concepto de estabilidad entrada- estado [15]. Usando el método del control óptimo inverso, una ley de control, capaz de producir caos en una red neuronal dinámica, fue presentada en [24]. El presente trabajo extiende éstos resultados al seguimiento de trayectorias en sistemas dinámicos desconocidos.

Los esquemas de control adaptable propuestos están compuestos de un identificador neuronal dinámico, el cual construye en línea un modelo para la planta desconocida y una ley de control, la cual asegura que la planta desconocida siga la trayectoria de referencia satisfactoriamente; en el caso del control óptimo inverso se demuestra que esta acción de control es óptima con respecto a un funcional de costo.



Controlar los sistemas caóticos, es relativamente un nuevo concepto [2]. Estos sistemas no siguen trayectorias predecibles, al evolucionar en formas irregulares. Por lo tanto son sistemas a considerar al desarrollar algoritmos de seguimiento de trayectorias, dada la complejidad de estos sistemas.

El control para seguimiento de trayectorias es una aplicación importante en el control de robots. La dinámica de los robots es altamente no lineal por lo que ha habido un incremento en la búsqueda de controladores debido a la necesidad de obtener un desempeño satisfactorio en presencia de incertidumbres en el modelado y variación en la carga. Una de las áreas de mayor investigación es el control adaptable, el cual presenta un buen desempeño en presencia de las incertidumbres mencionadas. Se ha motivado la línea de control mediante redes neuronales por el hecho de que el modelo exacto de la planta no requiere ser conocido. En este trabajo se presenta la unificación de ambas corrientes para obtener un control adaptable usando redes neuronales. El combinar ambos métodos presenta las ventajas de obtener un control en base a un modelo adaptable y que es robusto; con esto se resuelve el problema de la falta de robustez ante no linealidades desconocidas usando una adaptación de parámetros para compensar estas incertidumbres.

En esta memoria se presentan dos aplicaciones para los algoritmos desarrollados en el seguimiento de trayectorias. Se considera el control de sistemas caóticos en problemas de estabilización, seguimiento y sincronización de sistemas caóticos. Por otra parte se aplica el método del control óptimo inverso para seguimiento de trayectorias en manipuladores robóticos con énfasis en plantas con dinámicas no modeladas y rechazo a perturbaciones.

La tesis se organiza de la siguiente forma: en el Capítulo Dos se presentan brevemente los sistemas caóticos y su caracterización; se definen las redes neuronales dinámicas a utilizar como identificadores en los algoritmos; finalmente se discuten los conceptos de estabilidad en sistemas no lineales, que servirán como base para el desarrollo de los algoritmos de control neuronal adaptable. En el tercer Capítulo se desarrollan los algoritmos de estabilización cuadrática y control óptimo inverso para control neuronal adaptable para seguimiento de trayectorias en sistemas no lineales. Estos algoritmos se aplican en el Capítulo Cuatro para controlar sistemas caóticos. En el Capítulo Cinco se extienden los resultados obtenidos en el Capítulo dos para un sistema con menos entradas que estados; se aplica la metodología desarrollada para el seguimiento de trayectoria de un manipulador robótico de dos grados de libertad. En el Capítulo seis se presentan las conclusiones y el trabajo futuro.

Los Capítulos Tres y Cuatro se desarrollaron en estrecha colaboración con José Paz Perez a quien le agradezco las ideas aportadas.

# Capítulo 2

## Fundamentos

*En este capítulo se incluye una breve descripción de los sistemas caóticos, los cuales se controlarán mediante los algoritmos a presentarse en los capítulos subsecuentes. También se discuten brevemente las redes neuronales recurrentes de alto orden a utilizarse como identificadores. Finalmente se presentan los conceptos de estabilidad entrada-estado que sirven de base para derivar la teorías de estabilización cuadrática y control óptimo inverso a presentar en este trabajo.*

### 2.1 Sistemas no lineales y caos.

#### 2.1.1 Introducción.

En años recientes se ha incrementado el estudio de los sistemas no lineales. Esto, en parte, es debido a la disponibilidad de sistemas computacionales de mayor capacidad, puesto que se depende mucho de simulaciones numéricas para analizar los fenómenos no lineales.

Un acontecimiento importante fue el descubrimiento del caos, el cual elimina la creencia de poder predecir el comportamiento de un sistema determinístico. El caos (o comportamiento extraño) se puede definir como el comportamiento aleatorio de un sistema determinístico; es decir, representa la evolución inesperada de un sistema.

El comportamiento caótico ha jugado un importante papel en sistemas dinámicos que incluyen: mapeos matemáticos, sistemas Hamiltonianos, sistemas eléctricos, dinámica de fluidos, óptica no lineal, reacciones químicas, redes neuronales y comportamiento biológico.

El propósito de esta sección es explicar cómo el comportamiento de un sistema determinístico puede llegar a ser impredecible y conocer las técnicas para clasificar el comportamiento en estado estable de los sistemas no lineales con énfasis en el comportamiento extraño así como los métodos más utilizados para poder detectar el comportamiento caótico.

#### 2.1.2 Aspectos teóricos.

Los conceptos presentados en esta sección fueron tomados de [2], [3], [6], [12], [20], y [31].



**Sistemas dinámicos autónomos.**

Considérese el siguiente campo vector

$$\dot{x} = f(x) \quad x(t_0) = x_0 \quad x \in \mathbb{R}^n \quad f : \mathbb{R}^n \rightarrow \mathbb{R}^n \quad (2.1)$$

El sistema es lineal si el mapeo  $f$  es lineal.

La solución de (2.1) con condición inicial  $x_0$  es llamada trayectoria  $\phi_t(x_0)$ , donde  $\phi_t : \mathbb{R}^n \rightarrow \mathbb{R}^n$  es el flujo.

**Sistemas dinámicos no autónomos.** Considérese el sistema,

$$\dot{x} = f(x, t) \quad x(t_0) = x_0 \quad x \in \mathbb{R}^n \quad f : \mathbb{R}^n \rightarrow \mathbb{R}^n \quad (2.2)$$

El campo vectorial  $f$  depende del tiempo  $t$  y su solución está dada por  $\phi_t(x_0, t_0)$ .

El sistema será periódico en el tiempo si  $\exists T > 0$  tal que  $f(x, t) = f(x, t + T)$  donde  $T$  es llamado el período mínimo. Todo sistema periódico de dimensión  $n$ , puede ser convertido en un sistema autónomo de dimensión  $(n + 1)$ , añadiendo un estado  $\theta = \frac{2\pi t}{T}$ , tal que:

$$\begin{aligned} \dot{x} &= f\left(x, \frac{\theta T}{2\pi}\right) \\ \dot{\theta} &= \frac{2\pi}{T} \end{aligned} \quad (2.3)$$

**Propiedades de los sistemas dinámicos.**

i)  $\phi_t(x) = \phi_t(y) \iff x = y$

ii)  $\phi_t(x, t_0) = \phi_t(y, t_0) \iff x = y$

Si  $t_0 \neq t_1$  es posible que  $\phi_t(x, t_0) = \phi_t(y, t_1)$  y  $x \neq y$

iii)  $\phi_t(x_0, t_0)$  es continua con respecto a su condición inicial

**Sistemas dinámicos en tiempo discreto.**

$$x_{k+1} = f(x_k) \quad k = 1, 2, \dots \quad (2.4)$$

donde  $x_k$  es el estado y el mapeo  $f$  produce una secuencia de puntos  $\{x_k\}_{k=0}^{\infty}$  llamada órbita.

**Comportamiento en estado estable y conjuntos límite.** El término comportamiento en estado estable se refiere al comportamiento asintótico cuando  $t \rightarrow \infty$ ; en general se desea que el estado estable sea acotado.

Existen cuatro tipos de comportamiento en estado estable:

- **Puntos de equilibrio.** Un punto de equilibrio  $x_{eq}$  de un sistema autónomo es una solución constante de (2.1),  $f(x) = 0$  implica que  $x$  es un punto de equilibrio.

- **Soluciones periódicas.**  $\phi_t(x^*, t_0)$  es una solución periódica si

$$\phi_t(x^*, t_0) = \phi_{t+T'}(x^*, t_0) \tag{2.5}$$

para todo  $t$  y algún período mínimo  $T'$

En el caso no autónomo una solución periódica aislada  $\phi_t(x^*, t_0)$  es llamada ciclo límite; ésta es una oscilación sostenida y no puede ocurrir en un sistema lineal.

- **Soluciones cuasiperiódicas.** Son aquéllas que pueden ser escritas como la suma de soluciones periódicas y cuyas frecuencias son el resultado de la suma de un conjunto finito de las frecuencias base.
- **Caos.** Se puede definir como un comportamiento en estado estable acotado; pero no es un punto de equilibrio, ni es periódico y tampoco es cuasiperiódico. En la siguiente sección se presentarán las características que denotan un comportamiento caótico.

### 2.1.3 Caos

Consideremos la figura 2.1, donde se muestran algunas trayectorias caóticas; a partir de éstas se puede ver que las trayectorias están acotadas y que no son periódicas ni cuasiperiódicas

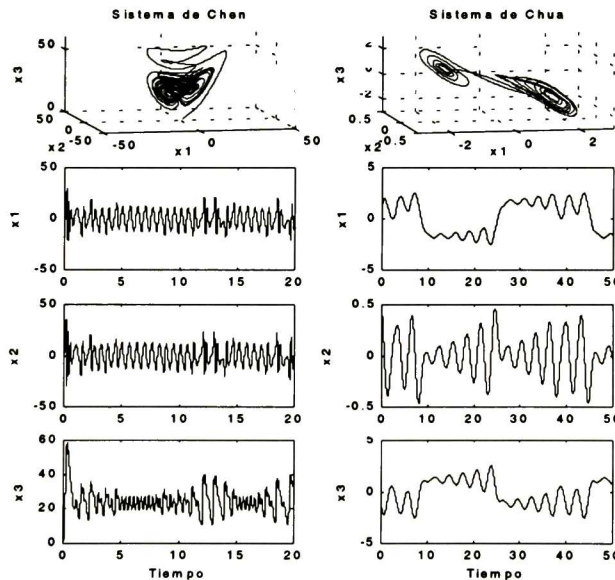


Figura 2.1: Trayectorias caóticas, sistemas de Chen y Chua

#### Características generales del caos.

- **Dinámica no lineal.** El caos es un fenómeno exclusivo de los sistemas no lineales.

- **Sensibilidad a condiciones iniciales (SCI).** Las órbitas que parten de valores iniciales muy cercanos se comportan de forma independiente, acercándose y alejándose de forma impredecible.
- **Compacidad.** La SCI no es suficiente para generar un comportamiento caótico (considere un sistema no lineal inestable). Para que el caos pueda darse se requiere que las trayectorias evolucionen en un conjunto de volumen finito.
- **Existencia de órbitas periódicas de distintos períodos.** El caos se presenta en sistemas cuyo modelo matemático posee una constante, tal que al cambiar su valor, el sistema presentará cambios en su dinámica (bifurcaciones), dependiendo del valor de dicha constante.
- **Propiedad de mezcla.** La órbita de cualquier punto recorrerá casi todos los puntos de una zona del espacio de fase sin llenar dicha zona.
- **Presencia de atractores extraños.** En su diagrama de fases los sistemas caóticos presentan estructuras geométricas de forma poco usual y sin ningún parecido a los objetos geométricos clásicos; estos diagramas de fase son llamados atractores extraños.

### Caracterización de caos.

Por su definición, los sistemas caóticos tienen dos características importantes: un sistema caótico generará trayectorias que evolucionan en un volumen finito del espacio de fase y presenta zonas de gran densidad. Por lo tanto, si la trayectoria caótica está acotada, entonces se deberá contraer en algunas direcciones y expandirse en otras, con la contracción superando a la expansión. En esta sección se discutirán dos métodos para poder determinar si un sistema es caótico, con base en las características mencionadas.

**Mapa de Poincaré [6][31]** La idea de reducir el estudio de una trayectoria al estudio de un sistema discreto asociado (órbita) es debida a Poincaré (aplicado a mecánica celeste). El mapeo de Poincaré reemplaza el flujo de un sistema continuo de orden  $n$ , por un sistema discreto de orden  $n - 1$ .

**Definición 1** *Mapa de Poincaré cerca de una órbita periódica.*

Considere la ecuación diferencial ordinaria

$$\dot{x} = f(x) \quad x \in R^n \quad (2.6)$$

Supóngase que (2.6) tiene una solución periódica de período  $T$  la cual se denota por  $\phi(t, x_0)$ , donde  $x_0$  es un punto por donde pasa esta solución. Sea  $\Sigma$  una hipersuperficie de dimensión  $(n - 1)$  transversal al campo vector en  $x_0$ . Es posible hallar un conjunto abierto  $V \subset \Sigma$ , tal que las trayectorias que empiezan en  $V$  regresen a  $\Sigma$  en un tiempo cercano a  $T$ . El mapa, que asocia los puntos en  $V$  con los puntos que regresan primero a  $\Sigma$ , es llamado el mapa de Poincaré.

$$\begin{aligned} P & : & V & \rightarrow \Sigma \\ x & \rightarrow & \phi(\tau(x), x) \end{aligned} \quad (2.7)$$



donde  $\tau(x)$  es el tiempo para que la trayectoria que sale de  $x$  regrese a  $\Sigma$ . Por construcción  $\tau(x_0) = T$  y  $P(x_0) = x_0$ .

Por lo tanto un punto fijo de  $P$  corresponde a una órbita periódica de (2.6) y un punto de período  $k$  ( $P^k(x) = x$ ) corresponde a una órbita periódica de (2.6), que cruza la superficie  $\Sigma$   $k$  veces antes de cerrarse.

**Definición 2** *Mapa de Poincaré de una ecuación diferencial ordinaria periódica en el tiempo.*

Considere la siguiente ecuación diferencial ordinaria,

$$\dot{x} = f(x, t) \quad x \in \mathbb{R}^n \tag{2.8}$$

Supóngase que la dependencia con respecto al tiempo es periódica con período  $T = \frac{2\pi}{w}$  y reescriba el sistema como uno autónomo,

$$\begin{aligned} \dot{x} &= f(x, \theta) & (x, \theta) \in \mathbb{R}^n \times S^1 \\ \dot{\theta} &= w \end{aligned} \tag{2.9}$$

Sea el flujo generado por (2.8)  $\phi_t = (x(t), \theta(t) = wt + \theta_0 \bmod 2\pi)$ . Se define una sección  $\Sigma$  transversal al campo vector como

$$\Sigma = \{(x, \theta) \in \mathbb{R}^n \times S^1 \mid \theta = \theta_0 \in (0, 2\pi]\} \tag{2.10}$$

Entonces, el mapa de Poincaré se define como

$$P_{\theta} : \Sigma^{\theta} \rightarrow \Sigma^{\theta} \\ \left( x \left( \frac{\theta - \theta_0}{w} \right), \theta \right) \rightarrow \left( x \left( \frac{\theta - \theta_0 + 2\pi}{w}, \theta + 2\pi \equiv \theta \right) \right) \tag{2.11}$$

### Ventajas del mapa de Poincaré.

- **Reducción de la dimensión.** Elimina al menos una de las variables.
- **Dinámica global.** En problemas de dimensión  $\leq 4$ .
- **Claridad conceptual.** Muchos conceptos son más fáciles de asociar. Considérese el caso de la estabilidad de una órbita periódica; esta órbita mapea a un solo punto y sólo hay que analizar los valores propios del mapeo linealizado en el punto.

Desafortunadamente no existe una forma general para construir un mapa de Poincaré de un sistema de ecuaciones diferenciales arbitrario, puesto que se requiere conocimiento de la estructura geométrica del espacio de fase.



**Exponentes de Lyapunov.**[20][4] El segundo método (y el más utilizado) para caracterizar los sistemas caóticos es el cálculo de los exponentes de Lyapunov. Los exponentes de Lyapunov (o números característicos) fueron inicialmente utilizados por Lyapunov con el propósito de estudiar la estabilidad de soluciones no estacionarias de ecuaciones diferenciales ordinarias.

Considérese el sistema  $\dot{x} = f(x, t)$  y sea  $x(t_0) = x_0$  su trayectoria para algún valor de  $t_0$ , calcúlese el Jacobiano de  $f(x, t)$  a lo largo de esta trayectoria,

$$F(t) = D_x f(x, t) \tag{2.12}$$

Sea  $\Psi$  la matriz fundamental del sistema variante en el tiempo

$$\dot{\xi} = F(t)\xi \tag{2.13}$$

y sean  $m_i(t)$  los valores propios de  $\Psi(t)$ . Entonces los exponentes de Lyapunov se definen como,

$$\lambda_i = \lim_{t \rightarrow \infty} \frac{1}{t} \ln |m_i(t)| \quad i = 1, \dots, n \tag{2.14}$$

si el límite existe.

**Dimensión de Lyapunov.** Sean  $\lambda_1 \geq \lambda_2 \geq \dots \lambda_n$  los exponentes de Lyapunov de un sistema dinámico. Sea  $j$  el máximo entero tal que,

$$\lambda_1 + \lambda_2 + \dots + \lambda_j \geq 0 \tag{2.15}$$

La dimensión de Lyapunov  $D_L$  se define como

$$D_L = j + \frac{\lambda_1 + \lambda_2 + \dots + \lambda_j}{|\lambda_{j+1}|} \tag{2.16}$$

En un sistema caótico tridimensional con exponentes de Lyapunov  $\lambda_+, 0, \lambda_-$ , la dimensión de Lyapunov está dada por

$$D_L = 2 + \frac{\lambda_+}{|\lambda_-|} \tag{2.17}$$

Para la existencia de un atractor extraño se requiere  $\lambda_+ + \lambda_- < 0$  de donde  $2 < D_L < 3$

La relación expansión-contracción en un sistema caótico puede determinarse examinando (2.17) Como  $2 < D_L < 3$ , entonces

$$\frac{\lambda_+}{|\lambda_-|} < 1 \quad \lambda_+ < |\lambda_-| \tag{2.18}$$

Por lo tanto en un sistema caótico, la contracción supera a la expansión.

### 2.1.4 Ejemplos de sistemas caóticos continuos, trayectorias y caracterización.

En esta sección se presentan algunos sistemas caóticos, su descripción matemática, el plano de fase que muestra el atractor y su caracterización por medio del mapa de Poincaré (en el caso del sistema de Duffing) y los exponentes de Lyapunov (y el cálculo de su correspondiente dimensión de Lyapunov). El mapeo de Poincaré se realizó en Maple y para el cálculo de los exponentes de Lyapunov se utilizó el programa LET (Lyapunov Exponents Toolbox) hecho en Matlab. Los algoritmos para la obtención de los exponentes de Lyapunov requieren la solución de la ecuación variacional

$$\begin{Bmatrix} \dot{x} \\ \dot{\Phi} \end{Bmatrix} = F(x, \Phi, t) := \begin{Bmatrix} f(x, t) \\ D_x f(x, t)\Phi \end{Bmatrix} \quad (2.19)$$

Este sistema se integra para obtener así  $\phi_t(x_0, t_0)$  y  $\Phi_t(x_0, t_0)$  simultáneamente. El algoritmo para calcular todos los exponentes de Lyapunov se presenta en [20] y el cálculo de la dimensión de Lyapunov se realiza con la fórmula (2.16)

#### Oscilador de Duffing [31]

El oscilador de Duffing es el ejemplo más común de osciladores no lineales en textos y artículos de investigación. Duffing presentó en 1918 un oscilador no lineal con un término cúbico para describir el efecto de rigidez de los resortes observado en muchos problemas mecánicos. El sistema dinámico de Duffing puede escribirse como

$$\ddot{x} + \delta\dot{x} - x + x^3 = \gamma \cos wt$$

o en variables de estado

$$\begin{aligned} \dot{x}_1 &= x_2 \\ \dot{x}_2 &= x_1 - x_1^3 - \delta x_2 + \gamma \cos wt \end{aligned}$$

el cual es un modelo presentado por Moon y Holmes en 1979. La figura 2.2 muestra el plano de fase del oscilador de Duffing y la figura 2.3 muestra el mapa de Poincaré de dicho sistema. Puede observarse que los puntos resultantes del mapeo se encuentran concentrados en ciertas regiones pero sin repetir el mismo punto, lo cual es una característica de los atractores caóticos.

#### Sistema de Lorenz.[30]

En [17], Lorenz presentó un sistema de tres ecuaciones diferenciales. Estas representaban la aproximación por ecuaciones diferenciales ordinarias del sistema de ecuaciones diferenciales parciales de Navier-Stokes, para convección en un fluido viscoso con gradiente de temperatura en una capa bidimensional calentada por debajo.

Las ecuaciones del sistema de Lorenz son

$$\begin{aligned} \dot{x}_1 &= \sigma(x_2 - x_1) \\ \dot{x}_2 &= rx_1 - x_2 - x_2x_3 \\ \dot{x}_3 &= -bx_3 + x_1x_2 \end{aligned} \quad (2.20)$$

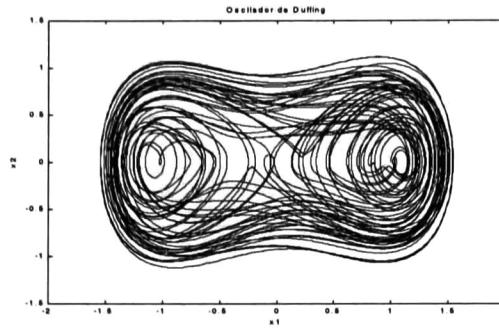


Figura 2.2: Orbitas del Oscilador de Duffing,  $w = 1.0$ ,  $\gamma = 0.30$ ,  $\delta = 0.15$ .

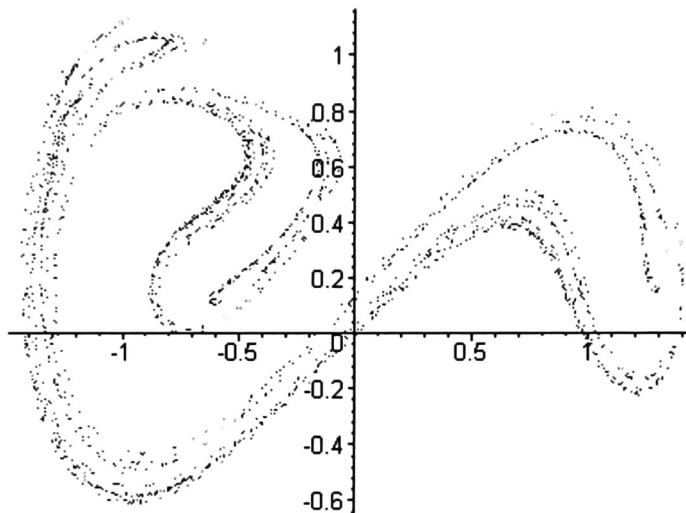


Figura 2.3: Mapa de Poincaré del oscilador de Duffing

donde

$$\sigma = 10, \quad b = 8/3, \quad r = 28, \quad \begin{pmatrix} x_1(0) \\ x_2(0) \\ x_3(0) \end{pmatrix} = \begin{pmatrix} 1 \\ 1 \\ 1 \end{pmatrix} \quad (2.21)$$

Con los parámetros anteriores, el sistema de Lorenz presenta un comportamiento caótico, acotado, de trayectorias densas y alta dependencia a las condiciones iniciales; el atractor de Lorenz se muestra en la figura 2.4.

**Exponentes de Lyapunov para el sistema de Lorenz.** Para calcular los exponentes de Lyapunov del sistema de Lorenz se realizó una simulación con el programa LET. Los resultados se presentan en la figura 2.5. En este caso los exponentes de Lyapunov son

$$\lambda_1 = 2.0669, \quad \lambda_2 = -0.0001, \quad \lambda_3 = -22.5045$$

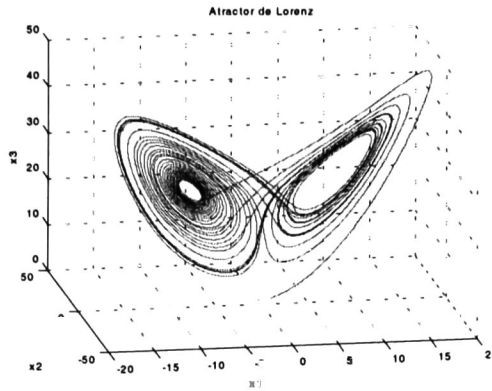


Figura 2.4: Atractor caótico de Lorenz

la dimensión de Lyapunov se calcula usando la ecuación (2.16).

$$D_L = 2.0669$$

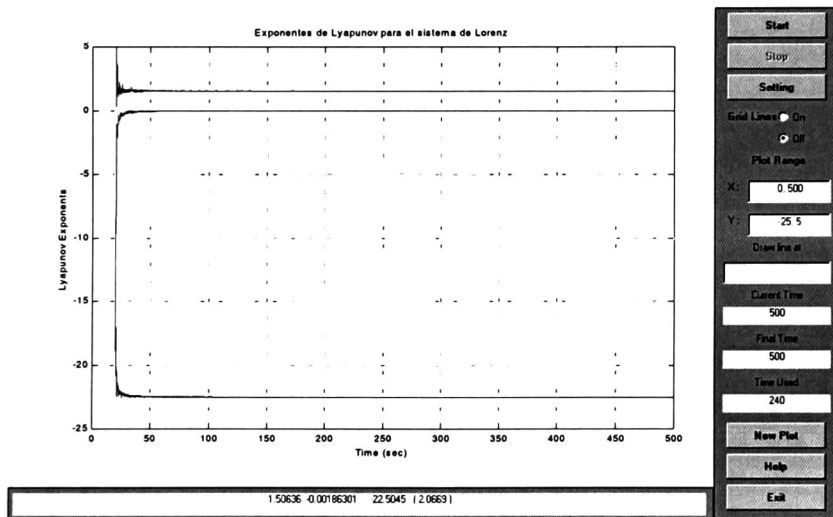


Figura 2.5: Exponentes de Lyapunov del sistema de Lorenz

### Sistema de Chen.[26]

El sistema de Chen es un sistema caótico descubierto recientemente, el cual ha demostrado ser particularmente difícil de controlar comparándolo con el atractor de Lorenz. El plano de fase del sistema de Chen (figura 2.6) muestra que este atractor es mucho más complejo, pues presenta un comportamiento tridimensional más rico.

El atractor de Chen está descrito por las siguientes ecuaciones,

$$\begin{aligned} \dot{x} &= a(y - x) \\ \dot{y} &= (c - a)x - xz + cy \\ \dot{z} &= xy - bz \end{aligned} \tag{2.22}$$

Para este caso ,  $a = 35, b = 3, c = 28$

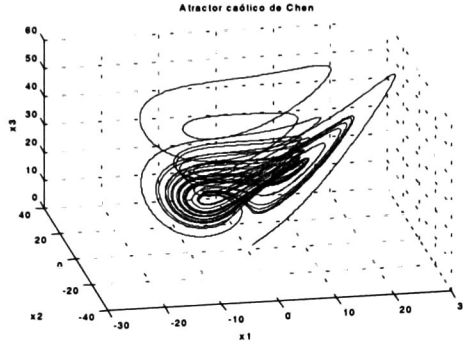


Figura 2.6: Atractor caótico de Chen

Exponentes de Lyapunov para el sistema de Chen.

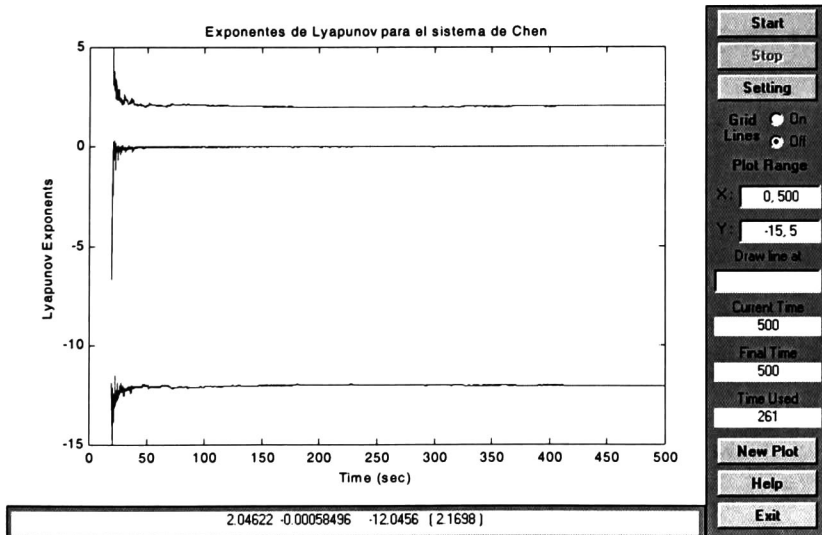


Figura 2.7: Exponentes de Lyapunov del sistema de Chen

Estos se calculan por medio del programa LET Los resultados son

$$\lambda_1 = 2.020 \quad \lambda_2 = 0.0078 \quad \lambda_3 = -12.028$$

Este sistema presenta las características de un sistema caótico de dimensión tres: un exponente de Lyapunov positivo, uno cero y el tercer exponente negativo.

**Sistema de Chua [3][20].** El circuito de Chua se muestra en la figura 2.8; consta de un inductor  $L$ , dos capacitores  $C_1, C_2$ , un resistor lineal y un resistor lineal a tramos  $g$ .

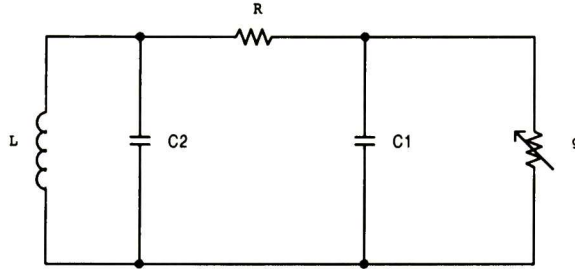


Figura 2.8: Circuito de Chua

La ecuación dinámica del circuito de Chua está descrita por

$$\begin{cases} C_1 \dot{v}_{C_1} = \frac{1}{R} (v_{C_2} - v_{C_1}) - g(v_{C_1}) \\ C_2 \dot{v}_{C_2} = \frac{1}{R} (v_{C_1} - v_{C_2}) - i_L \\ L \dot{i}_L = -v_{C_2} \end{cases} \quad (2.23)$$

donde  $i_L$  es la corriente a través del inductor  $L$ ,  $v_{C_1}$  y  $v_{C_2}$  son los voltajes a través de  $C_1$  y  $C_2$  respectivamente, y

$$g(v_{C_1}) = m_0 v_{C_1} + \frac{1}{2} (m_1 - m_0) (|v_{C_1} + 1| - |v_{C_1} - 1|) \quad (2.24)$$

con  $m_0 < 0$  y  $m_1 < 0$  constantes apropiadamente elegidas. La función lineal a tramos es mostrada en la figura 2.9. Definiendo  $p = \frac{C_2}{C_1} > 0$  y  $q = \frac{C_2 R^2}{L} > 0$  y haciendo el siguiente cambio de variables

$$\begin{aligned} x(\tilde{t}) &= v_{C_1}(\tilde{t}) & y(\tilde{t}) &= v_{C_2}(\tilde{t}) \\ z(\tilde{t}) &= R i_L(\tilde{t}) & \tilde{t} &= \frac{t}{(C_2 R)} \end{aligned} \quad (2.25)$$

Las ecuaciones del circuito de Chua se pueden reformular como

$$\begin{aligned} \dot{x} &= p(-x + y - \Phi(x)) \\ \dot{y} &= x - y + z \\ \dot{z} &= -qy \end{aligned} \quad (2.26)$$



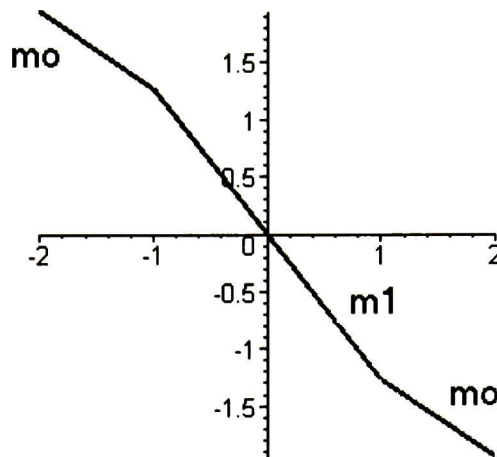


Figura 2.9: Función lineal a tramos del circuito de Chua

donde

$$\Phi(x) = \begin{cases} m_0x - m_1 & \text{si } x \leq -1 \\ m_1x & \text{si } |x| < 1 \\ m_0x + m_1 & \text{si } x \geq 1 \end{cases} \quad (2.27)$$

para este caso  $p = 10, q = 14.87, m_0 = -0.68, m_1 = -1.27$ .

Una forma alternativa para las ecuaciones del circuito de Chua es

$$\begin{aligned} \dot{x} &= \alpha(y - x) - \Phi(x) \\ \dot{y} &= \rho(x - y) + z \\ \dot{z} &= -\beta y \end{aligned} \quad (2.28)$$

$$\Phi(x) = -9(0.52x + 0.15(|x + 1| - |x - 1|)) \quad (2.29)$$

Para la figura 2.11, se utiliza  $\alpha = 6.3, \beta = 7, \rho = 0.7$ .

Ambos atractores se muestran en las figuras 2.10 y 2.11.

**Exponentes de Lyapunov para el sistema de Chua.** Para el cálculo de los exponentes de Lyapunov del circuito de Chua se utilizó el sistema descrito por (2.26) y (2.27); se utilizó el programa LET para calcular los exponentes de Lyapunov y se obtuvieron los siguientes resultados:

$$\lambda_1 = 0.4219 \quad \lambda_2 = 0.0014 \quad \lambda_3 = -3.8432$$

De nuevo hay un exponente de Lyapunov positivo y un exponente es negativo de valor absoluto mayor que  $\lambda_1$ .

La dimensión de Lyapunov para el sistema de Chua es:

$$D_L = 2.1102$$

Este resultado es similar al que se presenta en [20].



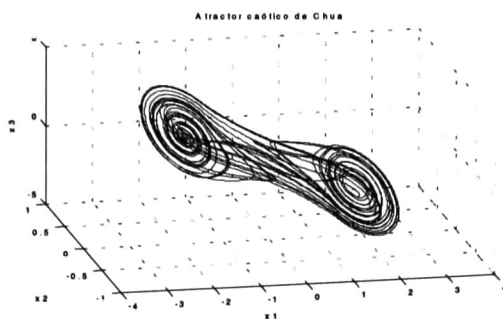


Figura 2.10: Atractor caótico de Chua con no linealidad dada por (2.27)

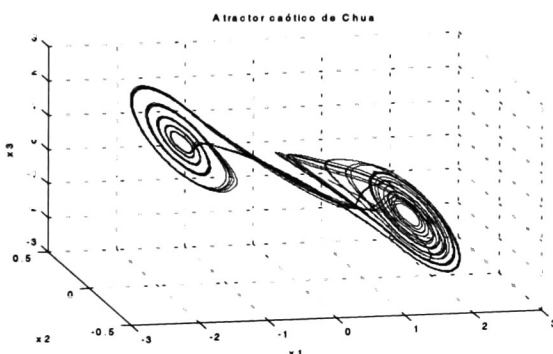


Figura 2.11: Atractor caótico de Chua con no linealidad dada por (2.29)

## 2.2 Conceptos de estabilidad de sistemas no lineales.

Los métodos de análisis de los algoritmos presentados en este trabajo están basados en la teoría de estabilidad y estabilización para sistemas no lineales, tanto con incertidumbres como sin ellas; estos conceptos se toman de [12]. Se utilizan funciones de Lyapunov de control (clf de "Control Lyapunov Functions") y su extensión a funciones adaptables para seguimiento de trayectorias. Por medio del uso de estas funciones es posible asegurar la estabilidad asintótica del error de seguimiento de trayectorias. Finalmente se puede diseñar una acción de control mediante el método del control óptimo inverso; dicha acción optimiza un funcional de costo relacionado con un sistema de ecuaciones diferenciales parciales de Hamilton-Jacobi-Bellman. Este método se presenta en [15].

### Definición 3 Función Clase $K$ [12]

Una función continua  $\alpha : \mathbb{R}^+ \rightarrow \mathbb{R}^+$  se dice que pertenece a la clase  $K$ , si es estrictamente creciente y  $\alpha(0) = 0$ . Además, se dice que pertenece a la clase  $K_\infty$  si  $\alpha \in K$  y  $\alpha(r) \rightarrow \infty$  cuando  $r \rightarrow \infty$ .

### Definición 4 Función Clase $KL$ [12]

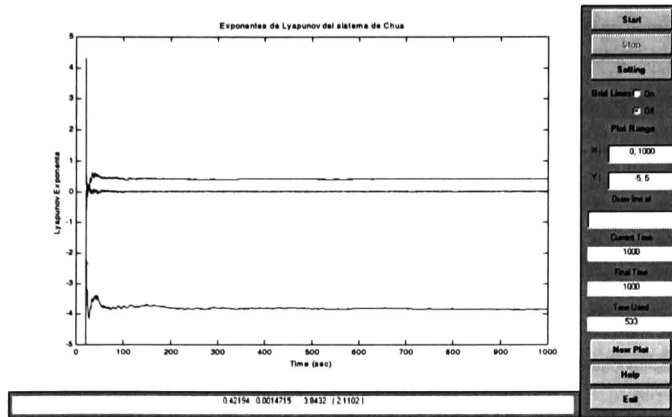


Figura 2.12: Exponentes de Lyapunov para el sistema de Chua

Una función continua  $\beta : \mathbb{R}^+ \times \mathbb{R}^+ \rightarrow \mathbb{R}^+$  se dice que pertenece a la clase  $KL$ , si para cada  $s$  fijo el mapeo  $\beta(r, s)$  pertenece a la clase  $K_\infty$  con respecto a  $r$ , y para cada  $r$  fijo el mapeo  $\beta(r, s)$  es decreciente con respecto a  $s$  y  $\beta(r, s) \rightarrow 0$  cuando  $s \rightarrow \infty$ .

Considérese el sistema variante en el tiempo

$$\dot{x} = f(x, t) \tag{2.30}$$

donde  $f : \mathbb{R}^n \times \mathbb{R}^+ \rightarrow \mathbb{R}^n$  es localmente Lipschitz en  $x$  y continua a tramos en  $t$ .

**Definición 5** El origen  $x = 0$  es un punto de equilibrio de (2.30) si [12]

$$f(0, t) = 0, \forall t \geq 0. \tag{2.31}$$

**Lema 6** El punto de equilibrio de (2.30) es [12]

- *Global Uniformemente Estable en el sentido de Lyapunov*, si existe una función  $\gamma(\cdot)$  clase  $K_\infty$  tal que

$$|x(t)| \leq \gamma(|x(t_0)|), \forall t \geq t_0 \geq 0, \forall x(t_0) \in \mathbb{R}^n; \tag{2.32}$$

- *Global Uniforme Asintóticamente Estable en el sentido de Lyapunov*, si existe una función  $\beta(\cdot, \cdot)$  clase  $KL$  tal que

$$|x(t)| \leq \beta(|x(t_0)|, t - t_0), \forall t \geq t_0 \geq 0, \forall x(t_0) \in \mathbb{R}^n; \tag{2.33}$$

- *Global Exponencialmente Estable en el sentido de Lyapunov*, si (2.33) se satisface con

$$\beta(r, s) = kre^{-\alpha s}, k > 0, \alpha > 0.$$

Considérese ahora una clase general de sistemas afines en el control  $u \in \mathbb{R}^m$  e invariantes en el tiempo

$$\dot{x} = f(x) + g(x)u \tag{2.34}$$

Para estos sistemas, se establece

(2.34) es estable entrada-estado (ISS de "Input to State Stability") si existen funciones  $\beta \in KL$  y  $\gamma \in K$  tales que, para cada condición inicial  $x_0$  y para cada entrada acotada  $u \in L_\infty$  ( $\|u\|_{L_\infty} = \sup_{t \geq 0} \|u(t)\|_2 = \sup_{t \geq 0} \|u(t)\|_\infty$ ), cada solución  $x(t)$ , que empieza en  $t = 0$ , satisface [15]

$$\|x(t)\| \leq \beta(\|x(t_0)\|, t) + \gamma(\|u\|_\infty), \forall t \geq 0 \quad (2.35)$$

Si se considera que (2.34) recibe la acción de una perturbación externa ( $d$ ), tal que se pueda reformular como

$$\dot{x} = f(x) + g_1(x)d + g_2(x)u \quad (2.36)$$

entonces se puede establecer

**Lema 7** (2.36) es estabilizable entrada estado si existe una ley  $u = \alpha(x)$  continua, con  $\alpha(0) = 0$ , tal que el sistema en lazo cerrado es ISS [15].

**Definición 8** Funciones Lyapunov de Control [15]

Consideremos la derivada de una función candidata de Lyapunov, para el sistema (2.34)

$$\begin{aligned} \dot{V} &= \frac{\partial V}{\partial x} f(x) + \frac{\partial V}{\partial x} g(x)u \\ &\triangleq L_f V + L_g V u \end{aligned} \quad (2.37)$$

El objetivo es encontrar una ley de control  $u = \alpha(x)$  que garantice que  $\dot{V}$  sea definida negativa. En general, la función de Lyapunov no es conocida, así que el investigador tiene como objetivo obtener una función de Lyapunov y una ley de control adecuadas.

**Definición 9** Una función suave definida positiva radialmente no acotada ( $V(x) \rightarrow \infty$  cuando  $\|x\| \rightarrow \infty$ )  $V(x)$  es llamada una función Lyapunov de control ("Control Lyapunov Function" clf) si [15]

$$\inf_{u \in \mathbb{R}^m} \{L_f V + L_g V u\} < 0, \forall x \neq 0. \quad (2.38)$$

**Lema 10** Una función suave definida positiva radialmente no acotada  $V$  es una clf si y sólo si  $L_g V = 0 \Rightarrow L_f V < 0, \forall x \neq 0$  [15].

**Lema 11** Una clf  $V(x)$  satisface la propiedad del control pequeño ("small control property" scp) si existe una ley de control  $\alpha_c(x)$  continua en  $\mathbb{R}^n$  tal que [15]

$$L_f V(x) + L_g V(x)\alpha_c(x) < 0, \forall x \neq 0. \quad (2.39)$$

**Lema 12** (Sontag) El sistema (2.36) es estabilizable por un control de retroalimentación continuo en el origen y suave fuera de él si y sólo si existe una clf con la propiedad del control pequeño [15].

**Teorema 13** *El problema de asignación de ganancia óptima inversa, para el sistema (2.36), tiene solución si existen: a) una función  $R(x)$  valuada matricialmente tal que  $R(x) = R(x)^\top > 0 \forall x$ , b) funciones  $l(x)$ ,  $E(x)$  definidas positivas y radialmente no acotadas, y c) una ley de control  $u = \alpha(x)$  continua en todas partes con  $\alpha(0) = 0$ , la cual minimiza el funcional de costo*

$$J(u) = \lim_{t \rightarrow \infty} \left\{ E(x) + \int_0^t (l(x) + u^\top R(x)u) d\tau \right\} \quad (2.40)$$

dada por

$$u = -\beta R(x)^{-1} (L_g V)^\top, \quad (2.41)$$

y  $l(x)$  definida como sigue

$$l(x) = -2\beta L_f V + \beta^2 (L_g V) R(x)^{-1} (L_g V)^\top, \text{ con } \beta > 0 \quad (2.42)$$

*El problema de asignación de ganancia óptima inversa, para el sistema (2.36), tiene solución si y sólo si el sistema es entrada estado estabilizable [15].*

Consideremos el sistema no lineal variante en el tiempo

$$\dot{x} = f(x, t) + g_1(x, t)d \quad (2.43)$$

donde  $x \in \mathbb{R}^n$  es el estado,  $d \in \mathbb{R}^r$  es una perturbación externa, y  $f(0, t) = 0$ .

**Teorema 14** *El sistema (2.43) se dice ser entrada-estado estable (ISS) si existe una función  $\beta \in KL$  y una función  $\chi \in K$  tal que, para cualquier  $x(t_0)$  y para cualquier perturbación continua  $d$  en  $[0, \infty)$ , su solución existe para todo  $t \geq 0$  y satisface [15]:*

$$|x(t)| \leq \beta(x(t_0), t - t_0) + \chi\left(\sup_{t_0 \leq \tau \leq t} |d(\tau)|\right) \quad (2.44)$$

$$\forall t_0 \text{ y } t \text{ tal que } 0 \leq t_0 \leq t.$$

**Teorema 15** (Sontag) *Supóngase que para el sistema (2.43) existe una función  $V : \mathbb{R}^n \times \mathbb{R}^+ \rightarrow \mathbb{R}^+$ , tal que para todo  $x \in \mathbb{R}^n$  y  $d \in \mathbb{R}^m$ ,*

$$\gamma_1(|x|) \leq V(x, t) \leq \gamma_2(|x|)$$

$$x \geq \rho(|d|) \Rightarrow \frac{\partial V}{\partial t} + \frac{\partial V}{\partial x} f(x, t) + \frac{\partial V}{\partial x} g_1(x, t)d \leq \gamma_3(|x|) \quad (2.45)$$

donde  $\gamma_1, \gamma_2$  y  $\rho$  son funciones clase  $K_\infty$  y  $\gamma_3$  es clase  $K$ . Entonces el sistema (2.43) es ISS con  $\chi = \gamma_1^{-1} \circ \gamma_2 \circ \rho$  [15].

**Teorema 16** *Una función  $V$ , que satisface las condiciones del teorema anterior, es llamada función ISS Lyapunov, el inverso de este teorema también es verdadero, esto es, la propiedad ISS implica la existencia de una función ISS-Lyapunov [15]*

Considérese ahora el sistema (2.43), el cual, además de una entrada de perturbación, tiene una acción de control  $u$

$$\begin{aligned} \dot{x} &= f(x) + g_1(x)d + g_2(x)u \\ u &\in \mathbb{R}^m \text{ y } f(0) = 0 \end{aligned} \quad (2.46)$$

**Definición 17** Se dice que el sistema (2.43) es entrada estado estabilizable, si existe una ley de control  $u = \alpha(x)$  continua en todas partes, con  $\alpha(0) = 0$ , tal que el sistema en lazo cerrado es ISS con respecto a  $d$  [15].

**Definición 18** Una función suave definida positiva y radialmente no acotada  $V : \mathbb{R}^n \rightarrow \mathbb{R}^+$  es llamada una función Lyapunov de control ISS “iss-clf”, para el sistema (2.43), si existe una función  $\rho \in K_\infty$  tal que la siguiente implicación se satisface  $\forall x \neq 0$  y  $\forall d \in \mathbb{R}^r$  [15] :

$$x \geq \rho(|d|) \Rightarrow \inf_{u \in \mathbb{R}^m} \{L_f V + L_{g_1} V d + L_{g_2} V u\} < 0. \quad (2.47)$$

**Definición 19** [15] Problema de seguimiento adaptable de trayectorias. Considérese el sistema

$$\begin{aligned} \dot{x} &= f(x) + F(x)\theta + u \\ y &= h(x) \end{aligned} \quad (2.48)$$

donde  $x \in \mathbb{R}^n, u \in \mathbb{R}^n, y \in \mathbb{R}$  y los mapeos  $f(x), F(x), h(x)$  son suaves y  $\theta$  es un parámetro constante desconocido el cual puede tomar cualquier valor en  $\mathbb{R}^p$ . La variable  $y$ , es la salida del sistema. Se desea que ésta siga a una trayectoria de referencia [15].

**Proposición 20** Para una función suave dada  $y_r(t)$ , existen funciones suaves  $\rho(t, \theta)$  y  $\alpha_r(t, \theta)$  tales que [15]

$$\begin{aligned} \frac{d\rho(t, \theta)}{dt} &= f(\rho(t, \theta)) + F\rho(t, \theta)\theta + \alpha_r(t, \theta) \\ y_r(t) &= h(\rho(t, \theta)) \end{aligned} \quad (2.49)$$

Considérese la señal  $x_r(t) = \rho(t, \hat{\theta})$  la cual está gobernada por,

$$\dot{x}_r = \frac{\partial \rho(t, \hat{\theta})}{\partial t} + \frac{\partial \rho(t, \hat{\theta})}{\partial \hat{\theta}} \dot{\hat{\theta}} = f(x_r) + F(x_r)\hat{\theta} + \alpha_r(t, \hat{\theta}) + \frac{\partial \rho(t, \hat{\theta})}{\partial \hat{\theta}} \dot{\hat{\theta}} \quad (2.50)$$

defínase el error de seguimiento  $e = x - x_r = x - \rho(t, \hat{\theta})$  y calcúlese su derivada,

$$\dot{e} = \tilde{f} + \tilde{F}\theta + F_r\tilde{\theta} - \frac{\partial \rho(t, \hat{\theta})}{\partial \hat{\theta}} \dot{\hat{\theta}} + \tilde{u} \quad (2.51)$$

$$\text{donde } \tilde{\theta} = \theta - \hat{\theta}$$

$$\tilde{f}(t, e, \hat{\theta}) = f(x) - f(x_r) + \alpha_r(t, \hat{\theta})$$

$$\tilde{F}(t, e, \hat{\theta}) = F(x) - F(x_r)$$

$$F_r(t, \hat{\theta}) = F(x_r)$$

$$\tilde{u} = u - \alpha_r(t, \hat{\theta})$$



**Definición 21** [15] *El problema de seguimiento adaptable para el sistema (2.48) tiene solución si las condiciones de la Proposición 20 se satisfacen y si existen funciones continuas  $\tilde{\alpha}(t, e, \hat{\theta})$  con  $\tilde{\alpha}(t, 0, \hat{\theta}) \equiv 0$ , una función continua  $\tau(t, e, \hat{\theta})$  y una matriz simétrica definida positiva  $\Gamma$  tal que el controlador dinámico*

$$\begin{aligned}\tilde{u} &= \tilde{\alpha}(t, e, \hat{\theta}) \\ \dot{\hat{\theta}} &= \Gamma\tau(t, e, \hat{\theta})\end{aligned}\quad (2.52)$$

*garantiza que el punto de equilibrio  $e = 0, \tilde{\theta} = 0$  del sistema (2.51) es globalmente estable y  $e(t) \rightarrow 0$  cuando  $t \rightarrow \infty$  para cualquier valor del parámetro desconocido  $\theta$  [15].*

**Definición 22** [15] *Una función suave  $V_a : R^+ \times R^n \times R^p \rightarrow R^+$  definida positiva, decreciente y radialmente no acotada en  $e$  (uniformemente en  $t$ ) para todo  $\theta$  es llamada una función de control adaptable de seguimiento de Lyapunov (atclf de "Adaptive Tracking Control Lyapunov Function") para (2.48), o alternativamente una función de control de Lyapunov (aclf de Adaptive Control Lyapunov Function) para (2.51), si se satisfacen las condiciones de la Proposición 20 y existe una matriz simétrica definida positiva  $\Gamma \in R^{p \times p}$  tal que para todo  $\theta \in R^p, V_a$  es una clf para el sistema modificado no adaptable [15]*

$$\dot{e} = \tilde{f} + \tilde{F}\theta + F\Gamma \left( \frac{\partial V_a}{\partial \theta} \right)^T - \frac{\partial \rho}{\partial \theta} \Gamma \left( \frac{\partial V_a}{\partial \theta} F \right)^T + \tilde{u} \quad (2.53)$$

*esto es,  $V_a$  satisface,*

$$\inf_{\tilde{u}} \left\{ \frac{\partial V_a}{\partial t} + \frac{\partial V_a}{\partial e} \left( \tilde{f} + \tilde{F}\theta + F\Gamma \left( \frac{\partial V_a}{\partial \theta} \right)^T - \frac{\partial \rho}{\partial \theta} \Gamma \left( \frac{\partial V_a}{\partial \theta} F \right)^T + \tilde{u} \right) \right\} < 0 \quad (2.54)$$

**Definición 23** *El problema de seguimiento adaptable óptimo inverso para el sistema (2.51) tiene solución si existe una constante positiva  $\beta$ , una función definida positiva  $r(t, e, \hat{\theta})$ , una función  $l(t, e, \hat{\theta})$  definida positiva en  $e$ , para todo  $\theta$ , y una acción dinámica de control (2.52) la cual resuelve el problema de seguimiento adaptable cuadrático y además minimiza el funcional de costo*

$$J = \beta \lim_{t \rightarrow \infty} |\theta - \hat{\theta}(t)|^2 + \int_0^\infty (l(t, e, \hat{\theta}) + r(t, e, \hat{\theta})\tilde{u}^2) dt \quad (2.55)$$

*para todo  $\theta \in R^p$*

Esta definición de optimización crea restricciones en  $e$  y en  $\tilde{u}$ , así como en el valor final de  $|\theta - \hat{\theta}(t)|$ . Aún cuando no se garantiza que  $\tilde{\theta}$  tienda a un límite en el caso general de seguimiento, la existencia del límite de  $|\theta - \hat{\theta}(t)|^2$  se garantiza gracias a la suposición que el problema de seguimiento adaptable cuadrático tiene solución. No se tiene restricción sobre  $\tilde{\theta}$  en la integral, ya que en general en sistemas adaptables con retroalimentación no se garantiza convergencia al valor real de  $\theta$  [15].

**Teorema 24** *Supóngase que existe una atclf  $V_a(t, e, \hat{\theta})$  para el sistema (2.51) y una ley de control  $\tilde{u} = \tilde{\alpha}(t, e, \hat{\theta})$  que estabiliza el sistema [15]*

$$\dot{e} = \tilde{f} + \tilde{F}\theta + F\Gamma \left( \frac{\partial V_a}{\partial \theta} \right)^T - \frac{\partial \rho}{\partial \theta} \Gamma \left( \frac{\partial V_a}{\partial \theta} F \right)^T + \tilde{u} \quad (2.56)$$

y tiene la forma

$$\tilde{\alpha}(t, e, \hat{\theta}) = -r^{-1}(t, e, \hat{\theta}) \frac{\partial V_a}{\partial e} \quad (2.57)$$

donde  $r(t, e, \hat{\theta}) > 0$  para todo  $t, e, \theta$ . Entonces

1. La ley de control no adaptable

$$\tilde{u} = \tilde{\alpha}^*(t, e, \hat{\theta}) = \beta \tilde{\alpha}(t, e, \hat{\theta}) \quad \beta > 2 \quad (2.58)$$

minimiza el funcional de costo

$$J_a = \int_0^\infty \left( l(t, e, \hat{\theta}) + r(t, e, \hat{\theta}) \tilde{u}^2 \right) dt \quad (2.59)$$

a lo largo de las soluciones del sistema no adaptable (2.51) donde

$$l(t, e, \hat{\theta}) = -2\beta \left[ \frac{\partial V_a}{\partial t} + \frac{\partial V_a}{\partial t} (\tilde{f} + \tilde{\alpha}) \right] + \beta(\beta - 2) r^{-1} \left( \frac{\partial V_a}{\partial e} \right)^2 \quad (2.60)$$

2. El problema de seguimiento adaptable óptimo inverso tiene solución.

## 2.3 Redes Neuronales dinámicas de alto orden.

### 2.3.1 Redes neuronales en control.

Los conceptos presentados en esta sección son tomados de [14] y [23]

Las redes neuronales artificiales se han convertido en una herramienta útil en aplicaciones de ingeniería, gracias a su aplicabilidad en el modelado, estimación y control de sistemas dinámicos complejos a diferencia de la teoría convencional del control que logra esto sólo para sistemas lineales. Usando las redes neuronales, se pueden obtener algoritmos de control que son robustos al ruido, complejidad e incertidumbres en la planta.

En el presente trabajo se utilizan las redes neuronales dinámicas (ó recurrentes) las cuales a diferencia de las redes estáticas se modelan usando ecuaciones diferenciales. Las redes utilizadas están basadas en el modelo dinámico de Hopfield. Estas son consideradas como buenos candidatos para operaciones satisfactorias de sistemas no lineales que tienen que tratar con una gran cantidad de incertidumbres. Las redes neuronales son atractivas para propósitos de control en parte por su facilidad de implementación, estructura relativamente simple, robustez y su capacidad de cambiar sus parámetros en línea. Dentro de esta clasificación se encuentran las redes neuronales de alto orden (ó RHONN de "Recurrent High Order Neural Nets"). Las RHONN usualmente presentan una gran rapidez de adaptación y permiten que el proceso se remodele rápidamente, así como tratar efectivamente parámetros inciertos.

En [14], las redes neuronales recurrentes de alto orden (RHONN) se definen como

$$\dot{x}_i = -a_i x_i + \sum_{k=1}^L w_{ik} \prod_{j \in I_k} y_j^{d_j(k)} \quad i = 1, \dots, n \quad (2.61)$$

donde  $x_i$  es el  $i$ -ésimo estado de la neurona,  $L$  es el número de conexiones de alto orden,  $\{I_1, I_2, \dots, I_L\}$  es una colección de subconjuntos no ordenados de  $\{1, 2, \dots, m+n\}$ ,  $a_i > 0$ ,  $w_{ik}$  son los pesos ajustables de la red neuronal,  $d_j(k)$  son enteros no negativos, y  $y$  es un vector que se define como:

$$y = [y_1, \dots, y_n, y_{n+1}, \dots, y_{n+m}]^T = [S(x_1), \dots, S(x_n), S(u_1), \dots, S(u_m)]^T$$

donde  $u = [u_1, u_2, \dots, u_m]^T$  es la entrada a la red neuronal, y  $S(\cdot)$  una función sigmoide suave dada por  $S(x) = \frac{1}{1+\exp(-\beta x)} + \varepsilon$ . Para la sigmoide,  $\beta$  es una constante positiva y  $\varepsilon$  es un número real positivo. Luego,  $S(x) \in [\varepsilon, \varepsilon + 1]$ .

Como se puede ver, (2.61) permite la introducción de términos de alto orden.

Definiendo un vector  $z(x, u) = [z_1(x, u), \dots, z_L(x, u)]^T = \left[ \prod_{j \in I_1} y_j^{d_j(1)}, \dots, \prod_{j \in I_L} y_j^{d_j(L)} \right]^T$  (2.61) puede reescribirse como

$$\dot{x}_i = -a_i x_i + \sum_{k=1}^L w_{ik} z_k(x, u), \quad i = 1, \dots, n$$

6

$$\dot{x}_i = -a_i x_i + w_i^T z_i(x, u), \quad i = 1, \dots, n \quad (2.62)$$

donde  $w_i = [w_{i,1} \dots w_{i,L}]^T$

En el tercer capítulo de esta memoria, se consideran los términos del tipo

$$y = [y_1, \dots, y_n, y_{n+1}, \dots, y_{n+m}]^T = [S(x_1), \dots, S(x_n), u_1, \dots, u_m]^T$$

( esto significa que se utiliza el mismo número de entradas que de estados). En el quinto capítulo se extienden estos resultados a menos entradas que estados. También se supone que la entrada afecta a la red neuronal directamente. Luego (2.61) puede escribirse como

$$\dot{x}_i = -a_i x_i + w_i^T z_i(x) + u_i, \quad i = 1, \dots, n \quad (2.63)$$

Reescribiendo (2.62) en forma matricial se obtiene

$$\dot{x} = Ax + Wz(x) + u \quad (2.64)$$

donde  $x \in \mathbb{R}^n$ ,  $W \in \mathbb{R}^{n \times L}$ ,  $z(x) \in \mathbb{R}^L$ ,  $u \in \mathbb{R}^n$ , y  $A = -\lambda I$ ,  $\lambda > 0$ .

En la presente memoria, se considera la siguiente modificación a la red de alto orden

$$\dot{x} = Ax + W\Gamma z(x) + u \quad (2.65)$$

donde  $\Gamma \in \mathfrak{R}^{L \times L}$  es una matriz diagonal definida como

$$\Gamma = \begin{pmatrix} \gamma_1 & 0 & & 0 \\ 0 & \gamma_2 & & 0 \\ & & & \gamma_{L-1} & 0 \\ 0 & 0 & & 0 & \gamma_L \end{pmatrix}$$

$\gamma_i > 0, \quad i = 1 \dots L$

Estos parámetros  $\gamma$  juegan el mismo papel que el tamaño del paso en el algoritmo del gradiente descendiente.



# Capítulo 3

## Control neuronal para seguimiento de trayectorias.

*En este capítulo se presentan dos métodos para obtener leyes de control para seguimiento de trayectorias. El primer método denominado de estabilización cuadrática utiliza una red neuronal como identificador que modelará la planta desconocida. Posteriormente usando funciones cuadráticas adaptables de Lyapunov se obtendrá una ley de aprendizaje para la red neuronal tal que minimize el error de modelado y se estabilizará la dinámica del error de seguimiento. Con esto se obtendrá una ley de control que asegurará que tanto la red neuronal como la planta seguirán la trayectoria de referencia.*

*El segundo método a presentar es el control óptimo inverso. Este método deriva una acción de control la cual optimiza a un funcional de costo. La derivación de dicha acción de control se realiza mediante funciones adaptables de Lyapunov.*

### 3.1 Introducción.

Muchas de las aplicaciones en control tratan con procesos que presentan altas no linealidades e incertidumbres en el modelado, así como una baja tolerancia a perturbaciones; si dichas perturbaciones no son tomadas en cuenta en el diseño del controlador, pueden dar lugar a un pobre desempeño del sistema en lazo cerrado.

Bajo la suposición de que todos los estados están disponibles para medición, se presentan dos técnicas basadas en control neuronal adaptable para seguimiento de trayectorias en sistemas no lineales. Ambos métodos parten de la necesidad de controlar una planta no lineal desconocida; se utiliza una red neuronal dinámica para construir un modelo en línea de la planta y se desarrollan dos métodos para obtener una acción de control que asegure el seguimiento de una trayectoria de referencia por la red neuronal y la planta desconocida.

En el primer método, de estabilización cuadrática, se desarrolla un control no lineal robusto por medio de funciones cuadráticas de Lyapunov basándonos en los conceptos estabilidad entrada-estado presentados en el segundo capítulo. Sin embargo, en general la ley de control obtenida no es óptima con respecto a un funcional de costo. El segundo método a presentar es el control óptimo inverso motivado por el control no lineal  $H_\infty$ . El control óptimo inverso calcula una ley de control, la cual junto con las restricciones apropiadas, resulta ser óptima con respecto a un funcional de costo bien definido. La obtención de dicha acción de control

se logra mediante funciones adaptables de control de Lyapunov.

En ambos métodos la función adaptable de Lyapunov nos lleva a una ley de adaptación de pesos para la red neuronal, los cuales alcanzan un valor que minimiza el error de modelado. Los nuevos métodos desarrollados se ilustrarán en el siguiente capítulo aplicándolos al control de sistemas caóticos tanto en su estabilización como en el seguimiento de trayectorias.

## 3.2 Método de estabilización cuadrática.

Nuestro objetivo es obtener un control robusto para seguimiento de trayectorias mediante funciones de Lyapunov adaptables que estabilizen asintóticamente el error de seguimiento entre la planta, la cual se modelará mediante una red neuronal dinámica, y una trayectoria de referencia. En esta sección presentamos el uso de funciones cuadráticas de Lyapunov.

### 3.2.1 Modelado de la planta

Sea la planta no lineal y desconocida modelada por,

$$\dot{x}_p = F_p(x_p, u) \triangleq f_p(x_p) + g_p(x_p)u \quad (3.1)$$

donde  $x_p, f_p \in \mathbb{R}^n$ ,  $g_p \in \mathbb{R}^{n \times n}$ ,  $u \in \mathbb{R}^n$ .

Como se mencionó los campos  $f_p$  y  $g_p$  son desconocidos. Nuestro propósito es modelar (3.1) por medio de una red neuronal dinámica de la forma (2.65). Por lo tanto, el modelo de la planta toma la siguiente forma

$$\dot{x}_p = Ax + W^*\Gamma z(x) + w_{per} + u \quad (3.2)$$

donde el término  $w_{per}$  representa el error de modelado, y  $W^*$  son los pesos constantes pero desconocidos de la red neuronal, los cuales minimizan el error de modelado. Una forma natural de definir el error de modelado es  $w_{per} = x - x_p$ .

### 3.2.2 Análisis de seguimiento de trayectorias

El error de seguimiento de trayectoria se define como la diferencia entre la salida de la planta desconocida modelada por (3.2) y la trayectoria de referencia dada por

$$\dot{x}_r = f_r(x_r, u_r), \quad x_r \in \mathbb{R}^n \quad (3.3)$$

Se define el error de seguimiento como

$$e := x_p - x_r \quad (3.4)$$

Su derivada con respecto al tiempo es

$$\dot{e} = \dot{x}_p - \dot{x}_r = Ax + W^*\Gamma z(x) + w_{per} + u - f_r(x_r, u_r) \quad (3.5)$$

Ahora, sumando y restando al lado derecho de (3.5) los términos  $\hat{W}\Gamma z(x_r)$ ,  $\alpha_r(t, \hat{W})$ ,  $Ae$ , y recordando que  $w_{per} = x - x_p$ , se obtiene

$$\begin{aligned}
 \dot{e} &= Ax + W^*z(x) + x - x_p + u - f_r(x_r, u_r) \pm \hat{W}\Gamma z(x_r) \pm \alpha_r(t, \hat{W}) \pm Ae \\
 &= Ae + W^*\Gamma z(x) + u + \left( -f_r(x_r, u_r) + \hat{W}\Gamma z(x_r) + x - x_p + Ax - Ae + \alpha_r(t, \hat{W}) \right) \\
 &\quad - \hat{W}\Gamma z(x_r) - \alpha_r(t, \hat{W}) \tag{3.6} \\
 &= Ae + W^*\Gamma z(x) + u + \left( -f_r(x_r, u_r) + Ax_r + \hat{W}\Gamma z(x_r) + (A + I)(x - x_p) + \alpha_r(t, \hat{W}) \right) \\
 &\quad - \hat{W}\Gamma z(x_r) - \alpha_r(t, \hat{W})
 \end{aligned}$$

donde  $\hat{W}$  es un estimado de  $W^*$ .

Por la proposición 20 del capítulo 2 podemos suponer que existen funciones suaves  $x_r$  y  $\alpha_r(t, \hat{W})$  tales que  $-f_r(x_r, u_r) + Ax_r + \hat{W}\Gamma z(x_r) + (A + I)(x - x_p) + \alpha_r(t, \hat{W})$  es igual a cero; es decir, que existe un  $u = \alpha_r(t, \hat{W})$  tal que  $x_r$  es solución de la ecuación (3.2). Por lo tanto se tiene,

$$Ax_r + \hat{W}\Gamma z(x_r) + (A + I)(x - x_p) + \alpha_r(t, \hat{W}) = f_r(x_r, u_r) \tag{3.7}$$

y (3.6) se reduce a

$$\dot{e} = Ae + W^*\Gamma z(x) + u - \hat{W}\Gamma z(x_r) - \alpha_r(t, \hat{W}) \tag{3.8}$$

Sumando y restando a (3.8), el término  $\hat{W}\Gamma z(x)$  se tiene,

$$\dot{e} = Ae + \left( W^* - \hat{W} \right) \Gamma z(x) + \hat{W}\Gamma (z(x) - z(x_r)) + u - \alpha_r(t, \hat{W}) \tag{3.9}$$

Definiendo

$$\begin{aligned}
 \tilde{W} &= W^* - \hat{W}, & \tilde{u} &= u - \alpha_r(t, \hat{W}), \\
 \phi_z(x, x_r) &= (z(x) - z(x_r))
 \end{aligned} \tag{3.10}$$

y sustituyendo (3.10) en (3.9), se obtiene

$$\dot{e} = Ae + \tilde{W}\Gamma z(x) + \hat{W}\Gamma \phi_z(x, x_r) + \tilde{u} \tag{3.11}$$

Finalmente el problema de seguimiento de trayectorias se resuelve mediante la estabilización del error de seguimiento, cuya dinámica está dada por (3.11).

### 3.2.3 Estabilización del error de seguimiento de trayectorias.

Una vez que se obtuvo (3.11), se puede proceder a estudiar su estabilización, para lo cual en esta tesis se usan funciones cuadráticas de Lyapunov. Con estas funciones se deriva una ley de adaptación de pesos para la red neuronal y una ley de control tal que el sistema (3.11) en lazo cerrado sea globalmente asintóticamente estable.

Si  $\tilde{u} = 0$ , entonces un estado de equilibrio de la dinámica del error (3.11) es  $e = 0$ ,  $\tilde{W} = 0$ , lo que es cierto de como están definidos  $Ae$ ,  $z(x)$  y  $\phi_z(x, x_r)$ .



Ahora se procede a estabilizar a (3.11) globalmente asintóticamente en este punto de equilibrio.

Con este propósito, se selecciona una función candidata Lyapunov cuadrática adaptable

$$V = \frac{1}{2}\|e\|^2 + \frac{1}{2}tr \left\{ \tilde{W}^T \tilde{W} \right\} \quad (3.12)$$

Donde  $V$  es una función definida positiva. Derivando la ecuación anterior a lo largo de las trayectorias de (3.11), se obtiene lo siguiente:

$$\dot{V} = e^T A e + e^T \tilde{W} \Gamma z(x) + e^T \hat{W} \Gamma (z(x) - z(x_r)) + e^T \tilde{u} + tr \left\{ \dot{\tilde{W}}^T \tilde{W} \right\} \quad (3.13)$$

Como en [23], usamos la siguiente ley de aprendizaje:

$$tr \left\{ \dot{\tilde{W}}^T \tilde{W} \right\} = -e^T \tilde{W} \Gamma z(x) \quad (3.14)$$

la cual se puede escribir elemento a elemento como,

$$\dot{\hat{w}}_{i,j} = e_i (\gamma_j z(x_j)), \quad i = 1, 2, \dots, n, \quad j = 1, 2, \dots, L. \quad (3.15)$$

Sustituyendo (3.15) en (3.13) se obtiene,

$$\dot{V} = -\lambda e^T e + e^T \hat{W} \Gamma (z(x) - z(x_r)) + e^T \tilde{u} \quad (3.16)$$

Considerando (3.16), se propone la siguiente ley de control:

$$\tilde{u} = -\hat{W} \Gamma (z(x) - z(x_r)) - \mu e \quad \mu \in R^+ \quad (3.17)$$

De donde,

$$\dot{V} = -(\lambda + \mu) \|e\|^2 < 0, \quad \forall e \neq 0 \text{ y } \forall \hat{W} \neq 0 \quad (3.18)$$

Si  $t \rightarrow \infty$ , los valores de los pesos estimados de la red neuronal se estabilizan en valores constantes; aún cuando en control adaptable no es posible demostrar que los pesos estimados aproximan exactamente los valores de los pesos desconocidos, se puede demostrar que el error de estimación de pesos se mantiene acotado y en un valor constante; de donde si  $t \rightarrow \infty$ ,  $V \rightarrow R^+$ ; por el Lema de Barbalat [18],  $\lim_{t \rightarrow \infty} e(t) \rightarrow 0$ . El error de seguimiento de trayectorias es llevado asintóticamente al origen y el error de estimación de pesos se mantiene en un valor constante cuando  $t \rightarrow \infty$ .

Finalmente, se obtiene una acción de control  $u$ , la cual actúa tanto en la red neuronal como en la planta, dada por:

$$\begin{aligned} u &= \tilde{u} + \alpha_r(t, \hat{W}) \\ u &= -\hat{W} \Gamma (z(x) - z(x_r)) - \mu e + f_r(x_r, u_r) - A x_r - \hat{W} \Gamma z(x_r) - (A + I)(x - x_r) \end{aligned} \quad (3.19)$$

Los resultados obtenidos se pueden resumir en el siguiente

**Teorema 25** *Para el sistema no lineal desconocido modelado por (3.2), la ley de aprendizaje en línea (3.15) y la ley de control (3.19) en conjunto aseguran el seguimiento de la referencia no lineal (3.3).*



### 3.3 Método del control óptimo inverso.

En nuestro problema de control el objetivo no es sólo lograr el seguimiento de trayectorias, sino también obtener una acción de control óptima con respecto a un funcional de costo bien definido.

El control óptimo inverso se basa en el concepto en el cual la estabilización entrada-estado es equivalente a la resolución del problema de control no lineal  $H_\infty$  [15]. Para esto, se hace uso de funciones de control de Lyapunov robustas; este interés en el método inverso por encima del directo se debe a que no todo problema de optimización no lineal directa tiene solución y también a que la optimización inversa garantiza que el controlador permanece estabilizando entrada-estado en presencia de una cierta clase de dinámicas no modeladas las cuales no tienen que ser pequeñas desde el punto de vista  $H_\infty$ , no tienen que ser lineales o estabilizables entrada-estado [29].

La idea principal del método de optimización inversa es el obtener una acción de control robusta junto con las respectivas restricciones que hacen un funcional de costo bien definido y significativo. Este método proporciona una forma adecuada de obtener controladores robustos donde la robustez en lazo cerrado obtenida como resultado de la optimalidad del controlador es independiente de la elección del funcional de costo [29].

#### 3.3.1 Modelado de la planta

Consideremos de nuevo la planta no lineal desconocida modelada por (3.1), y que toma la forma dada por (3.2). Procedemos a formular una nueva ley de control la cual optimizará un funcional de costo relacionado con la solución del sistema de Hamilton- Jacobi-Bellman.

#### 3.3.2 Análisis de seguimiento de trayectorias

De nuevo procedemos a analizar el error de seguimiento de la planta desconocida (3.2) con respecto a la trayectoria de referencia dada por (3.3). Definimos el error de seguimiento por (3.4) y cuya derivada con respecto al tiempo está dada por (3.5). Ahora, sumando y restando al lado derecho de (3.5) los términos  $\hat{W}\Gamma z(x_r)$ ,  $\alpha_r(t, \hat{W})$ ,  $Ae$ ,  $Ax_r$  y  $x_r$  y simplificando se tiene

$$\begin{aligned} \dot{e} = & Ae + W^*\Gamma z(x) + u + \left( -f_r(x_r, u_r) + Ax_r + \hat{W}\Gamma z(x_r) + x_r - x_p + \alpha_r(t, \hat{W}) \right) \\ & - Ae - \hat{W}\Gamma z(x_r) - \alpha_r(t, \hat{W}) - Ax_r - x_r + x + Ax \end{aligned} \quad (3.20)$$

De nuevo basándonos en la proposición 20, la red neuronal seguirá la señal de referencia dada aún con la presencia de perturbaciones (error de modelado) si los términos que están entre parentesis en la ecuación anterior su suma es igual a cero, esto es

$$Ax_r + \hat{W}\Gamma z(x_r) + x_r - x_p + \alpha_r(t, \hat{W}) = f_r(x_r, u_r) \quad (3.21)$$

donde  $\hat{W}$  es un estimado de  $W^*$  A partir de lo anterior podemos reducir (3.20) a

$$\dot{e} = Ae + W^*\Gamma z(x) - \hat{W}\Gamma z(x_r) - Ae + (A + I)(x - x_r) + u - \alpha_r(t, \hat{W}) \quad (3.22)$$

Ahora, sumando y restando a (3.22) el término  $\hat{W}\Gamma z(x)$  obtenemos

$$\dot{e} = Ae + (W^* - \hat{W})\Gamma z(x) + \hat{W}\Gamma(z(x) - z(x_r)) + (A + I)(x - x_r) - Ae + u - \alpha_r(t, \hat{W}) \quad (3.23)$$

Definamos de nuevo,

$$\tilde{W} = W^* - \hat{W}. \quad \tilde{u} = u - \alpha_r(t, \hat{W}), \quad (3.24)$$

y sustituyendo (3.24) en (3.23), obtenemos

$$\begin{aligned} \dot{e} &= Ae + \tilde{W}\Gamma z(x) + \hat{W}\Gamma(z(x) - z(x_r)) + (A + I)(x - x_r) - Ae + \tilde{u} \\ &= Ae + \tilde{W}\Gamma z(x) + \hat{W}\Gamma(z(x) - z(x_p) + z(x_p) - z(x_r)) \\ &\quad + (A + I)(x - x_p + x_p - x_r) - Ae + \tilde{u} \end{aligned} \quad (3.25)$$

Sea ahora

$$\tilde{u} = u_1 + u_2 \quad (3.26)$$

Tomando

$$u_1 = -\hat{W}\Gamma(z(x) - z(x_p)) - (A + I)(x - x_p) \quad (3.27)$$

entonces (3.25) se reduce a

$$\dot{e} = Ae + \tilde{W}\Gamma z(x) + \hat{W}\Gamma z(x_p) - z(x_r) + (A + I)(x_p - x_r) - Ae + u_2$$

Teniendo en cuenta que  $e = x_p - x_r$ , la ecuación anterior se reduce a

$$\dot{e} = (A + I)e + \tilde{W}\Gamma z(x) + \hat{W}\Gamma z(x_p) - z(x_r) + u_2 \quad (3.28)$$

Entonces el problema consiste en obtener una acción de control  $u_2$ , la cual estabilice al sistema (3.28). Dicha ley de control se obtendrá usando la metodología del control óptimo inverso

### 3.3.3 Estabilización de la dinámica del error

Una vez que obtuvimos (3.28), podemos proceder a estabilizar dicho sistema. De nuevo  $e = 0$ ,  $\tilde{W} = 0$ , es un punto de equilibrio para el sistema autónomo sin perturbaciones. Con el propósito de desarrollar un análisis de estabilidad, utilicemos de nuevo la función candidata de Lyapunov dada por

$$V = \frac{1}{2}\|e\|^2 + \frac{1}{2}tr \left\{ \tilde{W}^T \tilde{W} \right\} \quad (3.29)$$

cuya derivada a lo largo de las trayectorias de (3.28), es

$$\dot{V} = e^T(A + I)e + e^T\tilde{W}\Gamma z(x) + e^T\hat{W}\Gamma(z(x_p) - z(x_r)) + e^T u_2 + tr \left\{ \dot{\tilde{W}}^T \tilde{W} \right\} \quad (3.30)$$

Como en [23], proponemos la siguiente ley de adaptación de pesos y su descripción elemento a elemento,

$$\text{tr} \left\{ \dot{\hat{W}}^\top \hat{W} \right\} = -e^\top \tilde{W} \Gamma z(x) \quad (3.31)$$

$$\dot{\hat{w}}_{i,j} = e_i(\gamma_j z(x_j)), i = 1, 2, \dots, n, j = 1, 2, \dots, L.$$

Sustituyendo (3.31) en (3.30) nos da

$$\begin{aligned} \dot{V} &= -(\lambda - 1)e^\top e + e^\top \hat{W} \Gamma (z(x_p) - z(x_r)) + e^\top u_2 \\ L_f V &= -(\lambda - 1)e^\top e + e^\top \hat{W} \Gamma (z(x_p) - z(x_r)) \\ L_g V &= e^\top \end{aligned} \quad (3.32)$$

A continuación, considérese la siguiente desigualdad, demostrada en [24]:

$$X^\top Y + Y^\top X \leq X^\top \Lambda X + Y^\top \Lambda^{-1} Y \quad (3.33)$$

la cual se cumple para toda matriz  $X, Y \in \mathfrak{R}^{n \times k}$  y  $\Lambda \in \mathfrak{R}^{n \times n}$  con  $\Lambda = \Lambda^\top > 0$ . Aplicando esta desigualdad al segundo término del lado derecho de (3.32) obtenemos

$$\begin{aligned} \dot{V} &\leq -(\lambda - 1)e^\top e + \frac{1}{2}e^\top e + \frac{1}{2}(z(x_p) - z(x_r))^\top \Gamma (\hat{W}^\top \hat{W}) \Gamma (z(x_p) - z(x_r)) + e^\top u_2 \\ \dot{V} &\leq -(\lambda - 1)e^\top e + \frac{1}{2}e^\top e + \frac{1}{2}\|\hat{W}\|^2 \|\Gamma\|^2 \|z(x_p) - z(x_r)\|^2 + e^\top u_2 \end{aligned} \quad (3.34)$$

donde  $\|\hat{W}\|$ ,  $\|\Gamma\|$  las normas matriciales de  $\hat{W}$ ,  $\Gamma$ .

Considerando  $x_p = e + x_r$  y definiendo  $\phi_z = z(x_p) - z(x_r) = z(e + x_r) - z(x_r)$ , se tiene

$$\begin{aligned} \dot{V} &\leq -(\lambda - 1)e^\top e + \frac{1}{2}e^\top e + \frac{1}{2}L_{\phi_z}^2 \|\hat{W}\|^2 \|\Gamma\|^2 \|e\|^2 + e^\top u_2 \\ \dot{V} &\leq -(\lambda - 1)e^\top e + e^\top \left( \frac{1}{2} + \frac{1}{2}L_{\phi_z}^2 \|\hat{W}\|^2 \|\Gamma\|^2 \right) e + e^\top u_2 \end{aligned} \quad (3.35)$$

donde  $L_{\phi_z}^2$  es la constante de Lipschitz de  $\phi_z$ , que cumple con la siguiente propiedad

$$\begin{aligned} \phi_z &= z(x_p) - z(x_r) = z(e + x_r) - z(x_r) \\ &\leq L_{\phi_z}^2 \|e\| \end{aligned}$$

Ahora podemos definir la siguiente ley de control

$$\begin{aligned} u_2 &= -2 \left( \frac{1}{2} + \frac{1}{2}L_{\phi_z}^2 \|\hat{W}\|^2 \|\Gamma\|^2 \right) e \\ &\triangleq -\beta \left( R(e, \hat{W}) \right)^{-1} (L_g V)^\top, \beta \left( R(e, \hat{W}) \right)^{-1} = 2 \left( \frac{1}{2} + \frac{1}{2}L_{\phi_z}^2 \|\hat{W}\|^2 \|\Gamma\|^2 \right) \end{aligned} \quad (3.36)$$

con el escalar  $R(e, \hat{W}) > 0, \beta > 0$

Sustituyendo (3.36) en (3.35), obtenemos

$$\dot{V} = -(\lambda - 1)e^\top e - e^\top \left( \frac{1}{2} + \frac{1}{2}L_{\phi_z}^2 \|\hat{W}\|^2 \|\Gamma\|^2 \right) e < 0, \forall e \neq 0 \text{ y } \forall \hat{W} \neq 0 \quad (3.37)$$

Por lo tanto,  $\lim_{t \rightarrow \infty} e(t) \rightarrow 0$ .

Posteriormente se muestra que la acción de control dada por (3.36) minimiza un funcional de costo.



### 3.3.4 Funcional a Optimizar

Una vez que se tiene resuelto el problema de determinar la acción de control (3.36) que estabiliza a (3.28) se procede a formular el funcional el cual toma su valor óptimo precisamente usando dicha ley de control. El funcional está dado por:

$$J(u_2) = \lim_{t \rightarrow \infty} \left\{ 2\beta V + \int_0^t \left( l(e, \hat{W}) + u_2^\top R(e, \hat{W}) u_2 \right) d\tau \right\} \quad (3.38)$$

Además, la función de Lyapunov resuelve el siguiente sistema de ecuaciones en derivadas parciales de Hamilton-Jacobi-Bellman parametrizadas por el parámetro  $\beta > 0$ :

$$l(e, \hat{W}) + 2\beta L_f V - \beta^2 L_g V R(e, \hat{W})^{-1} L_g V^\top = 0, \quad (3.39)$$

Cabe aclarar que el término  $2\beta V$  en (3.38) es acotado cuando  $t \rightarrow \infty$ , ya que por (3.37)  $V$  es decreciente y acotada por abajo por  $V(0)$ , por lo que  $\lim_{t \rightarrow \infty} V(t)$  existe y es finito.

En [15] se requiere  $l(e, \hat{W})$  sea definido positivo y radialmente no acotado con respecto a  $e$ .

Despejando de (3.39) se tiene:

$$l(e, \hat{W}) = -2\beta L_f V + \beta^2 L_g V R(e, \hat{W})^{-1} L_g V^\top$$

y sustituyendo (3.36) en la ecuación anterior y aplicando la desigualdad (3.33) al segundo término del lado derecho de  $L_f V$  se demuestra que:

$$\begin{aligned} l(e, \hat{W}) &= -2\beta \left( -(\lambda - 1)e^\top e + e^\top \hat{W} \Gamma (z(x_p) - z(x_r)) \right) + \beta^2 e R(e, \hat{W})^{-1} e^\top, \\ l(e, \hat{W}) &\geq -2\beta \left( -(\lambda - 1)\|e\|^2 - \beta\|e\|^2 - \beta \Gamma^\top (z(x_p) - z(x_r))^\top \hat{W}^\top \hat{W} (z(x_p) - z(x_r)) \right. \\ &\quad \left. + \beta e \left( 1 + L_{\phi_z}^2 \|\hat{W}\|^2 \|\Gamma\|^2 \right) e^\top \right) \\ l(e, \hat{W}) &\geq -2\beta \left( -(\lambda - 1)\|e\|^2 - \beta\|e\|^2 - \beta \|\Gamma\|^2 L_{\phi_z}^2 \|\hat{W}\|^2 + \beta e \left( 1 + L_{\phi_z}^2 \|\hat{W}\|^2 \|\Gamma\|^2 \right) e^\top \right) \\ l(e, \hat{W}) &\geq 2\beta(\lambda - 1)\|e\|^2 - \beta\|e\|^2 - \beta \|\Gamma\|^2 L_{\phi_z}^2 \|\hat{W}\|^2 \|e\|^2 + \beta e \left( 1 + L_{\phi_z}^2 \|\hat{W}\|^2 \|\Gamma\|^2 \right) e^\top \\ l(e, \hat{W}) &\geq 2\beta(\lambda - 1)\|e\|^2 \\ l(e, \hat{W}) &\geq (\lambda - 1)\|e\|^2 \end{aligned}$$

Como  $\lambda > 1$ ,  $l(e, \hat{W})$  cumple la condición de ser positivo y radialmente no acotado, por lo que (3.38) es realmente un funcional de costo.

El integrando de (3.38) puede escribirse como

$$l(e, \hat{W}) + u_2^\top R(e, \hat{W}) u_2 = -2\beta (L_f V) + 2\beta^2 (L_g V) \left[ R(e, \hat{W}) \right]^{-1} (L_g V)^\top \quad (3.40)$$

Por otra parte, de (3.32) se tiene

$$\dot{V} = L_f V + L_g V \left[ -\beta \left( R(e, \hat{W}) \right)^{-1} \right] (L_g V)^\top$$

multiplicando por  $-2\beta$  se obtiene

$$-2\beta\dot{V} = -2\beta(L_f V) + 2\beta^2(L_g V) \left[ R(e, \hat{W}) \right]^{-1} (L_g V)^T$$

De donde

$$l(e, \hat{W}) + u_2^T R(e, \hat{W}) u_2 = -2\beta\dot{V} \quad (3.41)$$

Sustituyendo (3.41) en el funcional (3.38) se tiene

$$\begin{aligned} J(u_2) &= \lim_{t \rightarrow \infty} \left\{ 2\beta V - 2\beta \int_0^t \dot{V} d\tau \right\} \\ &= \lim_{t \rightarrow \infty} \{ 2\beta V(t) - 2\beta V(t) + 2\beta V(0) \} \\ &= 2\beta V(0) \end{aligned}$$

por lo que el valor óptimo de (3.38) es  $J^* = 2\beta V(0)$  y toma ese valor óptimo cuando la acción de control es de la forma (3.36).

Por último solo falta determinar la acción de control  $u$  que alimenta tanto a la red neuronal como la planta. Dicha ley se obtiene como sigue:

Teniendo en cuenta las ecuaciones (3.24), (3.21), (3.26), (3.27) y (3.36), se obtiene

$$\begin{aligned} u &= \tilde{u} + \alpha_r(t, \hat{W}) \\ &= u_1 + u_2 + \alpha_r(t, \hat{W}) \\ &= -\hat{W}\Gamma(z(x) - z(x_p)) - (A + I)(x - x_p) - 2 \left( \frac{1}{2} + \frac{1}{2} L_{\phi_z}^2 \|\hat{W}\|^2 \|\Gamma\|^2 \right) e + \alpha_r(t, \hat{W}) \end{aligned}$$

Por lo que la acción de control es

$$\begin{aligned} u &= -\hat{W}\Gamma(z(x) - z(x_p)) - (A + I)(x - x_p) - 2 \left( \frac{1}{2} + \frac{1}{2} L_{\phi_z}^2 \|\hat{W}\|^2 \|\Gamma\|^2 \right) e \\ &\quad + f_r(x_r, u_r) - Ax_r - \hat{W}\Gamma z(x_r) - x_r + x_p \end{aligned} \quad (3.42)$$

Los resultados obtenidos en esta sección los podemos establecer en el siguiente Teorema:

**Teorema 26** *Para el sistema no lineal desconocido modelado por (3.2), la ley de aprendizaje en línea (3.31) y la ley de control (3.42) en conjunto aseguran el seguimiento de la referencia no lineal (3.3).*



# Capítulo 4

## Control de sistemas caóticos

*En este capítulo se aplican los métodos desarrollados en el capítulo tercero para control de sistemas caóticos. Se consideran los casos de estabilización, la producción y la sincronización de caos, usando el esquema compuesto del identificador neuronal para la planta desconocida y las leyes de control obtenidas mediante estabilización cuadrática y optimización inversa.*

### 4.1 Introducción

El caos, como característica del comportamiento dinámico de un sistema, puede ser deseable o indeseable, dependiendo de la aplicación en cuestión; en procesos de mezclado, un comportamiento caótico podría mejorar la eficiencia del sistema, sin embargo en procesos que involucran vibraciones, el caos puede ocasionar fallas estructurales. Por lo tanto es importante manipular la naturaleza caótica, ya sea llevando sistemas estables a un comportamiento caótico cuando se deseé, o estabilizar un sistema caótico. También es deseable alterar la naturaleza caótica de un sistema sin que éste pierda dicho comportamiento. El propósito de este capítulo es mostrar la aplicabilidad de los esquemas de control presentados en el capítulo anterior para la estabilización, la producción y la sincronización de sistemas caóticos.

Se utilizan los esquemas presentados en el capítulo anterior, compuestos del identificador neuronal dado por (3.2)

$$\dot{x}_p = \dot{x} + w_{per} = Ax + W^* \Gamma z(x) + w_{per} + u$$

la ley de adaptación de pesos estimados que minimiza el error de modelado dada por (3.14)

$$\text{tr} \left\{ \dot{\hat{W}}^T \tilde{W} \right\} = -e^T \tilde{W} \Gamma z(x)$$

y las leyes de control obtenidas mediante estabilización cuadrática (3.19)

$$u = -\hat{W} \Gamma (z(x) - z(x_r)) - \mu e + f_r(x_r, u_r) - Ax_r - \hat{W} \Gamma z(x_r) - (A + I)(x - x_p)$$

y control óptimo inverso (3.42)

$$u = -\hat{W} \Gamma (z(x) - z(x_p)) - (A + I)(x - x_p) - 2 \left( \frac{1}{2} + \frac{1}{2} L_{\phi_z}^2 \|\hat{W}\|^2 \|\Gamma\|^2 \right) e \\ + f_r(x_r, u_r) - Ax_r - \hat{W} \Gamma z(x_r) - x_r + x_p$$

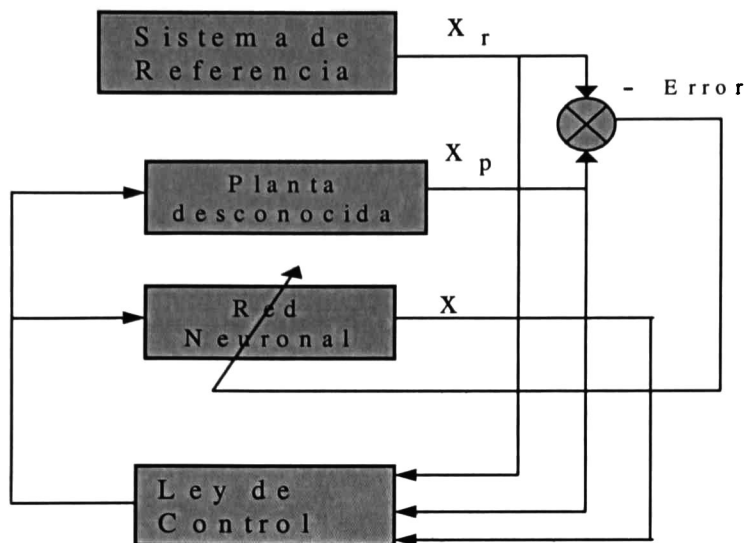


Figura 4.1: Esquema de control neuronal adaptable

Ambas leyes de control estabilizan globalmente asintóticamente la dinámica del error y aseguran el seguimiento de la trayectoria de referencia. Con el propósito de comparar las características de cada uno de los sistemas propuestos se aplicarán ambas leyes de control a cada uno de los ejemplos.

## 4.2 Estabilización de sistemas caóticos.

En esta sección se realiza la aplicación más simple de los esquemas de control propuestos: la estabilización de sistemas caóticos. Se utiliza como algoritmo de control la estabilización cuadrática. Se tratan los casos del atractor de Roessler y el atractor de Chua.

### 4.2.1 Estabilización del sistema de Roessler.

El oscilador de Roessler [30] es considerado el sistema no lineal más simple que presenta un comportamiento caótico puesto que presenta solamente una no linealidad cuadrática. El oscilador de Roessler está descrito por

$$\begin{aligned}
 \dot{x}_1 &= -x_2 - x_3 \\
 \dot{x}_2 &= x_1 + ax_2 \\
 \dot{x}_3 &= -cx_3 + b + x_1x_3
 \end{aligned}
 \tag{4.1}$$

donde los valores  $a, b$ , y  $c$  son parámetros físicos del sistema. En la figura 4.2 se presenta el atractor caótico para  $a = 0.4, b = 2, c = 4$  y condiciones iniciales  $x(0) = (4 \ -10 \ 15)$ . Nuestro propósito es estabilizar el sistema en el punto  $(1 \ 1 \ 1)$ . Por

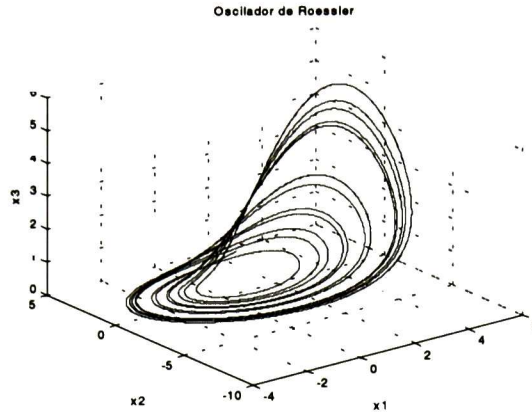


Figura 4.2: Diagrama de fase del oscilador de Rössler

lo tanto, se tiene

$$\begin{pmatrix} x_{r1} \\ x_{r2} \\ x_{r3} \end{pmatrix} = \begin{pmatrix} 1 \\ 1 \\ 1 \end{pmatrix} \rightarrow f_r(x_r) = \begin{pmatrix} 0 \\ 0 \\ 0 \end{pmatrix}$$

Se inicializó el sistema de Rössler desde tres condiciones iniciales distintas,

$$x_p(0) = \begin{pmatrix} 0 \\ -10 \\ 0 \end{pmatrix}, \begin{pmatrix} 1.1 \\ 0.5 \\ -1.2 \end{pmatrix}, \begin{pmatrix} 5 \\ -5 \\ -1.5 \end{pmatrix}$$

Consideremos el oscilador caótico de Rössler como planta desconocida, y usamos la siguiente red neuronal dinámica para modelar dicho sistema,

$$\dot{x} = Ax + W^* \Gamma z(x) + u$$

con los siguientes parámetros

$$A = -5I \quad \Gamma = 15I \quad z(x) = \begin{pmatrix} \tanh(kx_1) \\ \tanh(kx_2) \\ \tanh(kx_3) \end{pmatrix} \quad k = 0.65$$

Para este ejemplo, se usó el algoritmo de estabilización cuadrática; los resultados se muestran en las figuras 4.3 y 4.4, donde puede apreciarse que la respuesta del sistema es rápida. En la figura 4.4 se muestran los resultados obtenidos al realizar la simulación utilizando distintas condiciones iniciales. En la figura 4.5 se muestra la evolución en el tiempo de los pesos de la red neuronal. Estos se estabilizan en valores constantes, una vez que se logra un error de seguimiento cercano a cero.

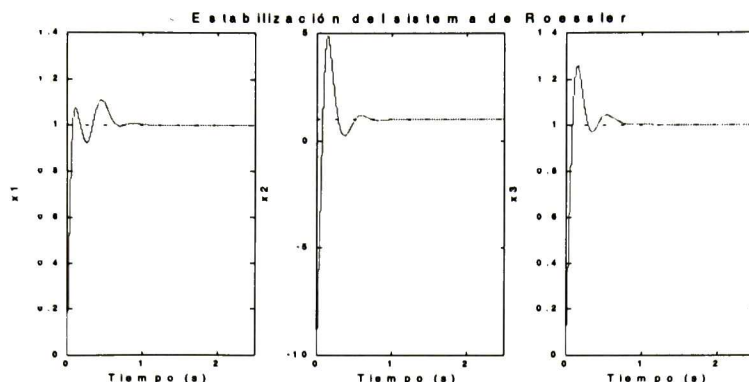


Figura 4.3: Estabilización del sistema de Roessler, evolución en el tiempo para  $x(0) = (0, -10, 0)$ .

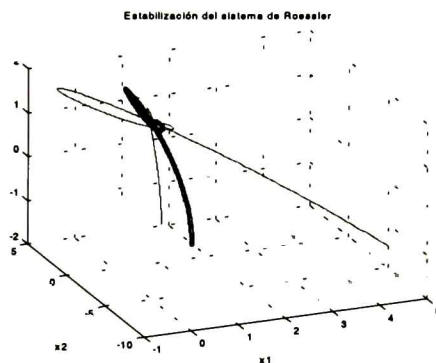


Figura 4.4: Espacio de fase de la estabilización del sistema de Roessler

### 4.2.2 Estabilización del sistema de Chua.

Se considera ahora el atractor de Chua definido por las ecuaciones (2.26), (2.27), con los parámetros

$$p = 10, q = 14.87, m_0 = -0.68, m_1 = -1.27$$

Se procede a estabilizar este sistema en su punto de equilibrio  $(\sqrt{5} \ 0 \ -\sqrt{5})$ ; se utilizó la misma estructura que para la estabilización del sistema de Roessler con los mismos parámetros.

Se inicializó el sistema de Chua desde tres condiciones iniciales distintas,

$$x_p(0) = \begin{pmatrix} 1.1 \\ 0.5 \\ -1.2 \end{pmatrix} \quad \begin{pmatrix} 5 \\ -5 \\ -1.5 \end{pmatrix} \quad \begin{pmatrix} 1 \\ 1 \\ 1 \end{pmatrix}$$

La evolución en el tiempo de los estados para el tercer caso se presenta en la figura 4.6 y el diagrama de fase para los tres casos en la figura 4.7. Como puede verse en las gráficas,

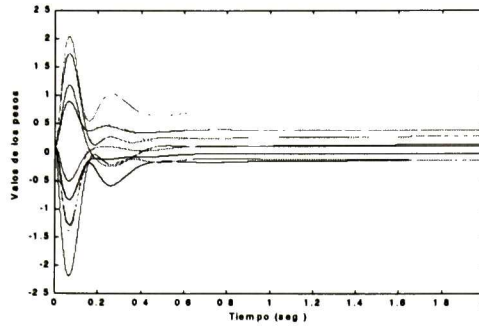


Figura 4.5: Estabilización del sistema de Roessler, evolución en el tiempo de los pesos de la red neuronal.

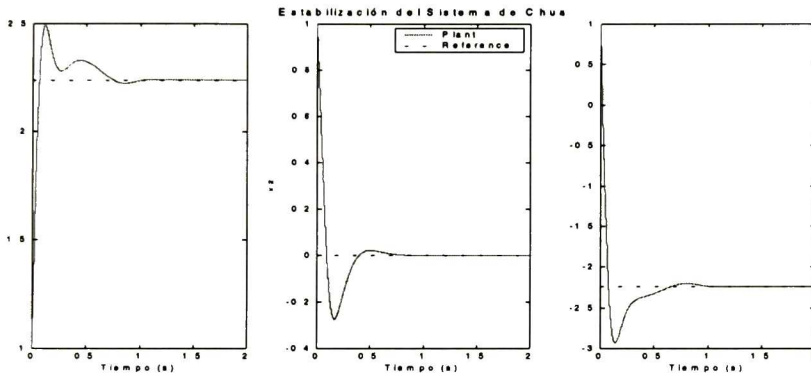


Figura 4.6: Estabilización del sistema de Chua usando funciones cuadráticas de Lyapunov

la estabilización se logra en forma rápida, al igual que en el sistema de Roessler, a pesar de que el sistema dinámico es mucho más complejo. Los pesos de la red neuronal (figura 4.8) se estabilizan en valores acotados una vez que se logra el error de seguimiento cercano a cero.

### 4.3 Producción de caos.

En contraste con los métodos usuales en control donde referencias constantes o periódicas son usadas para controlar los sistemas dinámicos, el concepto de usar como referencia una señal no periódica o caótica es relativamente nuevo.

En esta sección aplicamos los métodos descritos en el segundo capítulo para producir caos en sistemas dinámicos; es decir, sincronizar sistemas dinámicos con señales caóticas. Como se describió en el capítulo anterior, se utilizará una red neuronal recurrente para construir un modelo en línea del sistema desconocido



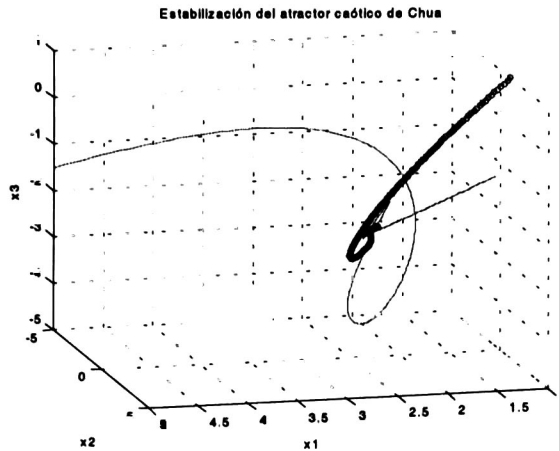


Figura 4.7: Plano de fase del sistema de Chua estabilizado mediante funciones cuadráticas de Lyapunov

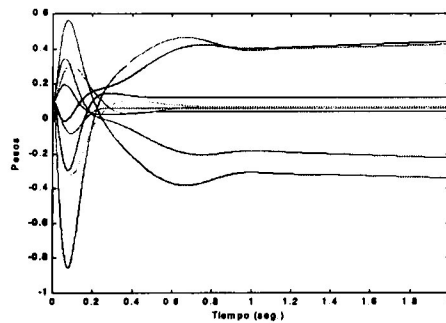


Figura 4.8: Evolución en el tiempo de los pesos de la red neuronal para la estabilización del sistema de Chua.

### 4.3.1 Seguimiento del atractor caótico de Chua.

Consideramos como planta desconocida el siguiente sistema lineal estable

$$\begin{aligned}
 \begin{pmatrix} \dot{x}_{p1} \\ \dot{x}_{p2} \\ \dot{x}_{p3} \end{pmatrix} &= \begin{pmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ -4.5 & -6.5 & -3 \end{pmatrix} \begin{pmatrix} x_{p1} \\ x_{p2} \\ x_{p3} \end{pmatrix} + \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix} u & \quad (4.2) \\
 \begin{pmatrix} y_{p1} \\ y_{p2} \\ y_{p3} \end{pmatrix} &= \begin{pmatrix} x_{p1} \\ x_{p2} \\ x_{p3} \end{pmatrix} & \quad \begin{pmatrix} x_{p1}(0) \\ x_{p2}(0) \\ x_{p3}(0) \end{pmatrix} = \begin{pmatrix} 0.5 \\ 0.5 \\ 0.5 \end{pmatrix}
 \end{aligned}$$

Nuestro objetivo, es forzar esta planta lineal a comportarse como el atractor caótico de Chua, descrito por las ecuaciones (2.28), (2.29) y donde

$$x_p(0) = \begin{pmatrix} 1.1 \\ 0.5 \\ -1.2 \end{pmatrix}$$

Se utiliza la misma estructura de red neuronal dinámica descrita en la sección 2.3.1

$$\dot{x} = Ax + W*\Gamma z(x) + u \tag{4.3}$$

Con el objeto de comparar, aplicamos los dos métodos propuestos. Para el método de estabilización cuadrática, se tomaron los siguientes parámetros para la estructura de la red neuronal dinámica.

$$A = -5I, \Gamma = 15I \quad z(x) = \begin{pmatrix} \tanh(kx_1) \\ \tanh(kx_2) \\ \tanh(kx_3) \end{pmatrix} \quad k = 0.65 \quad \mu = 10$$

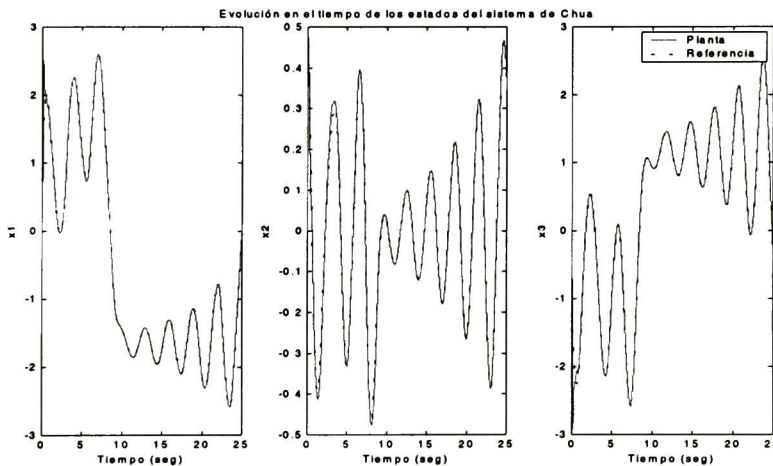


Figura 4.9: Seguimiento del atractor de Chua usando estabilización cuadrática. Evolución en el tiempo de los estados.

En las figuras 4.9, 4.10, 4.11 se muestran las gráficas contra tiempo de los estados y la acción de control, así como el diagrama de fase de la referencia y planta. El seguimiento de trayectorias se logra con éxito. También se puede observar que la acción de control es suave y acotada.

Para el método del control óptimo inverso los parámetros utilizados para el control fueron los mismos utilizados con estabilización cuadrática, sin embargo podemos reducir el parámetro de ganancia en la red neuronal y aún obtener un seguimiento de trayectoria exitoso.

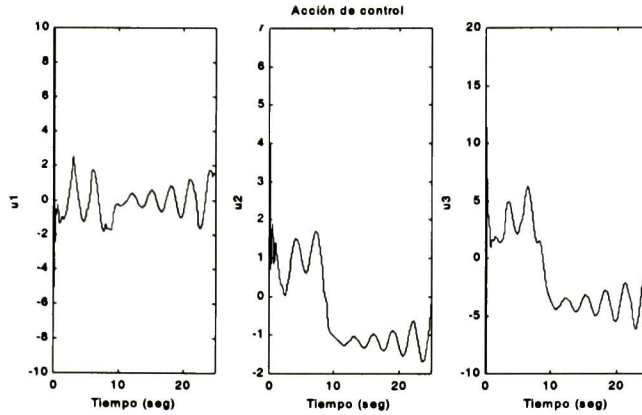


Figura 4.10: Seguimiento del sistema de Chua usando estabilización cuadrática. Acción de control

$$A = -5I, \Gamma = 5I \quad z(x) = \begin{pmatrix} \tanh(kx_1) \\ \tanh(kx_2) \\ \tanh(kx_3) \end{pmatrix} \quad k = 0.65$$

Los resultados obtenidos con el método del control óptimo inverso indican una estabilización de la dinámica del error más rápida que con estabilización cuadrática, pero demanda una magnitud mayor en la acción de control.

Una forma de aumentar la complejidad del problema de control es utilizar como modelo del sistema de Chua el dado por el sistema (2.26) y (2.27); utilizando los mismo parámetros que en el ejemplo anterior, se obtiene un seguimiento de trayectorias satisfactorio, pero con una acción de control más oscilatoria aunque todavía acotada, consecuencia de la complejidad del sistema. Los resultados se presentan en la figuras 4.15, 4.16 y 4.17.

### 4.3.2 Seguimiento del atractor caótico de Chen.

En este caso consideraremos como referencia el atractor caótico de Chen, descrito en la sección 2.1.4. Este sistema tiene una dinámica mucho más rápida y compleja que la del atractor de Chua. La dinámica del sistema de Chen está dada por el sistema (2.22) donde

$$x_r(0) = \begin{pmatrix} -10 & 5 & 0.5 \end{pmatrix}^T$$

Para el método de estabilización cuadrática se utilizó el identificador neuronal

$$\dot{x} = Ax + W^*\Gamma z(x) + u \tag{4.4}$$

donde

$$A = -5I \quad \Gamma = 150I \quad z(x) = \tanh(0.65x) \quad \mu = 20$$

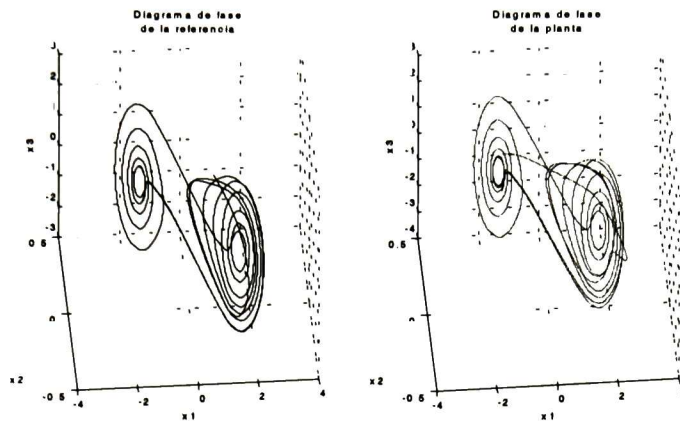


Figura 4.11: Seguimiento del sistema de Chua usando estabilización cuadrática: diagrama de fase de la referencia y planta

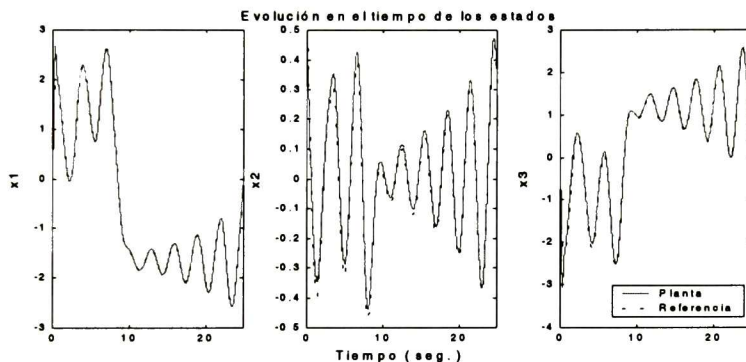


Figura 4.12: Seguimiento del sistema de Chua usando el método del control óptimo inverso. Evolución de los estados.

Al usar la ley de control (3.42); se emplearon los siguientes parámetros para el identificador neuronal

$$A = -5I \quad \Gamma = 15I \quad z(x) = \tanh(0.65x)$$

En las figuras 4.20 y 4.21 se verifica que el seguimiento de logra de forma más rápida que utilizando la ley de control (3.19)

## 4.4 Sincronización de caos.

Con el propósito de mostrar la aplicabilidad del algoritmo desarrollado en el Capítulo tercero, ante plantas más complejas que las presentadas en la sección anterior, se presenta la sincronización de caos. Se tomará como planta desconocida a un sistema caótico y como referencia un sistema caótico distinto.



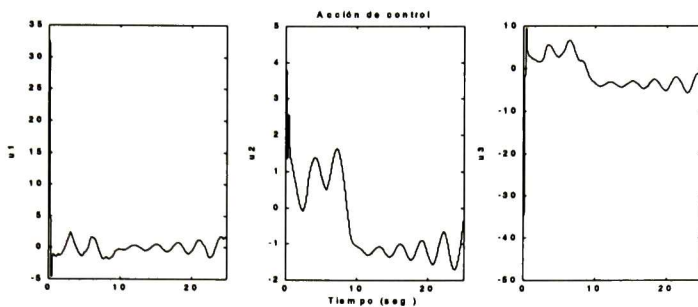


Figura 4.13: Seguimiento del sistema de Chua usando el método del control óptimo inverso. Respuesta de control.

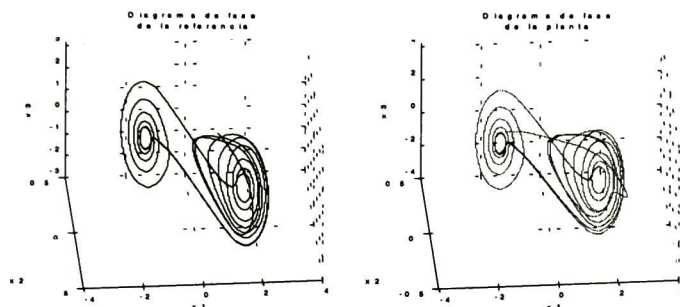


Figura 4.14: Seguimiento del sistema de Chua usando el método del control óptimo inverso. Diagramas de fase de referencia y planta

En este caso, la planta desconocida considerada es el atractor caótico de Chen generado por (2.22) donde

$$\begin{pmatrix} x_{r1}(0) \\ x_{r2}(0) \\ x_{r3}(0) \end{pmatrix} = \begin{pmatrix} -10 \\ 5 \\ 0.5 \end{pmatrix}$$

El objetivo es forzar el atractor caótico de Chen a seguir una referencia dada por el atractor de Lorenz cuyas ecuaciones son

$$\begin{aligned} \dot{x}_{r1} &= \sigma(x_{r2} - x_{r1}) \\ \dot{x}_{r2} &= rx_{r1} - x_{r2} - x_{r2}x_{r3} \\ \dot{x}_{r3} &= -bx_{r3} + x_{r1}x_{r2} \end{aligned}$$

donde

$$\sigma = 10, \quad b = 8/3, \quad r = 28, \quad \begin{pmatrix} x_{r1}(0) \\ x_{r2}(0) \\ x_{r3}(0) \end{pmatrix} = \begin{pmatrix} 1 \\ 1 \\ 1 \end{pmatrix}$$

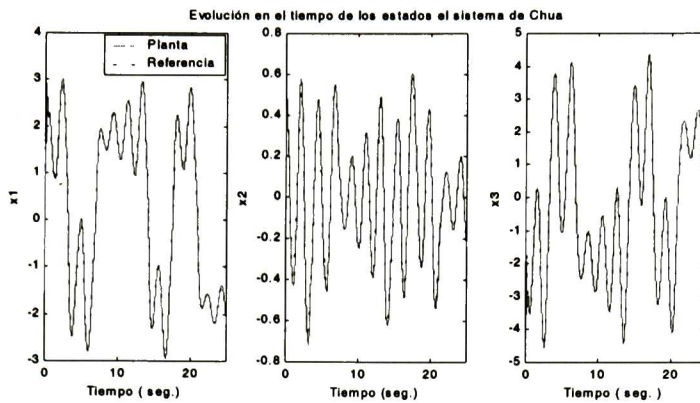


Figura 4.15: Seguimiento del sistema de Chua dado por (2.26) y (2.27) usando el método del control óptimo inverso. Evolución de los estados.

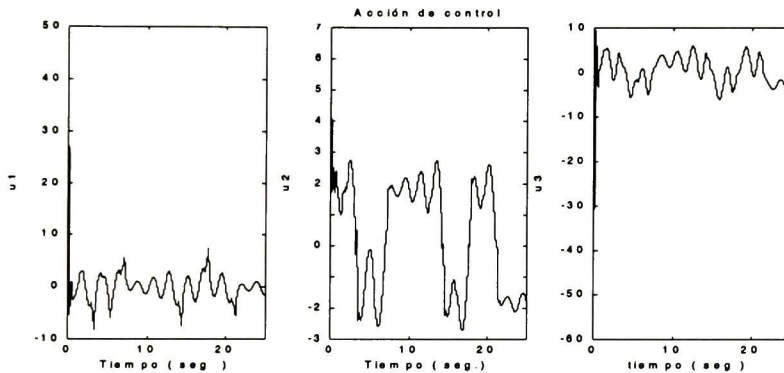


Figura 4.16: Seguimiento del sistema de Chua dado por (2.26) y (2.27) usando el método del control óptimo inverso. Acción de control obtenida con el método de estabilización cuadrática

En las simulaciones, se usó la siguiente red neuronal dinámica:

$$\dot{x} = Ax + W^*\Gamma z(x) + u$$

Los parámetros de la red  $A, \Gamma$ , y  $z(x)$  se definirán para cada uno de los esquemas de control propuestos.

Los parámetros de la red y la acción de control utilizados con estabilización cuadrática fueron

$$A = \begin{pmatrix} -5 & 0 & 0 \\ 0 & -5 & 0 \\ 0 & 0 & -5 \end{pmatrix} \quad \Gamma = \begin{pmatrix} 150 & 0 & 0 \\ 0 & 150 & 0 \\ 0 & 0 & 150 \end{pmatrix} \quad z(x) = \begin{pmatrix} \tanh(\beta x_1) \\ \tanh(\beta x_2) \\ \tanh(\beta x_3) \end{pmatrix}, \quad \beta = 0.5, \quad \mu = 10$$

En las figuras 4.22 y 4.23 se muestran los resultados. Como puede apreciarse la sincronización de caos se logra con éxito.

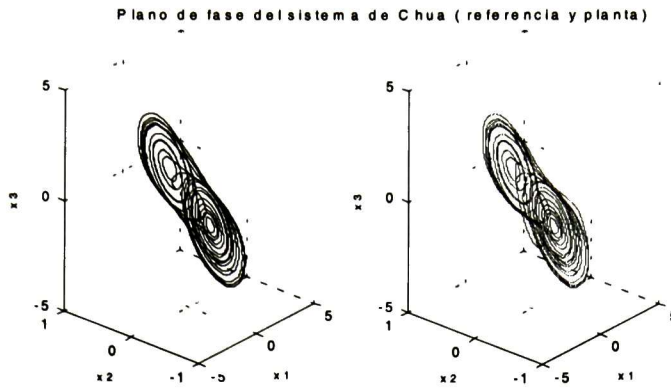


Figura 4.17: Planos de fase de referencia y planta para seguimiento del sistema de Chua dado por (2.26) y (2.27) usando control óptimo inverso

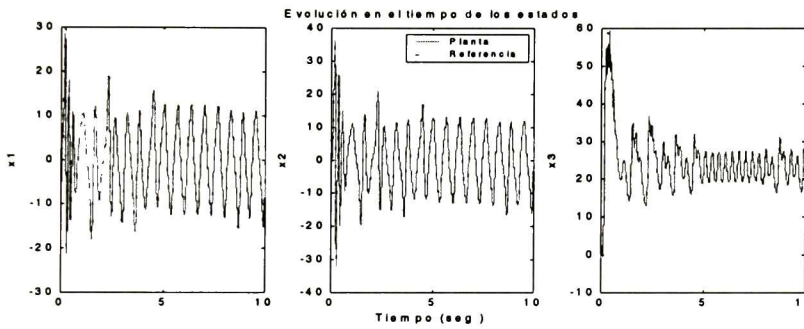


Figura 4.18: Seguimiento del sistema de Chen usando estabilización cuadrática

Para la ley de control basada en control óptimo inverso (3.42), los parámetros utilizados en el identificador neuronal fueron

$$A = \begin{pmatrix} -5 & 0 & 0 \\ 0 & -5 & 0 \\ 0 & 0 & -5 \end{pmatrix} \quad \Gamma = \begin{pmatrix} 15 & 0 & 0 \\ 0 & 15 & 0 \\ 0 & 0 & 15 \end{pmatrix} \quad z(x) = \begin{pmatrix} \tanh(\beta x_1) \\ \tanh(\beta x_2) \\ \tanh(\beta x_3) \end{pmatrix} \quad \beta = 0.65$$

en la figuras 4.24 a 4.28 se muestran los resultados de la sincronización. La respuesta de la planta es más rápida usando la ley (3.42) lo cual se refleja en el diagrama de fase. Los esquemas propuestos funcionaron adecuadamente, presentando características diferentes; por una parte el método de estabilización cuadrática presenta una acción de control de amplitud menor que la obtenida con el óptimo inverso, pero con respuesta ligeramente más lenta. El control óptimo inverso produce una rápida estabilización del error en el origen pero la amplitud de la acción de control es de mayor amplitud y frecuencia que la obtenida con el óptimo inverso. Ambas características pueden ser modificadas variando los parámetros de la red neuronal para obtener un desempeño adecuado.

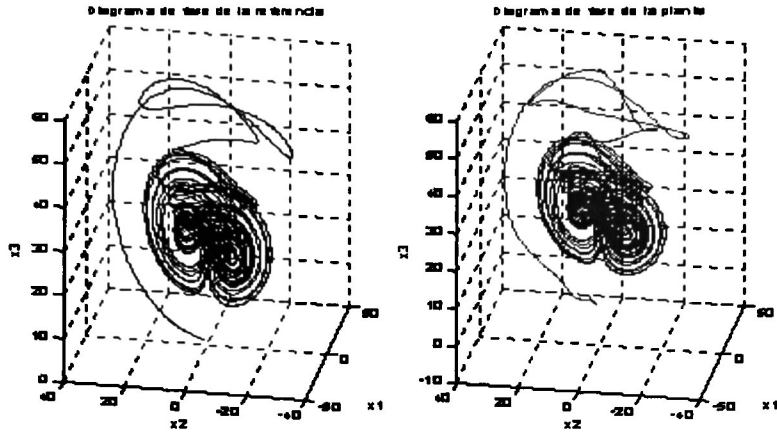


Figura 4.19: Diagrama de fase de referencia y planta para seguimiento del sistema de Chen usando estabilización cuadrática

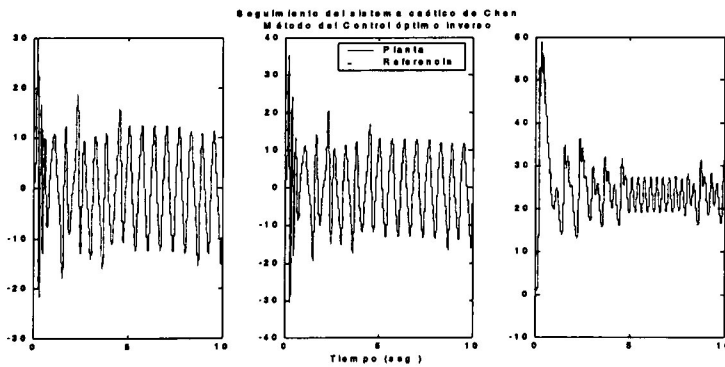


Figura 4.20: Evolución de los estados para seguimiento del sistema de Chen usando control óptimo inverso.



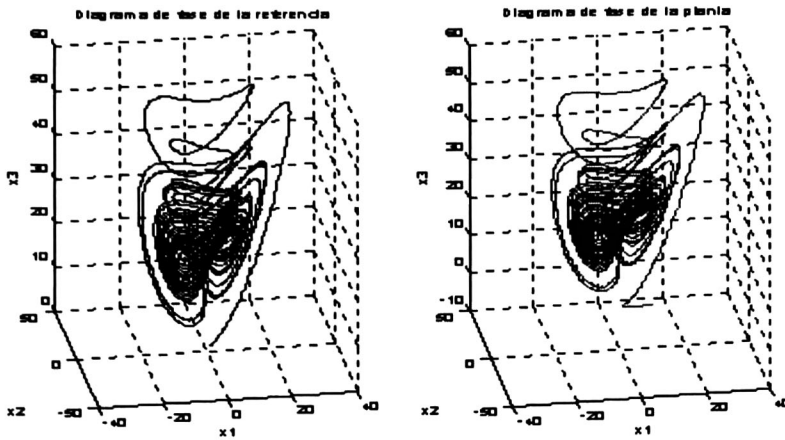


Figura 4.21: Diagrama de fase de referencia y planta para seguimiento del sistema de Chen usando control óptimo inverso.

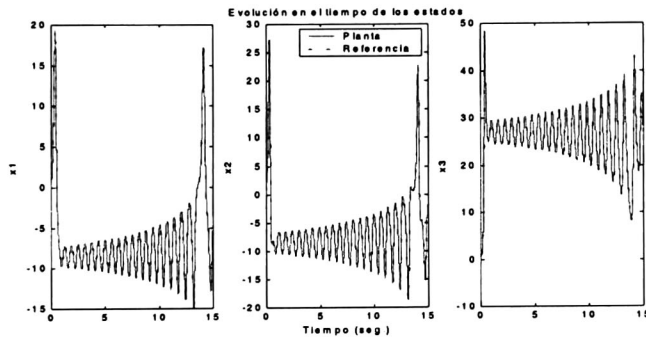


Figura 4.22: Sincronización de caos mediante estabilización cuadrática: evolución en el tiempo de los estados.

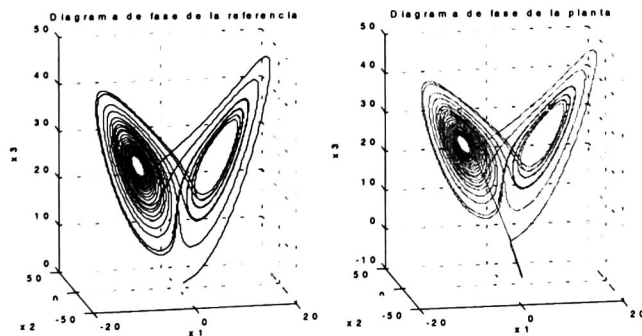


Figura 4.23: Sincronización de caos mediante estabilización cuadrática: diagramas de fase de referencia y planta

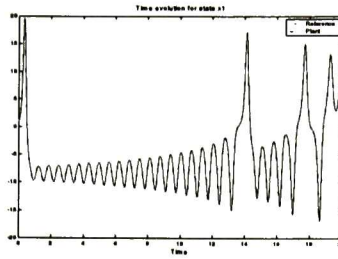


Figura 4.24: Sincronización de caos mediante control óptimo inverso: evolución en el tiempo del estado 1.

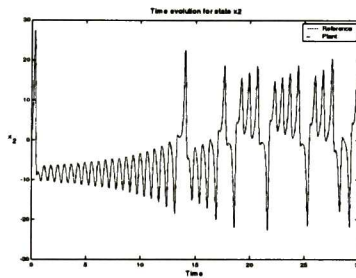


Figura 4.25: Sincronización de caos mediante control óptimo inverso: evolución en el tiempo del estado 2

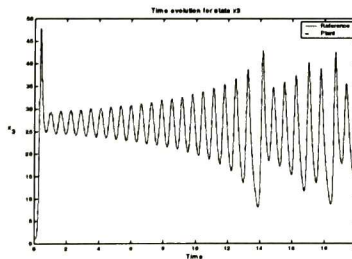


Figura 4.26: Sincronización de caos mediante control óptimo inverso: evolución en el tiempo del estado 3

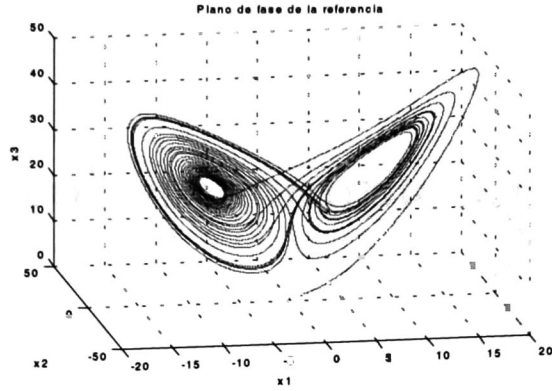


Figura 4.27: Sincronización de caos mediante control óptimo inverso: diagrama de fase de la referencia (Atractor de Lorenz)

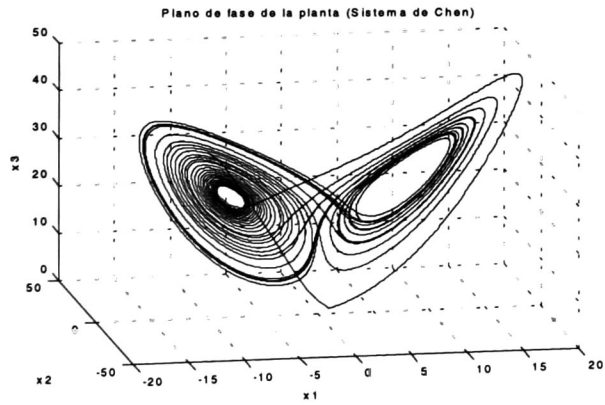


Figura 4.28: Sincronización de caos mediante control óptimo inverso: diagrama de fase de la planta (Atractor de Chen)

# Capítulo 5

## Control de robots completamente actuados.

*En este capítulo se extienden los resultados obtenidos previamente para seguimiento de trayectorias en plantas desconocidas usando redes neuronales recurrentes. Se considera el caso en el que se tienen menos entradas que estados. La estructura propuesta consta de un identificador neuronal y una ley de control obtenida mediante el método del control óptimo inverso. La aplicabilidad de este algoritmo se muestra mediante el control de una manipulador robótico completamente actuado.*

### 5.1 Introducción.

Los manipuladores robóticos son sistemas usados comúnmente en la industria en tareas como soldadura, pintura, manipulación de materiales en líneas de manufactura, etc. En general, se desea que el manipulador conduzca su elemento final a lo largo de una trayectoria. Uno de los problemas que enfrentan las leyes de control desarrolladas es la reducida área de trabajo que se obtiene con algunos controladores, así como tratar con los problemas que trae el manejar masas de diferente tamaño y peso lo cual hace que cambie la dinámica del robot. Para tratar con dicho problema se ha desarrollado el control adaptable, el cual da como resultado la robustez del sistema bajo ciertas condiciones. Con el propósito de evitar los problemas relacionados con la variación en los parámetros inciertos y perturbaciones, se hace uso de un identificador neuronal dinámico.

En el Capítulo tercero se presentaron algoritmos de control neuronal adaptable para seguimiento de trayectorias no lineales en plantas desconocidas. En este capítulo se extienden los resultados obtenidos para el caso en que se tienen menos entradas que estados. Debido a que en la aplicación de control de sistemas caóticos presentó un mejor desempeño, se realizó el análisis utilizando el método del control óptimo inverso.

Nos enfocaremos en control óptimo robusto para sistemas lineales con incertidumbres en el modelado. Bajo la suposición de estabilidad entrada-estado del sistema (red neuronal) se obtendrá una acción de control que asegurará estabilidad asintótica y seguimiento de trayectorias con rechazo a perturbaciones y atenuación del efecto de las incertidumbres en el modelado. Utilizando el mismo esquema de control compuesto del identificador neuronal y la ley de control óptima, se aplicará en el seguimiento de trayectorias para un manipulador



robótico de dos grados de libertad completamente actuado.

## 5.2 Control de un manipulador completamente actuado.

### 5.2.1 Descripción del algoritmo.

Para controlar un manipulador completamente actuado de dos grados de libertad, consideramos el esquema de control basado en el método del control óptimo inverso presentado en el Capítulo tercero, tomando en cuenta ahora que se tienen menos entradas que estados. Luego el sistema que describe la planta modelada por la red neuronal estará dado por,

$$\dot{x}_p = \dot{x} + w_{per} = Ax + W^*\Gamma z(x) + (x - x_p) + \Omega u \quad (5.1)$$

donde  $x_p, x, z(x) \in \mathbb{R}^n$ ,  $W^*, A, \Gamma \in \mathbb{R}^{n \times n}$ ,  $u \in \mathbb{R}^m$  y  $\Omega \in \mathbb{R}^{n \times m}$ .

De nuevo, consideramos la trayectoria de referencia dada por,

$$\dot{x}_r = f_r(x_r, u_r), \quad x_r \in \mathbb{R}^n \quad (5.2)$$

y definamos el error de seguimiento por

$$e := x_p - x_r \quad (5.3)$$

Su derivada con respecto al tiempo es

$$\dot{e} = \dot{x}_p - \dot{x}_r = Ax + W^*\Gamma z(x) + w_{per} + \Omega u - f_r(x_r, u_r) \quad (5.4)$$

Un análisis análogo al realizado para obtener (3.20) nos lleva a

$$\begin{aligned} \dot{e} = & Ae + W^*\Gamma z(x) + \Omega u + \left( -f_r(x_r, u_r) + Ax_r + \hat{W}\Gamma z(x_r) + x_r - x_p + \Omega\alpha_r(t, \hat{W}) \right) \\ & - Ae - \hat{W}\Gamma z(x_r) - \Omega\alpha_r(t, \hat{W}) - Ax_r - x_r + x + Ax \end{aligned} \quad (5.5)$$

y si se satisfacen las condiciones de la proposición 20, se tiene

$$Ax_r + \hat{W}\Gamma z(x_r) + x_r - x_p + \Omega\alpha_r(t, \hat{W}) = f_r(x_r, u_r) \quad (5.6)$$

De donde

$$\alpha_r(t, \hat{W}) = (\Omega^T \Omega)^{-1} \Omega^T \left( f_r(x_r, u_r) - Ax_r - \hat{W}\Gamma z(x_r) - (x_r - x_p) \right) \quad (5.7)$$

Reduciendo (5.5), definiendo de nuevo con

$$\tilde{W} = W^* - \hat{W} \quad \tilde{u} = u - \alpha_r(t, \hat{W}), \quad (5.8)$$

y sustituyendo (5.6) en (5.5), obtenemos

$$\begin{aligned} \dot{e} = & Ae + \tilde{W}\Gamma z(x) + \hat{W}\Gamma(z(x) - z(x_r)) + (A + I)(x - x_r) - Ae + \Omega\tilde{u} \\ = & Ae + \tilde{W}\Gamma z(x) + \hat{W}\Gamma(z(x) - z(x_p) + z(x_p) - z(x_r)) \\ & + (A + I)(x - x_p + x_p - x_r) - Ae + \Omega\tilde{u} \end{aligned} \quad (5.9)$$

Sea ahora:

$$\tilde{u} = u_1 + u_2 \quad (5.10)$$

Definimos

$$u_1 = (\Omega^T \Omega)^{-1} \Omega^T \left( -\hat{W} \Gamma (z(x) - z(x_p)) - (A + I)(x - x_p) \right) \quad (5.11)$$

por lo que (5.9) se reduce a:

$$\dot{e} = Ae + \tilde{W} \Gamma z(x) + \hat{W} \Gamma z(x_p) - z(x_r) + (A + I)(x_p - x_r) - Ae + \Omega u_2$$

Teniendo en cuenta que  $e = x_p - x_r$ , la ecuación anterior se reduce a la siguiente:

$$\dot{e} = (A + I)e + \tilde{W} \Gamma z(x) + \hat{W} \Gamma z(x_p) - z(x_r) + \Omega u_2 \quad (5.12)$$

Ahora el problema consiste en obtener una acción de control  $u_2$ , la cual estabilice al sistema (5.12). Dicha ley de control se obtendrá usando la metodología del control óptimo inverso

### Estabilización de la dinámica del error.

Una vez que obtuvimos (5.12), procedemos a estabilizar dicho sistema. De nuevo  $e = 0$ ,  $\tilde{W} = 0$ , es un punto de equilibrio para el sistema autónomo sin perturbaciones. Con el propósito de desarrollar un análisis de estabilidad, utilicemos de nuevo la función candidata de Lyapunov dada por

$$V = \frac{1}{2} \|e\|^2 + \frac{1}{2} \text{tr} \left\{ \tilde{W}^T \tilde{W} \right\} \quad (5.13)$$

cuya derivada a lo largo de las trayectorias de (5.12), es

$$\dot{V} = e^T (A + I)e + e^T \tilde{W} \Gamma z(x) + e^T \hat{W} \Gamma (z(x_p) - z(x_r)) + e^T \Omega u_2 + \text{tr} \left\{ \dot{\tilde{W}}^T \tilde{W} \right\} \quad (5.14)$$

Como en [23], proponemos la siguiente ley de adaptación de pesos:

$$\text{tr} \left\{ \dot{\tilde{W}}^T \tilde{W} \right\} = -e^T \tilde{W} \Gamma z(x) \quad (5.15)$$

la cual puede escribirse elemento a elemento como,

$$\dot{\hat{w}}_{i,j} = e_i (\gamma_j z(x_j)), \quad i = 1, 2, \dots, n, \quad j = 1, 2, \dots, L.$$

Sustituyendo (5.15) en (5.14) nos da

$$\begin{aligned} \dot{V} &= -(\lambda - 1)e^T e + e^T \hat{W} \Gamma (z(x_p) - z(x_r)) + e^T \Omega u_2 \\ L_j V &= -(\lambda - 1)e^T e + e^T \hat{W} \Gamma (z(x_p) - z(x_r)) \\ L_g V &= e^T \end{aligned} \quad (5.16)$$

Considerando la desigualdad (3.33) y aplicándola al segundo término del lado derecho de (5.16) obtenemos

$$\begin{aligned}\dot{V} &\leq -(\lambda - 1)e^\top e + \frac{1}{2}e^\top e + \frac{1}{2}(z(x_p) - z(x_r))^\top \Gamma (\hat{W}^\top \hat{W}) \Gamma (z(x_p) - z(x_r)) \\ &\quad + e^\top \Omega u_2 \\ \dot{V} &\leq -(\lambda - 1)e^\top e + \frac{1}{2}e^\top e + \frac{1}{2}\|\hat{W}\|^2 \|\Gamma\|^2 \|z(x_p) - z(x_r)\|^2 + e^\top \Omega u_2\end{aligned}\quad (5.17)$$

donde  $\|\hat{W}\|$ ,  $\|\Gamma\|$  son las normas matriciales de  $\hat{W}$ ,  $\Gamma$

Considerando  $x_p = e + x_r$  y definiendo  $\phi_z = z(x_p) - z(x_r) = z(e + x_r) - z(x_r)$ , se tiene

$$\begin{aligned}\dot{V} &\leq -(\lambda - 1)e^\top e + \frac{1}{2}e^\top e + \frac{1}{2}L_{\phi_z}^2 \|\hat{W}\|^2 \|\Gamma\|^2 \|e\|^2 + e^\top \Omega u_2 \\ \dot{V} &\leq -(\lambda - 1)e^\top e + e^\top \left( \frac{1}{2} + \frac{1}{2}L_{\phi_z}^2 \|\hat{W}\|^2 \|\Gamma\|^2 \right) e + e^\top \Omega u_2\end{aligned}\quad (5.18)$$

donde  $L_{\phi_z}$  es la constante de Lipschitz de  $\phi_z$

Ahora podemos definir la siguiente ley de control

$$\begin{aligned}u_2 &= -\mu (\Omega^T \Omega)^{-1} \Omega^T \left( 1 + L_{\phi_z}^2 \|\hat{W}\|^2 \|\Gamma\|^2 \right) e \quad \mu > \frac{1}{2} \\ &\triangleq -\beta \left( R(e, \hat{W}) \right)^{-1} (L_g V)^\top, \\ \beta \left( R(e, \hat{W}) \right)^{-1} &= \mu (\Omega^T \Omega)^{-1} \Omega^T \left( 1 + L_{\phi_z}^2 \|\hat{W}\|^2 \|\Gamma\|^2 \right)\end{aligned}\quad (5.19)$$

con el escalar  $R(e, \hat{W}) > 0, \beta > 0$

Sustituyendo (5.19) en (5.18), obtenemos

$$\dot{V} = -(\lambda - 1)e^\top e - \left( \mu - \frac{1}{2} \right) e^\top \left( 1 + L_{\phi_z}^2 \|\hat{W}\|^2 \|\Gamma\|^2 \right) e < 0, \forall e, \hat{W} \neq 0 \quad (5.20)$$

Finalmente la acción de control está dada por

$$\begin{aligned}u &= \tilde{u} + \alpha_r(t, \hat{W}) \\ &= u_1 + u_2 + \alpha_r(t, \hat{W}) \\ &= (\Omega^T \Omega)^{-1} \Omega^T \left[ -\hat{W} \Gamma (z(x) - z(x_p)) - (A + I)(x - x_p) \right. \\ &\quad \left. - \mu \left( 1 + L_{\phi_z}^2 \|\hat{W}\|^2 \|\Gamma\|^2 \right) e + \alpha_r(t, \hat{W}) \right]\end{aligned}$$

Por lo que la acción de control que actúa sobre la red neuronal y el manipulador es:

$$\begin{aligned}u &= (\Omega^T \Omega)^{-1} \Omega^T \left( -\hat{W} \Gamma (z(x) - z(x_p)) - (A + I)(x - x_p) \right. \\ &\quad \left. - \mu \left( 1 + L_{\phi_z}^2 \|\hat{W}\|^2 \|\Gamma\|^2 \right) e + f_r(x_r, u_r) - A x_r - \hat{W} \Gamma z(x_r) - x_r + x_p \right)\end{aligned}\quad (5.21)$$

Esta ley de control estabiliza asintóticamente la dinámica del error y asegura el seguimiento de la señal de referencia.

### Funcional de costo a optimizar

Una vez que se ha resuelto el problema de hallar la ley de control (5.21) basada en el método del control óptimo inverso que establezca (5.12), podemos proceder a formular el funcional de costo que obtiene su valor óptimo con la ley de control (5.19). Este funcional está dado por

$$J(u_2) = \lim_{t \rightarrow \infty} \left\{ 2\beta V + \int_0^t \left( l(e, \hat{W}) + u_2^\top R(e, \hat{W}) u_2 \right) d\tau \right\} \quad (5.22)$$

donde la función de Lyapunov resuelve la familia de ecuaciones en derivadas parciales de Hamilton-Jacobi-Bellman parametrizadas con  $\beta > 0$ :

$$l(e, \hat{W}) + 2\beta L_f V - \beta^2 L_g V R(e, \hat{W})^{-1} L_g V^\top = 0, \quad (5.23)$$

En [15] se da como condición necesaria que  $l(e, \hat{W})$  sea definido positivo y radialmente no acotado con respecto a  $e$ .

De (5.23) tenemos

$$l(e, \hat{W}) = -2\beta L_f V + \beta^2 L_g V R(e, \hat{W})^{-1} L_g V^\top \quad (5.24)$$

Reemplazando (5.19) en (5.24) y aplicando (3.33) al segundo término del lado derecho de  $L_f V$  se puede demostrar que

$$l(e, \hat{W}) \geq (\lambda - 1) \|e\|^2 + (\mu - 1) (1 + L_\phi^2 \|\Gamma\|^2 \|W\|^2) \|e\|^2$$

Como se tiene que  $\lambda > 1$  y  $\mu > 1$ ,  $l(e, \hat{W})$  satisface la condición de ser definido positivo y radialmente no acotado; por lo tanto (5.22) es un funcional de costo.

Es fácil verificar que en (5.22):

$$l(e, \hat{W}) + u_2^\top R(e, \hat{W}) u_2 = -2\beta \dot{V}$$

con valor óptimo  $J^* = 2\beta V(0)$ , y toma ese valor con la ley de control (5.21).

### 5.2.2 Modelado del manipulador

Consideremos el modelo de un manipulador completamente actuado con dos grados de libertad, como se muestra en la figura 5.1. Para la obtención del modelo no lineal de este manipulador se utilizan las ecuaciones dinámicas de Euler-Lagrange.

En la figura se tiene la siguiente notación

- $q_1, q_2 \rightarrow$  Posición angular de los eslabones 1 y 2
- $l_1, l_2 \rightarrow$  Longitud de los eslabones 1 y 2
- $l_{c1}, l_{c2} \rightarrow$  Ubicación del centro de gravedad de los eslabones 1 y 2
- $I_1, I_2 \rightarrow$  Momento de inercia de cada uno de los eslabones
- $g \rightarrow$  Aceleración gravitacional (9.81  $m/s^2$ )



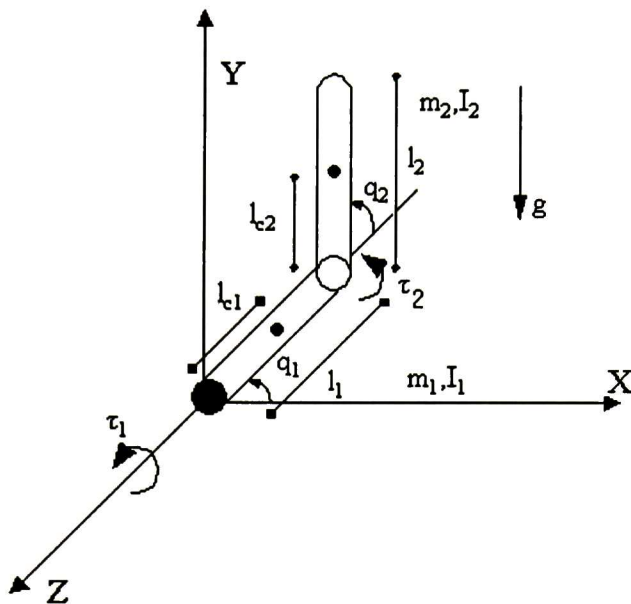


Figura 5.1: Manipulador planar de dos grados de libertad

La ecuación de Euler-Lagrange está dada por:

$$\frac{d}{dt} \left( \frac{\partial L}{\partial \dot{q}_i} \right) - \frac{\partial L}{\partial q_i} = \tau_i, \quad i = 1, 2, \dots, n \quad (5.25)$$

donde  $q_i$  es una coordenada generalizada que en este caso viene dada por la posición angular del eslabón  $i$ ,  $L$  es el Lagrangiano que se define como la diferencia entre la energía cinética  $K$  y la energía potencial  $V$  de cada uno de los eslabones.

$$L = K - V \quad (5.26)$$

y  $\tau_i$  es el par aplicado a el actuador  $i$ .

La energía cinética  $K$  está dada por:

$$K = \frac{1}{2} \left( \sum_{i=1}^2 m_i v_{ci}^T v_{ci} + \omega_i^T I_i \omega_i \right) \quad i = 1, 2 \quad (5.27)$$

donde  $v_{ci}$  y  $\omega_i$  representan los vectores de velocidad lineal y angular del  $i$ -ésimo eslabón e  $I_i$  es la matriz de inercia.

El vector de posición del centro de masa de cada uno de los eslabones está dado por

$$r_{c1}(q) = \begin{bmatrix} r_{1x}(q) \\ r_{1y}(q) \end{bmatrix} = \begin{bmatrix} l_{c1} \cos(q_1) \\ l_{c1} \sin(q_1) \end{bmatrix} \quad (5.28)$$

$$r_{c2}(q) = \begin{bmatrix} r_{2x}(q) \\ r_{2y}(q) \end{bmatrix} = \begin{bmatrix} l_1 \cos(q_1) + l_{c2} \cos(q_1 + q_2) \\ l_1 \sin(q_1) + l_{c2} \sin(q_1 + q_2) \end{bmatrix} \quad (5.29)$$

Tomando la derivada de las ecuaciones (5.28) y (5.29) con respecto al tiempo, se obtiene la velocidad lineal

$$\dot{r}_{c1}(q) = v_{c1} = \begin{bmatrix} -l_{c1} \text{sen}(q_1) & 0 \\ l_{c1} \cos(q_1) & 0 \end{bmatrix} \begin{bmatrix} \dot{q}_1 \\ \dot{q}_2 \end{bmatrix} \quad (5.30)$$

$$\dot{r}_{c2}(q) = v_{c2} = \begin{bmatrix} -l_1 \text{sen}(q_1) - l_{c2} \text{sen}(q_1 + q_2) & -l_{c2} \text{sen}(q_1 + q_2) \\ l_1 \cos(q_1) + l_{c2} \cos(q_1 + q_2) & l_{c2} \cos(q_1 + q_2) \end{bmatrix} \begin{bmatrix} \dot{q}_1 \\ \dot{q}_2 \end{bmatrix} \quad (5.31)$$

Las velocidades angulares son definidas con respecto al eje de rotación

$$\omega_1 = \begin{bmatrix} 0 \\ 0 \\ \dot{q}_1 \end{bmatrix} \quad \omega_2 = \begin{bmatrix} 0 \\ 0 \\ \dot{q}_1 + \dot{q}_2 \end{bmatrix} \quad (5.32)$$

En(5.27), la matriz de inercia  $I_i$  se define como

$$I_i = \begin{pmatrix} I_{xxi} & -I_{xyi} & -I_{xzi} \\ -I_{yxi} & I_{yyi} & -I_{zyi} \\ -I_{zxi} & -I_{zyi} & I_{zzi} \end{pmatrix} \quad i = 1, 2 \quad (5.33)$$

De donde el segundo término de la sumatoria en (5.27) está dado por

$$\omega_1^T I_1 \omega_1 = I_{zz1} \dot{q}_1^2 \quad (5.34)$$

$$\omega_2^T I_2 \omega_2 = I_{zz2} (\dot{q}_1^2 + 2\dot{q}_1 \dot{q}_2 + \dot{q}_2^2) \quad (5.35)$$

Se sustituyen las ecuaciones (5.30), (5.31), (5.32) y (5.34) en (5.35) para obtener la energía cinética del sistema

$$K = \frac{1}{2} \begin{bmatrix} \dot{q}_1 \\ \dot{q}_2 \end{bmatrix}^T \begin{bmatrix} D_{11} & D_{12} \\ D_{21} & D_{22} \end{bmatrix} \begin{bmatrix} \dot{q}_1 \\ \dot{q}_2 \end{bmatrix} \quad (5.36)$$

$$= \frac{1}{2} (D_{11} \dot{q}_1^2 + 2D_{12} \dot{q}_1 \dot{q}_2 + D_{22} \dot{q}_2^2) \quad (5.37)$$

donde:

$$D_{11} = m_1 l_{c1}^2 + m_2 (l_1^2 + l_{c2}^2 + 2l_1 l_{c2} \cos(q_2)) + I_{zz1} + I_{zz2} \quad (5.38)$$

$$D_{12} = m_2 (l_{c2}^2 + l_1 l_{c2} \cos(q_2)) + I_{zz2}$$

$$D_{21} = D_{12}$$

$$D_{22} = m_2 l_{c2}^2 + I_{zz2}$$

La energía potencial del sistema está dada por

$$V = \sum_{i=1}^2 m_i g h_i \quad (5.39)$$

donde  $m_i$  es la masa asociada al  $i$ -ésimo eslabón y  $h_i$  es la distancia que existe entre el centro de masa y el eje coordenado horizontal del  $i$ -ésimo eslabón.

De la figura 5.1 se tiene

$$\begin{aligned} h_1 &= l_{c1} \text{sen}(q_1) \\ h_2 &= l_1 \text{sen}(q_1) + l_{c2} \text{sen}(q_1 + q_2) \end{aligned} \quad (5.40)$$

Entonces, sustituyendo(5.40) en (5.39) se llega a

$$V = m_1 g l_{c1} \text{sen}(q_1) + m_2 g (l_1 \text{sen}(q_1) + l_{c2} \text{sen}(q_1 + q_2)) \quad (5.41)$$

Sustituyendo las ecuaciones (5.37) y(5.41) en (5.26) se obtiene el Lagrangiano

$$L = \frac{1}{2} (D_{11} \dot{q}_1^2 + 2D_{12} \dot{q}_1 \dot{q}_2 + D_{22} \dot{q}_2^2) - m_1 g l_{c1} \text{sen}(q_1) - m_2 g (l_1 \text{sen}(q_1) + l_{c2} \text{sen}(q_1 + q_2)) \quad (5.42)$$

Una vez conocidos la energía cinética, la potencial y el Lagrangiano, se procede a hallar cada uno de los términos en (5.25) para  $i = 1, 2$

$$\frac{\partial L}{\partial \dot{q}_1} = D_{11} \dot{q}_1 + D_{12} \dot{q}_2 \quad (5.43)$$

$$\begin{aligned} &= (m_1 l_{c1}^2 + m_2 (l_1^2 + l_{c2}^2 + 2l_1 l_{c2} \cos(q_2)) + I_{zz1} + I_{zz2}) \dot{q}_1 \\ &\quad + (m_2 (l_{c2}^2 + l_1 l_{c2} \cos(q_2)) + I_{zz2}) \dot{q}_2 \end{aligned}$$

$$\frac{d}{dt} \left( \frac{\partial L}{\partial \dot{q}_1} \right) = D_{11} \ddot{q}_1 + D_{12} \ddot{q}_2 - 2m_2 l_1 l_{c2} \dot{q}_1 \dot{q}_2 \text{sen}(q_2) - m_2 l_1 l_{c2} \dot{q}_2^2 \text{sen}(q_2) \quad (5.44)$$

$$\frac{\partial L}{\partial q_1} = -m_1 g l_{c1} \cos(q_1) - m_2 g (l_1 \cos(q_1) + l_{c2} \cos(q_1 + q_2)) \quad (5.45)$$

$$\frac{\partial L}{\partial \dot{q}_2} = D_{12} \dot{q}_1 + D_{22} \dot{q}_2 \quad (5.46)$$

$$= (m_2 (l_{c2}^2 + l_1 l_{c2} \cos(q_2)) + I_{zz2}) \dot{q}_1 + (m_2 l_{c2}^2 + I_{zz2}) \dot{q}_2$$

$$\frac{d}{dt} \left( \frac{\partial L}{\partial \dot{q}_2} \right) = D_{12} \ddot{q}_1 + D_{22} \ddot{q}_2 - m_2 l_1 l_{c2} \dot{q}_1 \dot{q}_2 \text{sen}(q_2) \quad (5.47)$$

$$\frac{\partial L}{\partial q_2} = -m_2 l_1 l_{c2} \dot{q}_1^2 \text{sen}(q_2) - m_2 l_1 l_{c2} \dot{q}_1 \dot{q}_2 \text{sen}(q_2) - m_2 g l_{c2} \cos(q_1 + q_2) \quad (5.48)$$

Finalmente, sustituyendo los resultados obtenidos en (5.43) a (5.48) , se definen las ecuaciones dinámicas del manipulador como

$$\begin{aligned} \tau_1 &= ((m_1 l_{c1}^2 + m_2 (l_1^2 + l_{c2}^2 + 2l_1 l_{c2} \cos(q_2)) + I_{zz1} + I_{zz2}) \ddot{q}_1) \\ &\quad + (m_2 (l_{c2}^2 + l_1 l_{c2} \cos(q_2)) + I_{zz2}) \ddot{q}_2 - 2m_2 l_1 l_{c2} \dot{q}_1 \dot{q}_2 \text{sen}(q_2) \\ &\quad - m_2 l_1 l_{c2} \dot{q}_2^2 \text{sen}(q_2) + m_1 g l_{c1} \cos(q_1) + m_2 g (l_1 \cos(q_1) + l_{c2} \cos(q_1 + q_2)) \end{aligned} \quad (5.49)$$

$$\begin{aligned} \tau_2 &= (m_2 (l_{c2}^2 + l_1 l_{c2} \cos(q_2)) + I_{zz2}) \ddot{q}_1 + (m_2 l_{c2}^2 + I_{zz2}) \ddot{q}_2 \\ &\quad + m_2 l_1 l_{c2} \dot{q}_1^2 \text{sen}(q_2) + m_2 g l_{c2} \cos(q_1 + q_2) \end{aligned} \quad (5.50)$$

las cuales se pueden escribir en forma matricial

$$D(q) \ddot{q} + C(q, \dot{q}) \dot{q} + G(q) = \tau \quad (5.51)$$

donde

$$\begin{aligned}
 q &= [q_1 \quad q_2]^T & \dot{q} &= [\dot{q}_1 \quad \dot{q}_2]^T & \ddot{q} &= [\ddot{q}_1 \quad \ddot{q}_2]^T \\
 D(q) &= \begin{bmatrix} D_{11} & D_{12} \\ D_{21} & D_{22} \end{bmatrix} \\
 D_{11} &= m_1 l_{c1}^2 + m_2 (l_1^2 + l_{c2}^2 + 2l_1 l_{c2} \cos(q_2)) + I_{zz1} + I_{zz2} \\
 D_{12} &= m_2 (l_{c2}^2 + l_1 l_{c2} \cos(q_2)) + I_{zz2} \\
 D_{21} &= D_{12} \\
 D_{22} &= m_2 l_{c2}^2 + I_{zz2} \\
 C(q, \dot{q}) &= \begin{bmatrix} -2m_2 l_1 l_{c2} \dot{q}_2 \sin(q_2) & -m_2 l_1 l_{c2} \dot{q}_2 \sin(q_2) \\ m_2 l_1 l_{c2} \dot{q}_1 \sin(q_2) & 0 \end{bmatrix} \\
 G(q) &= \begin{bmatrix} m_1 g l_{c1} \cos(q_1) + m_2 g (l_1 \cos(q_1) + l_{c2} \cos(q_1 + q_2)) \\ m_2 g l_{c2} \cos(q_1 + q_2) \end{bmatrix} \\
 \tau &= \begin{bmatrix} \tau_1 \\ \tau_2 \end{bmatrix}
 \end{aligned} \tag{5.52}$$

$D(q)$  es la matriz de inercia simétrica y definida positiva,  $C(q, \dot{q})$  es la matriz de Coriolis y  $G(q)$  es el vector de gravedad.

### 5.2.3 Resultados en simulación utilizando Matlab.

Con el propósito de aplicar el nuevo esquema de control propuesto, se considera el problema de seguimiento de trayectorias para un manipulador de dos grados de libertad. Retomamos de nuevo el modelo obtenido en la sección previa, el cual corresponde a un manipulador de dos grados de libertad completamente actuado y cuyo modelo está dado por (5.51)

$$D(q) \ddot{q} + C(q, \dot{q}) \dot{q} + G(q) = \tau$$

donde los términos de (5.51) se definen en (5.52)

Ahora, consideremos el modelo de manipulador robótico presentado en [11] y que tomaremos como la planta no lineal desconocida; además consideremos la trayectoria de referencia dada por

$$q_r = \begin{bmatrix} 1.57 \\ 1.57 \end{bmatrix} + \begin{bmatrix} 0.78 [1 - e^{-2t^3}] + 0.17 [1 - e^{-2t^3}] \sin(w_1 t) \\ 1.04 [1 - e^{-1.8t^3}] + 2.18 [1 - e^{-1.8t^3}] \sin(w_2 t) \end{bmatrix} \tag{5.53}$$

donde  $w_1$  y  $w_2$  son las frecuencias de las trayectorias deseadas para cada brazo. Esta trayectoria de referencia requiere de grandes velocidades y aceleraciones para cada eslabón cuando  $w_1 = 15$  rad/s y  $w_2 = 3.5$  rad/s. Seleccionamos como condición inicial  $q_1 = -90^\circ$  y  $q_2 = 0^\circ$  la cual corresponde al punto de equilibrio inferior.



Los parámetros del robot utilizado en [11] son los siguientes,

$$\begin{aligned}
 m_1 &= 23.902 \text{ kg.} \\
 m_2 &= 1.285 \text{ kg.} \\
 l_1 &= 0.45 \text{ m.} \\
 l_2 &= 0.45 \text{ m.} \\
 l_{c1} &= 0.091 \text{ m.} \\
 l_{c2} &= 0.048 \text{ m.} \\
 I_1 &= 1.226 \text{ kg.m}^2 \\
 I_2 &= 0.093 \text{ kg.m}^2 \\
 g &= 9.81 \text{ m/s}^2
 \end{aligned}$$

Se utilizó la siguiente red neuronal dinámica para modelar el sistema

$$\dot{x}_p = Ax + W^* \Gamma z(x) + (x - x_p) + \Omega u \quad (5.54)$$

con los parámetros

$$\begin{aligned}
 A &= \begin{pmatrix} -150 & 0 & 0 \\ 0 & -150 & 0 \\ 0 & 0 & -150 \end{pmatrix} & \Gamma &= 0.25 \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix} \\
 z(x) &= \begin{pmatrix} \tanh(kx_1) \\ \tanh(kx_2) \\ \tanh(kx_3) \end{pmatrix} & k &= 0.45, & \Omega &= \begin{pmatrix} 0 & 0 \\ 0 & 0 \\ 1 & 0 \\ 0 & 1 \end{pmatrix}
 \end{aligned}$$

En la ley de control (5.21) seleccionamos  $\mu = 85$ .

En las figuras ?? a ?? se muestran los modelos implementados en Simulink para la simulación del control del manipulador. En el apéndice 1 se incluyen los programas en Matlab y los diagramas de bloques en Simulink utilizados para calcular la acción de control obtenida con el control óptimo inverso.

La evolución en el tiempo de los ángulos de cada uno de los eslabones se muestra en las figuras 5.2 a 5.4. Los torques de control y la perturbación de torque se muestran en las figuras 5.5 y 5.6. El error máximo en el primer eslabón es de  $1.2^\circ$  (4% de la amplitud máxima de la señal de referencia) y en el segundo eslabón menor a  $3^\circ$  (2% de la amplitud máxima de la referencia) y se desempeña satisfactoriamente en presencia de perturbaciones de torque.

Se realizaron también pruebas con perturbaciones de tipo senoidal con frecuencia variable entre 1 y 5 Hz, y amplitud de 10 N m; las figuras 5.7 y 5.8 muestran que se rechaza también este tipo de perturbación. El esquema propuesto de control ha demostrado ser robusto en presencia de dinámicas no modeladas y perturbaciones.

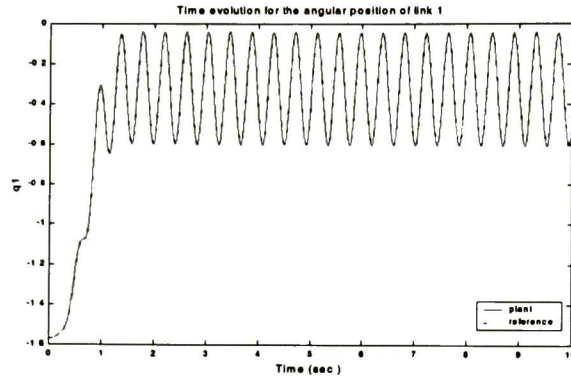


Figura 5.2: Evolución en el tiempo para la posición del eslabón 1

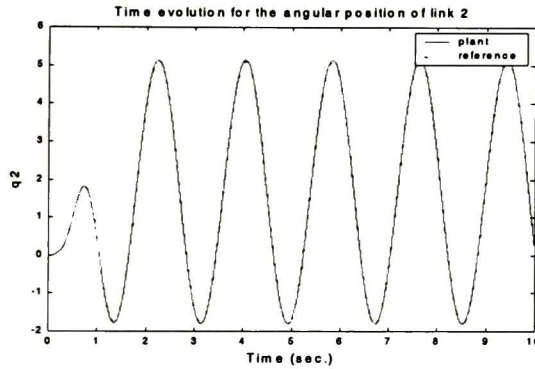


Figura 5.3: Evolución en el tiempo para la posición del eslabón 2

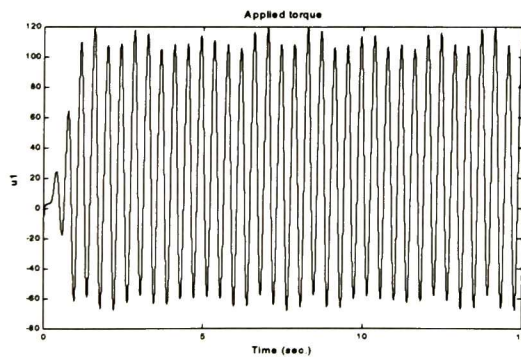


Figura 5.4: Torque aplicado al eslabón 1

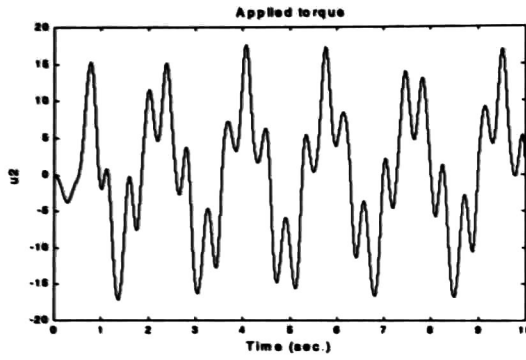


Figura 5.5: Torque aplicado al eslabón 2

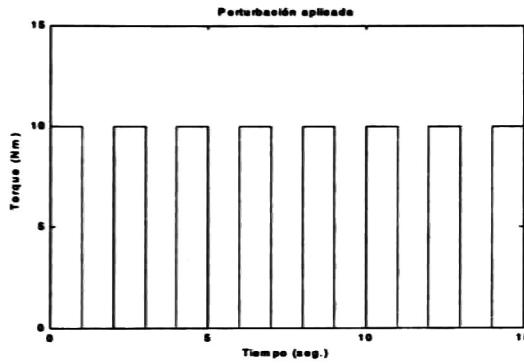


Figura 5.6: Perturbación de torque aplicada al primer eslabón

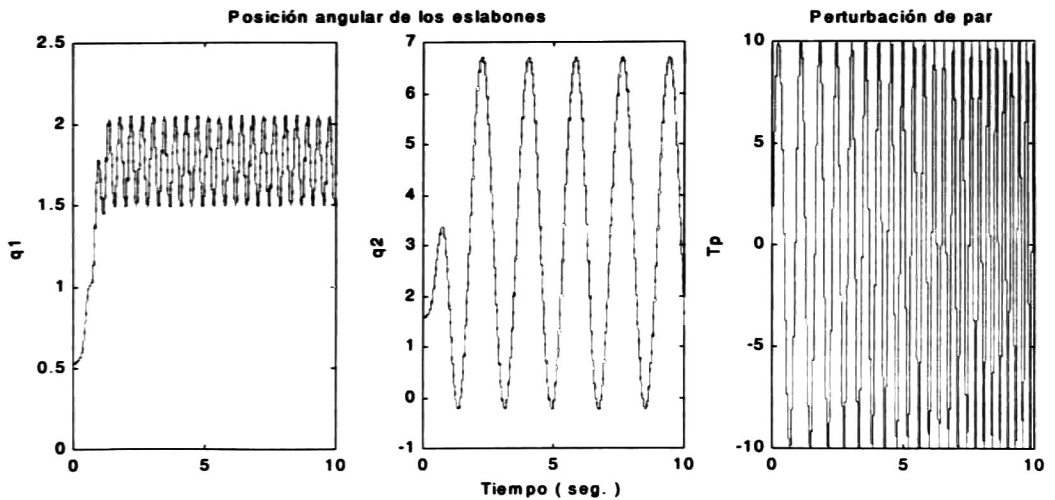


Figura 5.7: Posición angular de los eslabones 1,2 y perturbación de par de tipo senoidal con frecuencia variable para el manipulador completamente actuado.

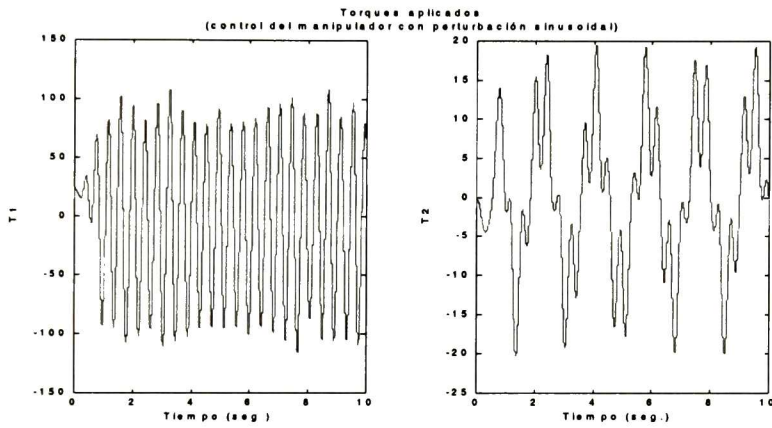


Figura 5.8: Torques aplicados cuando se tiene una perturbación senoidal de frecuencia variable.



# Capítulo 6

## Conclusiones y trabajo futuro

### 6.1 Conclusiones

Se ha desarrollado una nueva estructura adaptable basada en redes neuronales dinámicas para seguimiento de trayectorias en sistemas no lineales desconocidos. Esta estructura está compuesta de un identificador neuronal y una ley de control para seguimiento de trayectorias. El identificador neuronal contruye un modelo en línea de la planta no lineal desconocida y usa una ley de aprendizaje que minimiza el error de modelado que es considerado como la incertidumbre del sistema. El problema de seguimiento de trayectorias con incertidumbres se reduce a un problema de estabilización de la dinámica del error sin incertidumbres. La estabilidad de la dinámica del error ha sido establecida por medio de métodos de funciones de Lyapunov de control. La ley de control obtenida asegura seguimiento de trayectorias tanto de la red neuronal como de la planta desconocida.

Posteriormente se extienden los resultados para obtener una ley de control que sea óptima con respecto a un funcional de costo bien definido. Se utilizó el método del control óptimo inverso basado en conceptos de estabilidad entrada-estado para aliviar el problema de resolver el sistema de ecuaciones diferenciales parciales de Hamilton-Jacobi-Bellman, asociado con el funcional de costo del método  $H_\infty$ . Usando el control óptimo inverso se obtiene una ley de control óptima que estabiliza globalmente asintóticamente la dinámica del error y asegura el seguimiento de trayectorias tanto de la red neuronal como la planta.

En el capítulo cuarto se aplican los métodos desarrollados para controlar sistemas no lineales complejos. Usando el esquema compuesto del identificador neuronal (3.2) la ley de adaptación de pesos que reducen el error de modelado y las leyes de control obtenidas mediante estabilización cuadrática y el control óptimo inverso, se estabilizaron los sistemas caóticos de Roessler y Chua en un punto de equilibrio.

Usando el mismo esquema se produce caos en una planta desconocida lineal, y se fuerza ésta a seguir una trayectoria caótica dada por los sistemas de Chua y Chen. Posteriormente, se presenta una aplicación más compleja donde se presenta la sincronización de caos, tomando como planta no lineal desconocida un sistema caótico el cual es modelado por la red neuronal dinámica y tomando como trayectoria de referencia otro sistema caótico.

Los resultados obtenidos son motivantes; el esquema propuesto es robusto ante perturbaciones en línea e incertidumbres en el modelado. La flexibilidad que da el uso del identificador neuronal lo hace adecuado para una variedad de sistemas sin importar lo complejo de éstos. Los esquemas propuestos presentan una rápida respuesta y estabilización global asintótica

de la dinámica del error

En el capítulo quinto se extendieron los resultados obtenidos en el capítulo 3 en el control adaptable para seguimiento de trayectorias, basándonos en el método del control óptimo inverso. Se realizó el análisis para el caso en el que se tengan menos entradas que estados. Se utilizó el modelo de un robot de dos grados de libertad para verificar el problema de seguimiento de trayectorias. Se consideró el modelo del manipulador desconocido y se construyó un modelo en línea utilizando redes neuronales dinámicas. Se presentaron los resultados en simulación hechos en Matlab. Se obtuvo un desempeño satisfactorio en el seguimiento así como una acción de control suave y acotada en relación a los resultados presentados en [11]. El esquema de control es robusto ante las perturbaciones de par aplicadas en simulación.

El presente trabajo aporta una nueva estructura de control que unifica el control adaptable y el uso de redes neuronales como identificadores; este esquema de control aporta las ventajas de robustez ante incertidumbres en el modelado y una gran rapidez de adaptación. Este esquema se desempeñó satisfactoriamente en el caso en el que se tienen el mismo número de entradas que de estados y en sistemas con estructura triangular como es el caso de los manipuladores robóticos.

Se tiene la limitante en el caso en el que no se tengan sistemas triangulares y se desee realizar el control con menos entradas que estados, donde el algoritmo de control no mostró un desempeño satisfactorio; como trabajo futuro se pretende solucionar esta limitante.

## 6.2 Trabajo Futuro

Se pretende obtener resultados más generales para el caso en el que se tengan menos entradas que estados, considerando que el sistema no lineal desconocido no posea una estructura triangular.

Implementar en tiempo real el control para seguimiento de trayectoria del manipulador robótico y realizar pruebas en laboratorio para verificar la robustez de los algoritmos desarrollados.

Extender los resultados en control adaptable usando el método del control óptimo inverso para el caso en el que se tenga saturación en el control.

# Bibliografía

- [1] T. Basar and P. Bernhard, *H-Infinity Optimal Control and Related Minimax Design Problems*, Birkhauser, Boston, USA, 1995.
- [2] G. Chen and X. Dong "From chaos to order Perspectives and methodologies in controlling chaotic nonlinear dynamical systems" , *International Journal of Bifurcation and Chaos*, Vol. 3, No. 6, pp. 1363-1409, April, 1993.
- [3] G. Chen and J. L. Moiola, " An overview of bifurcation, chaos and nonlinear dynamics in control systems" *The Franklin Institute Journal*, 1995.
- [4] L. Dieci, R. D. Russell and E. S. Van Vleck, " On the computation of Lyapunov exponents for continuous dynamical systems" *Siam J. Numer. Anal.* vol. 34, no. 1 , pp. 402-423, February 1997.
- [5] N. H. El-Farra and P. D. Christofides "Robust near optimal output feedback control of non-linear systems", *International Journal of Control*, vol. 74, no. 2, pp. 133-157, 2001.
- [6] J. Guckenheimer and P. Holmes *Nonlinear oscilations, dynamical systems and bifurcations of vector fields*", Springer, New York, USA,1983.
- [7] M. M. Gupta and D. H. Rao (Eds.), *Neuro-Control Systems, Theory and Applications*, IEEE Press, Piscataway, N.J., USA; 1994.
- [8] D. J. Hill D.J. and P. Moylan "The Stability of nonlinear dissipative systems" . *IEEE Trans. on Auto. Contr.*, Vol. 21, 708-711, 1996.
- [9] K. Hunt, G. Irwin, and K. Warwick (Eds.), *Neural Networks Engineering in Dynamic Control Systems*, Springer Verlag, New York, USA, 1995.
- [10] A. Isidori, *Nonlinear Control Systems*, Third Edition, Springer Verlag, New York, USA, 1995.
- [11] M. A. Llama, R. Kelly and V. Santibañez, "Stable Computed torque of Robot manipulators via fuzzy self-tuning" *IEEE Trans. on Systems, Man. and Cybernetics*, Vol. 30, 143-150, February 2000
- [12] H. Khalil, *Nonlinear Systems*, Second Edition, Prentice Hall, Upper Saddle River, New Jersey, USA, 1996.
- [13] H. W. Knobloch, A. Isidori A. and Flockerzi D., *Topics in Control Theory*, Birkhauser, Boston, 1993.



- [14] E. B. Kosmatopoulos et al., "Dynamical neural networks that ensure exponential identification error convergence" *Neural Networks*, Vol. 1, No. 2, pp. 299-314, 1997.
- [15] M. Krstic M. and H. Deng H., *Stabilization of Nonlinear Uncertain Systems*, Springer Verlag, New York, USA, 1998.
- [16] Z. H. Li and M. Krstic "Optimal design of adaptive tracking controllers for non-linear systems" *Automatica*, vol. 33, no. 8, pp. 1459-1473, 1997.
- [17] E.N. Lorenz "Deterministic non-periodic flow", *J. Atmos. Sci.* 20, 130-141, 1965.
- [18] R. Marino and P. Tomei, *Nonlinear Control Design, geometric, adaptive and robust*, Prentice Hall, Great Britain, 1995.
- [19] K. S. Narendra, and K. Parthasarathy, "Identification and control of dynamical systems using neural networks", *IEEE Trans. on Neural Networks*, vol. 1, no. 1, pp 4-27,1990.
- [20] T. S. Parker and L. O. Chua, "Chaos: A tutorial for engineers" *Proceedings of the IEEE*, Vol. 75, No. 8, August 1987.
- [21] A. S. Poznyak, W. Yu, E. N. Sanchez, and J. P. Perez, "Nonlinear adaptive trajectory tracking using dynamic neural networks" *IEEE Trans. on Neural Networks*, vol. 10, No 6, pp 1402-1411, Nov. 1999
- [22] K. Suykens, L. Vandewalle, and R. de Moor, *Artificial Neural Networks for Modelling and Control of Nonlinear Systems*, Kluwer Academic Publishers, Boston, USA, 1996.
- [23] G. A. Rovitahkis and M. A. Christodoulou, *Adaptive Control with Recurrent High-Order Neural Networks*, Springer Verlag, New York, USA, 2000.
- [24] E. N. Sanchez and J. P. Perez, "Input-to-state stability analysis for dynamic neural networks", *IEEE Trans. Circuits Syst. I*, vol. 46, 1395-1398, 1999.
- [25] E. N. Sanchez, J. P. Perez, and G. Chen, "Using Dynamic Neural Control to Generate Chaos: An Inverse Optimal Control Approach", *Intl. Journ. of Bifurcation and Chaos*, to appear, 2001.
- [26] E. N. Sanchez, J. P. Perez , M. Martínez and G. Chen , "Global asymptotic stabilization of Chen's chaotic system via inverse optimal control"
- [27] E. N. Sanchez , J. P. Perez, L. Ricalde and G. Chen, "Trajectory tracking via adaptive neural control", *Proceedings of IEEE Int. Symposium on Intelligent Control*, Mexico City, pp. 286-289, September 2001.
- [28] E. N. Sanchez J. P. Perez, L. Ricalde and G. Chen "Chaos production and synchronization via adaptive neural control", to appear in *Proceedings of IEEE Conference on Decision and Control*, Orlando, Fl, USA, December 4-7, 2001.
- [29] R. Sepulchre, M. Jankovic and P. V. Kokotovic, "*Constructive nonlinear control*" Springer, New york, USA, 1997.



- [30] A. Vanecek and S. Celikovsky, *Control Systems: from linear analysis to synthesis of chaos*, Prentice Hall, U. K., 1996.
- [31] S. Wiggins *Introduction to applied nonlinear dynamical systems and chaos*. Springer, New York, USA, 1990.

# APENDICE 1.

## Códigos en Matlab y diagramas en Simulink

Programa que calcula la acción de control para seguimiento de trayectorias del manipulador robótico.

```
%AccionControlRobot
%Programa que calcula la acción de control desarrollada por el método del control
%óptimo inverso
```

```
function [out]= AccionControlRobot(z)
%Parámetros de la red neuronal
% éstos se definen desde la línea de comandos
% (como variables globales)
global k lambda gama mu;
```

```
%Entradas al programa:
% W (matriz de pesos), xp (estado de la planta)
% xr (estado de la referencia), xi (estado de la red neuronal),
% fr(xr) (sistema dinámico de la referencia)
%Se tienen entonces 32 entradas
w11=z(1,1);
w12=z(2,1);
w13=z(3,1);
w14=z(4,1);
w21=z(5,1);
w22=z(6,1);
w23=z(7,1);
w24=z(8,1);
w31=z(9,1);
w32=z(10,1);
w33=z(11,1);
w34=z(12,1);
w41=z(13,1);
w42=z(14,1);
w43=z(15,1);
w44=z(16,1);
xp1=z(17,1); xp2=z(18,1); xp3=z(19,1); xp4=z(20,1);
```

```

xr1=z(21,1); xr2=z(22,1); xr3=z(23,1); xr4=z(24,1);
xi1=z(25,1); xi2=z(26,1); xi3=z(27,1); xi4=z(28,1);
frxr1=z(29,1);
frxr2=z(30,1);
frxr3=z(31,1);
frxr4=z(32,1);

%Se construyen la matriz de pesos y los vectores de estado
W=[ w11 w12 w13 w14; w21 w22 w23 w24; w31 w32 w33 w34; w41 w42 w43 w44];
xi=[xi1; xi2; xi3; xi4]; xp=[xp1; xp2; xp3; xp4]; xr=[xr1; xr2; xr3; xr4];
frxr=[frxr1; frxr2; frxr3; frxr4];

% Matriz OMEGA
OM= [ 0 0; 0 0; 1 0; 0 1];

% Se halla la norma de W
norma= norm(W,1);

% Funciones de activación utilizadas
sigmar=[tanh(k*xr1); tanh(k*xr2); tanh(k*xr3); tanh(k*xr4)];
sigmaip=[tanh(k*xi1)-tanh(k*xp1); tanh(k*xi2)-tanh(k*xp2); tanh(k*xi3)-tanh(k*xp3);
tanh(k*xi4)-tanh(k*xp4)];

%Cálculo del error;
e1 = xp1-xr1; e2 = xp2-xr2; e3 = xp3-xr3; e4 = xp4-xr4;
e = [e1; e2; e3; e4];

%W*Gama*Sigma (término de la ley de control)
WGSig=gama*W*sigmar;
WGSigp=gama*W*sigmaip;

% Se divide u = u1 + u2 , donde
u1=-WGSigp-(lambda+1)*(xi-xp);

% Acción de control que minimiza el funcional de costo
u2=-(\mu)*(1+(gama*norma)^2)*e;

% La acción de control finalmente será
% u = u1 + u2 + alfa
% de último se multiplica por la pseudoinversa, aunque para la aplicación
% ésta es la identidad, si se toma así se ahorra bastante tiempo de computo
pseudo = (OM'*OM)OM'
control= pseudo*(u1+u2+(frxr-lambda*xr-WGSig-(xr-xp)));
out=[ control ];

```

Programa que evalúa la dinámica del manipulador robótico de dos grados de libertad completamente actuado.

```
%Bloque del modelo no lineal del pendubot. Recibe como parámetros
%la derivada del ángulo theta1, el ángulo theta1 y la entrada u1
%que corresponden al primer eslabón, el ángulo theta2 y la derivada
%del ángulo theta2, en un vector 5X1 en el orden que se
%menciona la salida que da este bloque es un vector de 2X1
%con la segunda derivada del ángulo theta1 y la segunda derivada
%del vector theta2
function y = ecuacionrobotkelly(xin)

%separación de las variables de entrada
dtheta1 = xin(1,1);
theta1 = xin(2,1);
u1 = xin(3,1);
u2 = xin(4,1);
theta2 = xin(5,1);
dtheta2 = xin(6,1);
pert = xin(7,1);

%valores de masa, longitud y gravedad utilizados
m1 = 23.902;
m2 = 1.285 ;
l1 = 0.45;
l2 = 0.45;
lc1 = 0.091;
lc2 = 0.048;
Izz1 = 1.226;
Izz2 = 0.093;
g = 9.81;
fric1 = 0;
fric2 = 0;

%Creacion de las matrices

%Matriz de inercias
D11 = m1*lc1*lc1 + m2*(l1*l1 + lc2*lc2 + 2*l1*lc2*cos(theta2)) + Izz1 + Izz2;
D12 = m2*(lc2*lc2 + l1*lc2*cos(theta2)) + Izz2;
D22 = m2*lc2*lc2 + Izz2;
DM = [D11,D12;D12,D22];

%Matriz de Coiolis-Centripeta
C11 = -2*m2*l1*lc2*dtheta2*sin(theta2);
C12 = -m2*l1*lc2*dtheta2*sin(theta2);
C21 = m2*l1*lc2*dtheta1*sin(theta2);
C22 = 0;
```



```
CM = [C11,C12;C21,C22];
```

```
%Vector que contiene el efecto de la gravedad
```

```
G11 = m1*g*lc1*cos(theta1) + m2*g*l1*cos(theta1) + m2*g*lc2*cos(theta1 + theta2);
```

```
G21 = m2*g*lc2*cos(theta1 + theta2);
```

```
GM = [G11;G21];
```

```
%Vector de ángulos derivados
```

```
dthetaM = [dtheta1;dtheta2];
```

```
%Vector de entrada
```

```
TM = [u1+pert;u2];
```

```
%Calculo de la salida
```

```
y = (DM)^(-1)*(TM - (CM*dthetaM + GM ));
```

# Diagramas de bloques en Simulink para la simulación del control del manipulador robótico.

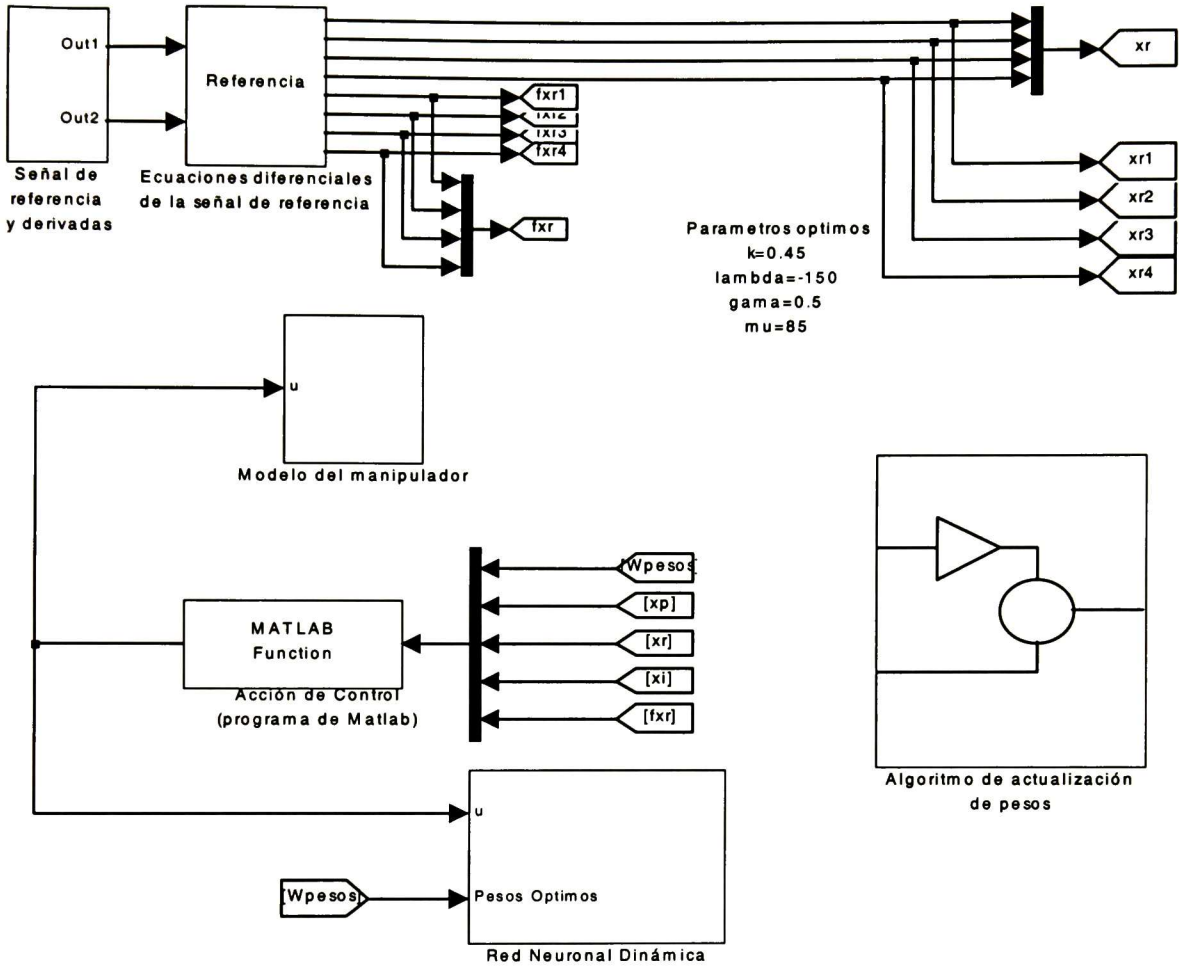


Figura 6.1: Esquema de simulación en Simulink para el control neuronal adaptable para el manipulador robótico

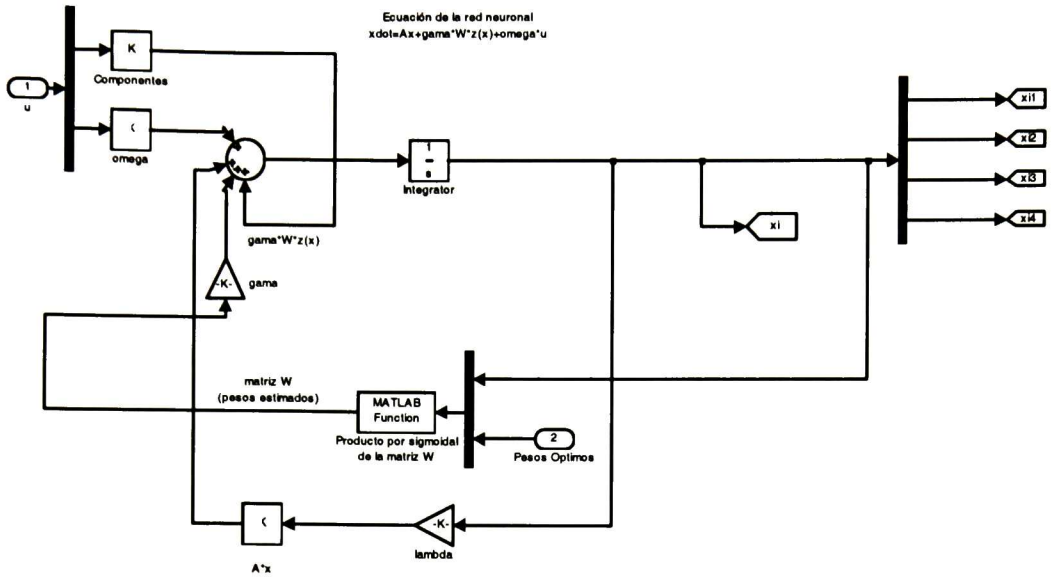


Figura 6.2: Diagrama de bloques en Simulink de la red neuronal

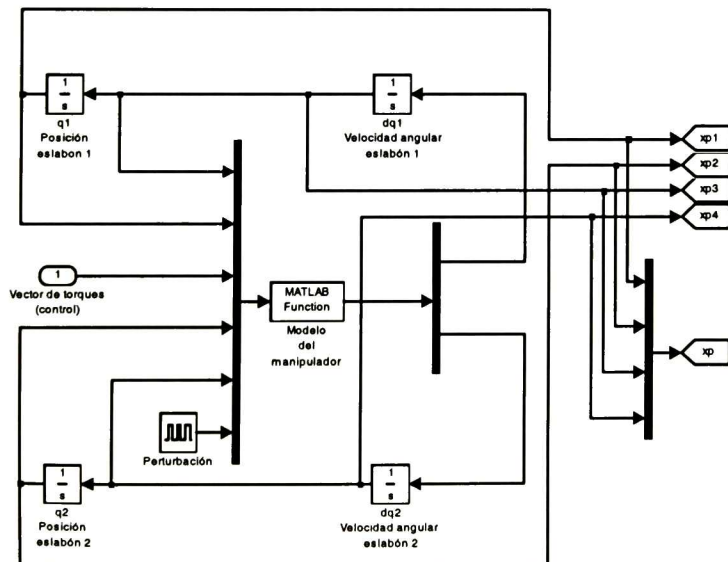


Figura 6.3: Diagrama de bloques en Simulink del modelo del manipulador robótico

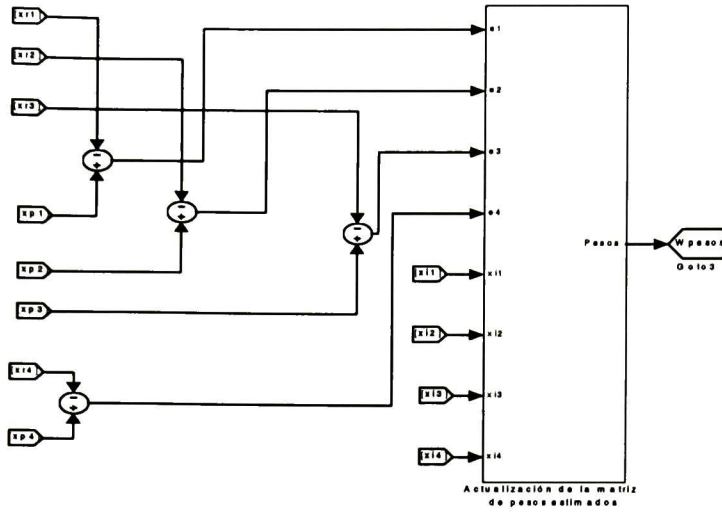


Figura 6.4: Ley de adaptación de pesos (a)

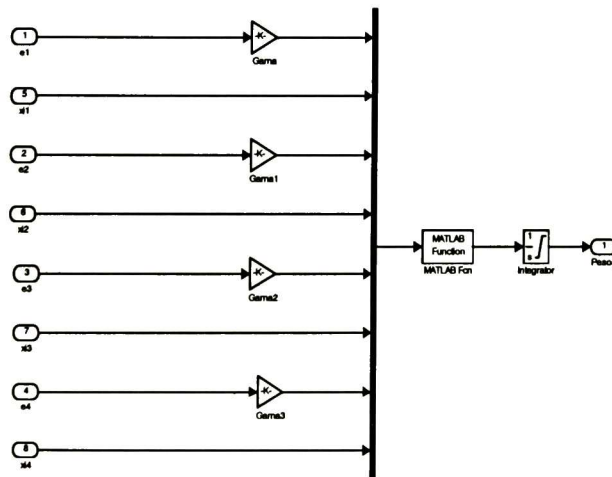


Figura 6.5: Ley de adaptación de pesos (b)



# APENDICE 2.

## Artículos enviados

**“Chaos production and synchronization via adaptive neural control”**. Artículo presentado en *Conference on Decision and Control*, Orlando, Fl, USA, Diciembre 4-7, 2001.

**”Recurrent Neural Control for Robot Trajectory Tracking”**. Artículo enviado para evaluación al IFAC World Congress 2001



Conference Editorial Board  
Control Systems Society



From: Professor J. Jim Zhu - CDC 2001  
353 Stocker Center, School of EECS  
Ohio University  
Athens, OH 45701, USA

Email: [author-support@cssceb.ece.ohiou.edu](mailto:author-support@cssceb.ece.ohiou.edu)  
Fax: +1 740 597 1519  
Voice: +1 740 593-9136  
Internet: <http://cosmo.ece.ohiou.edu/cssceb>

July 22, 2001

To: Dr. Edgar N. Sanchez  
Unidad Guadalajara  
CINVESTAV  
Plaza La Luna - Apto. Postal 31-438  
Guadalajara, JALISCO C.P. 45091  
MEXICO

PIN: 12059  
Phone: +52.36841580  
Fax: +52.36841708  
Email: [sanchez@gdl.cinvestav.mx](mailto:sanchez@gdl.cinvestav.mx)

Re: Paper ID **CDC01-INV1203**      *Session-Code* **ThP07-3**

Paper Title **Chaos synchronization via adaptive recurrent neural control**

Paper Accepted as **Invited, Regular**      *Presentation* **ORAL**

Authors (Surname, First Name; Institute [PIN]):

**Sanchez, Edgar N.; CINVESTAV [PIN: 12059]**  
**Perez, Jose P.; Cinvestav [PIN: 26846]**  
**Ricalde, Luis J.; CINVESTAV [PIN: 35868]**  
**Chen, Guanrong Ron; Univ. of Houston [PIN: 01041]**

Dear Dr. Sanchez:

This concerns your manuscript CDC01-INV1203 that was submitted to the 40th IEEE Conference on Decision and Control (CDC01). As indicated in the enclosed letter from the CDC01 Program Chair, your paper was accepted as a REGULAR INVITED paper for ORAL presentation in Session ThP07-3 at CDC01.

The Author Kit can be found at the CDC01 web site:

<http://www.ecs.umass.edu/cdc2001/>

Please contact your session organizer to see if there are any useful comments from the reviewers. I urge you to make every effort to address these comments in your final revision. If you need to change the title or authorship from the data listed above, do so on the Paper Certification and Abstract Form found in the Author Kit.

Thank you for your contribution to the 40th Conference on Decision and Control.

Sincerely,

J. Jim Zhu, Chair  
Conference Editorial Board

# CHAOS SYNCHRONIZATION VIA ADAPTIVE RECURRENT NEURAL CONTROL

Edgar N. Sanchez<sup>1</sup>, Jose P. Perez<sup>2</sup>, Luis J. Ricalde<sup>1</sup> and Guanrong Chen<sup>3</sup>

1) CINVESTAV, Unidad Guadalajara, Apartado Postal 31-430, Plaza La Luna, Guadalajara, Jalisco C.P. 45091, Mexico, e-mail: sanchez@gdl.cinvestav.mx

2) CINVESTAV, Unidad Guadalajara, on doctoral studies leave from, Universidad Autonoma de Nuevo Leon, Monterrey, Nuevo Leon, Mexico

3) University of Houston, Houston, Texas, USA, and City University of Hong Kong

## Abstract

This paper proposes a new adaptive control structure, based on a dynamic neural network, for trajectory tracking of unknown nonlinear plants. The main components of this structure include a neural identifier and a control law, which together guarantee the desired trajectory tracking performance. Stability of the tracking control is analyzed by using the Lyapunov function method, and the structure is tested by simulations on an example of complex dynamical systems: chaos synchronization

*keywords* - Dynamic neural networks, chaos production, chaos synchronization, adaptive control, Lyapunov function, stability.

## I. INTRODUCTION

Since the seminal paper [1], there has been continuously increasing interest in applying neural networks to identification and control of nonlinear systems. Most of these applications use feedforward structures [2], [3]. Lately, the use of dynamic neural networks is developed, which allows more efficient modeling of the underlying dynamical systems [4]. Two recent books [5], [6] have reviewed the application of dynamic neural networks for nonlinear system identification and control. In [5], in particular, off-line learning is used, while [6] analyzes adaptive identification and control by means of on-line learning, where stability of the closed-loop system is established based on the Lyapunov function method. In [6], the trajectory tracking problem is reduced to a linear model following problem, with an application to DC electric motors.

On the other hand, control methods that are applicable to general nonlinear systems have been extensively developed since the early 1980's, for example based on the differential geometry theory [7]. Recently, the passivity approach has generated some increasing interest for synthesizing control laws for general nonlinear systems [8]. An important problem in this approach is how to achieve robust nonlinear control in the presence of unmodelled dynamics and external disturbances. Along this line there is the so-called  $H_\infty$  nonlinear control approach [9], [10]. One major difficulty with this approach, alongside its possible system structural instability, seems to be the requirement of solving the associated partial differential equations. In order to alleviate this computational problem, the so-called inverse optimal control technique was recently suggested, based on the input-to-state stability concept [11].

Using the inverse optimal control approach, a control law, capable of producing chaos on a dynamic neural network, was discussed in [12]. The present paper further extends these results to the adaptive trajectory tracking of chaotic

orbits in unknown dynamical systems. The proposed new adaptive control scheme is composed of a dynamic neural identifier, which builds an on-line model for the unknown plant, and a control law, which ensures that the unknown plant tracks the reference trajectory satisfactorily. The new approach is illustrated by an example of complex dynamical systems: chaos synchronization.

## II. MATHEMATICAL PRELIMINARIES

The dynamic neural network to be used is first introduced.

### A. Recurrent Higher-Order Neural Networks

In [13], recurrent higher-order neural networks (RHONN) are defined as

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

where  $x_i$  is the  $i$ th neuron state,  $L$  is the number of higher-order connections,  $\{I_1, I_2, \dots, I_L\}$  is a collection of non-ordered subsets of  $\{1, 2, \dots, m+n\}$ ,  $a_i > 0$ ,  $w_{ik}$  are the adjustable weights of the neural network,  $d_j(k)$  are nonnegative integers, and  $y$  is a vector defined by  $y = [y_1, \dots, y_n, y_{n+1}, \dots, y_{n+m}]^\top = [S(x_1), \dots, S(x_n), S(u_1), \dots, S(u_m)]^\top$ , with  $u = [u_1, u_2, \dots, u_m]^\top$  being the input to the neural network, and  $S(\cdot)$  a smooth sigmoid function formulated by  $S(x) = \frac{1}{1 + \exp(-\beta x)} + \varepsilon$ . For the sigmoid function,  $\beta$  is a positive constant and  $\varepsilon$  is a small positive real number. Hence,  $S(x) \in [\varepsilon, \varepsilon + 1]$ .

As can be seen, (1) allows the inclusion of higher-order terms.

By defining a vector  $z(x, u) = [z_1(x, u), \dots, z_L(x, u)]^\top = [\prod_{j \in I_1} y_j^{d_j(1)}, \dots, \prod_{j \in I_L} y_j^{d_j(L)}]^\top$ , (1) can be rewritten as

$$\dot{x}_i = -a_i x_i + \sum_{k=1}^L w_{ik} z_k(x, u), \quad i = 1, \dots, n$$

or

$$\dot{x}_i = -a_i x_i + w_i^\top z_i(x, u), \quad i = 1, \dots, n \quad (2)$$

where  $w_i = [w_{i,1} \dots w_{i,L}]^\top$

In this paper, such terms as  $y = [y_1, \dots, y_n, y_{n+1}, \dots, y_{n+m}]^\top = [S(x_1), \dots, S(x_n), u_1, \dots, u_n]^\top$  are considered. This means that the same number of inputs and states is used. It is also assumed that the input to the neural network enters directly, so that (1) can be rewritten as

$$\dot{x}_i = -a_i x_i + w_i^\top z_i(x) + u_i, \quad i = 1, \dots, n \quad (3)$$

Reformulating (2) in the matrix form yields

$$\dot{x} = Ax + Wz(x) + u \quad (4)$$

where  $x \in \mathbb{R}^n$ ,  $W^* \in \mathbb{R}^{n \times L}$ ,  $z(x) \in \mathbb{R}^L$ ,  $u, \in \mathbb{R}^n$ , and  $A = -\lambda I, \lambda > 0$ .

In this paper, a slight modification of the RHONN is used:

$$\dot{x} = Ax + W\Gamma z(x) + u$$

where  $\Gamma \in \mathbb{R}^{L \times L}$  is a diagonal matrix defined as

$$\Gamma = \begin{pmatrix} \gamma_1 & 0 & & 0 \\ 0 & \gamma_2 & & 0 \\ & & & \\ 0 & 0 & \gamma_{L-1} & 0 \\ & & 0 & \gamma_L \end{pmatrix} \quad \gamma_i > 0, \quad i = 1, \dots, L$$

These  $\gamma$  parameters play the role of the step size for the gradient-descent algorithm introduced below.

### B. Plant Modelling

The unknown nonlinear plant is modeled as

$$\dot{x}_p = F_p(x_p, u) \triangleq f_p(x_p) + g_p(x_p)u \quad (5)$$

where  $x_p, f_p \in \mathbb{R}^n$ ,  $g_p \in \mathbb{R}^{n \times n}$ ,  $u \in \mathbb{R}^n$ .

Taking into account that both  $f_p$  and  $g_p$  are unknown, one may try to modelize (5) by a dynamic neural networks in the form of (4). Henceforth the plant model becomes

$$\dot{x}_p = \dot{x} + w_{per} = Ax + W^* \Gamma z(x) + w_{per} + u \quad (6)$$

where the term  $w_{per}$  represents the modelling error, and  $W^*$  are the unknown values of the network weights, which minimize the modelling error. Here,  $w_{per} := x - x_p$ .

### III. ORBIT-TRACKING ANALYSIS

The orbit-tracking error is defined to be the difference between the output of the unknown plant modeled by (6) and the reference orbit defined by

$$\dot{x}_r = f_r(x_r, u_r), \quad x_r \in \mathbb{R}^n \quad (7)$$

namely,

$$e := x_p - x_r \quad (8)$$

The time derivative of the tracking error is

$$\dot{e} = \dot{x}_p - \dot{x}_r = Ax + W^* \Gamma z(x) + w_{per} + u - f_r(x_r, u_r) \quad (9)$$

Now, by adding and substrating to the right-hand side of (9) the terms  $\hat{W} \Gamma z(x_r)$ ,  $\alpha_r(t, \hat{W})$ , and  $Ae$ , and by taking into account  $w_{per} = x - x_p$ , one obtains

$$\begin{aligned} \dot{e} &= Ax + W^* z(x) + x - x_p + u - f_r(x_r, u_r) \pm \hat{W} \Gamma z(x_r) \pm \alpha_r(t, \hat{W}) \pm Ae \\ &= Ae + W^* \Gamma z(x) + u + \left( -f_r(x_r, u_r) + \hat{W} \Gamma z(x_r) + x - x_p + Ax - Ae + \alpha_r(t, \hat{W}) \right) \\ &\quad - \hat{W} \Gamma z(x_r) - \alpha_r(t, \hat{W}) \\ &= Ae + W^* \Gamma z(x) + u + \left( -f_r(x_r, u_r) + Ax_r + \hat{W} \Gamma z(x_r) + (A + I) * (x - x_p) + \alpha_r(t, \hat{W}) \right) \\ &\quad - \hat{W} \Gamma z(x_r) - \alpha_r(t, \hat{W}) \end{aligned} \quad (10)$$



where  $\hat{W}$  is an estimation of  $W^*$

Assuming that the term  $-f_r(x_r, u_r) + Ax_r + \hat{W}\Gamma z(x_r) + (A + I) * (x - x_p) + \alpha_r(t, \hat{W})$  is equal to zero, one has

$$Ax_r + \hat{W}\Gamma z(x_r) + (A + I) * (x - x_p) + \alpha_r(t, \hat{W}) = f_r(x_r, u_r) \quad (11)$$

and (10) is reduced to

$$\dot{e} = Ae + W^*\Gamma z(x) + u - \hat{W}\Gamma z(x_r) - \alpha_r(t, \hat{W}) \quad (12)$$

Adding and substrating to (12) the term  $\hat{W}\Gamma z(x)$  gives

$$\dot{e} = Ae + (W^* - \hat{W})\Gamma z(x) + \hat{W}\Gamma(z(x) - z(x_r)) + u - \alpha_r(t, \hat{W}) \quad (13)$$

Then, by defining

$$\tilde{W} = W^* - \hat{W}, \quad \tilde{u} = u - \alpha_r(t, \hat{W}), \quad (14)$$

and substituting (14) into (13), one obtains

$$\dot{e} = Ae + \tilde{W}\Gamma z(x) + \hat{W}\Gamma(z(x) - z(x_r)) + \tilde{u} \quad (15)$$

#### A. Orbit-Tracking Stabilization

Once (15) is obtained, one can proceed to study its stabilization. Notice that  $e = 0, \tilde{W} = 0$ , is an equilibrium point for the undisturbed system.

In order to perform the stability analysis, the following Lyapunov function is considered:

$$V = \frac{1}{2}\|e\|^2 + \frac{1}{2}tr\left\{\tilde{W}^T\tilde{W}\right\} \quad (16)$$

Its time derivative, along the orbits of (15), is

$$\dot{V} = e^T Ae + e^T \tilde{W}\Gamma z(x) + e^T \hat{W}\Gamma(z(x) - z(x_r)) + e^T \tilde{u} + tr\left\{\dot{\tilde{W}}^T \tilde{W}\right\} \quad (17)$$

As in [6], the following learning law is used:

$$tr\left\{\dot{\tilde{W}}^T \tilde{W}\right\} = -e^T \tilde{W}\Gamma z(x) \quad (18)$$

which can be written, element by element, as

$$\dot{w}_{i,j} = e_i(\gamma_j z(x_j)), \quad i = 1, 2, \dots, n, \quad j = 1, 2, \dots, L. \quad (19)$$

Putting (18) into (17) yields

$$\dot{V} = -\lambda e^T e + e^T \hat{W}\Gamma(z(x) - z(x_r)) + e^T \tilde{u} \quad (20)$$

Considering (20), the following control law is used:

$$\tilde{u} = -\hat{W}\Gamma(z(x) - z(x_r)) - ke \quad k \in \mathbb{R}^+ \quad (21)$$

Therefore,

$$\dot{V} = -(\lambda + k) \|e\|^2 < 0, \quad \forall e, \hat{W} \neq 0 \quad (22)$$

Hence,  $\lim_{t \rightarrow \infty} e(t) \rightarrow 0$

Thus, the control law  $u$  is obtained, which is fed to the neural network and the plant, as follows:

$$\begin{aligned} u &= \tilde{u} + \alpha_r(t, \hat{W}) \\ u &= -\hat{W}\Gamma(z(x) - z(x_r)) - \mu e + f_r(x_r, u_r) - Ax_r - \hat{W}\Gamma z(x_r) - (A + I) * (x - x_p) \end{aligned} \quad (23)$$

To this end, the results obtained can be summarized as follows.

*Theorem 1:* For the unknown nonlinear system modeled by (6), the on-line learning law (19) and the control law (23) together ensure the tracking to the nonlinear reference model (12).

#### IV. AN APPLICATION EXAMPLE

In order to demonstrate the applicability of the proposed adaptive control scheme, the following example is tested.

In this example, the unknown plant considered is Chen's chaotic attractor generated by

$$\begin{aligned} \dot{x}_{p1} &= a(x_{p2} - x_{p1}) \\ \dot{x}_{p2} &= (c - a)x_{p1} - x_{p1}x_{p2} + cx_{p2} \\ \dot{x}_{p3} &= x_{p1}x_{p2} - bx_{p3} \end{aligned} \quad (24)$$

with

$$a = 35, \quad b = 3, \quad c = 28 \quad \text{and} \quad \begin{pmatrix} x_{r1}(0) \\ x_{r2}(0) \\ x_{r3}(0) \end{pmatrix} = \begin{pmatrix} 1.5 \\ 1.5 \\ 1.5 \end{pmatrix}$$

The goal is to force the chaotic Chen's attractor to track the reference – the Lorenz attractor generated by

$$\begin{aligned} \dot{x}_{r1} &= \sigma(x_{r2} - x_{r1}) \\ \dot{x}_{r2} &= rx_{r1} - x_{r2} - x_{r2}x_{r3} \\ \dot{x}_{r3} &= -bx_{r3} + x_{r1}x_{r2} \end{aligned} \quad (25)$$

with

$$\sigma = 10, \quad b = 8/3, \quad r = 28, \quad \begin{pmatrix} x_{r1}(0) \\ x_{r2}(0) \\ x_{r3}(0) \end{pmatrix} = \begin{pmatrix} 1 \\ 1 \\ 1 \end{pmatrix} \quad (26)$$

In the simulations, the following dynamic neural network was used:

$$\dot{x} = Ax + W^*\Gamma z(x) + u \quad (27)$$

with

$$A = \begin{pmatrix} -5 & 0 & 0 \\ 0 & -5 & 0 \\ 0 & 0 & -5 \end{pmatrix} \quad \Gamma = \begin{pmatrix} 150 & 0 & 0 \\ 0 & 150 & 0 \\ 0 & 0 & 150 \end{pmatrix} \quad z(x) = \begin{pmatrix} \tanh(\beta x_1) \\ \tanh(\beta x_2) \\ \tanh(\beta x_3) \end{pmatrix} \quad \beta = 0.5 \quad (28)$$

The simulation results are shown in Figures 1-5, where the time evolution of the states and phase portraits are presented.

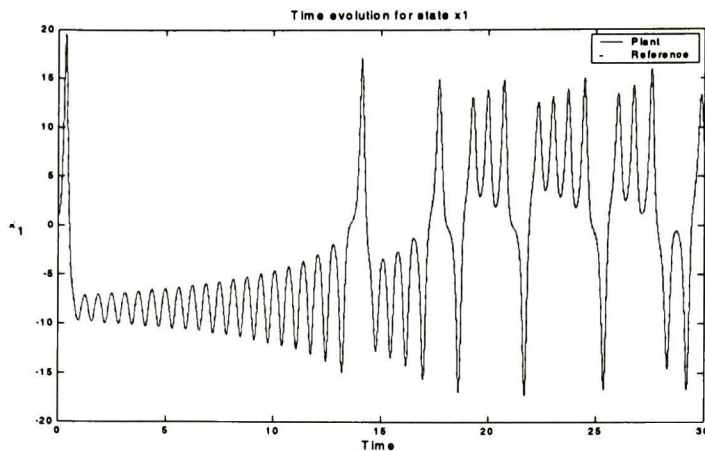


Fig. 1. Time evolution for state 1

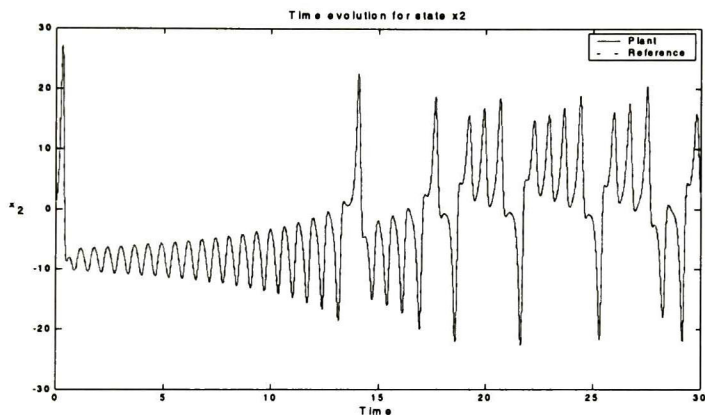


Fig. 2. Time evolution for state 2

## V. CONCLUSIONS

A new adaptive control structure based on a dynamic neural network for chaotic orbit tracking of unknown nonlinear systems has been developed. This structure is composed of a neural network identifier and a control law for orbit tracking. Stability of the tracking control system has also been established by means of the Lypunov function method.

The applicability of the proposed structure was tested via simulations, by an example of complex dynamical system: chaos synchronization. The results are quite encouraging. Research along this line is to further relax the required condition of having the same number of inputs and states in the control system.

## REFERENCES

- [1] K. S. Narendra and K. Parthasarathy, "Identification and control of dynamical systems using neural networks", *IEEE Trans. on Neural Networks*, vol. 1, no. 1, pp 4-27, 1990.
- [2] M. M. Gupta and D. H. Rao (Eds.), *Neuro-Control Systems, Theory and Applications*, IEEE Press, Piscataway, N.J., USA, 1994.
- [3] K. Hunt, G. Irwin and K. Warwick (Eds.), *Neural Networks Engineering in Dynamic Control Systems*, Springer Verlag, New York, USA, 1995.
- [4] A. S. Poznyak, W. Yu, E. N. Sanchez, and J. P. Perez, "Nonlinear adaptive trajectory tracking using dynamic neural networks", *IEEE Trans. on Neural Networks*, vol. 10, no 6, pp 1402-1411, Nov. 1999.

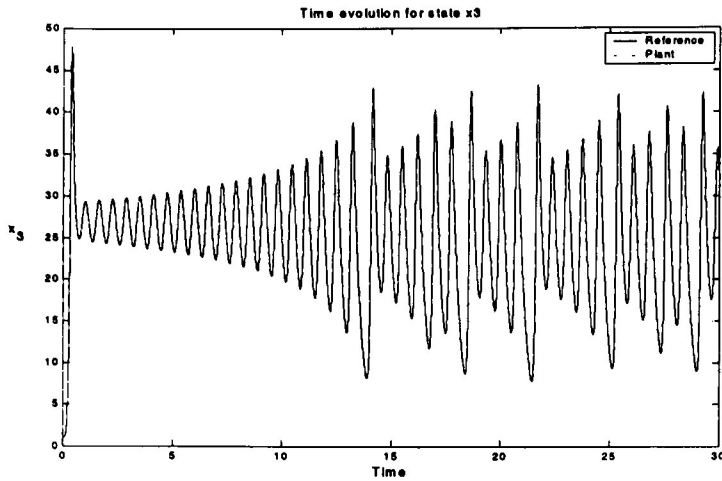


Fig. 3. Time evolution for state 3

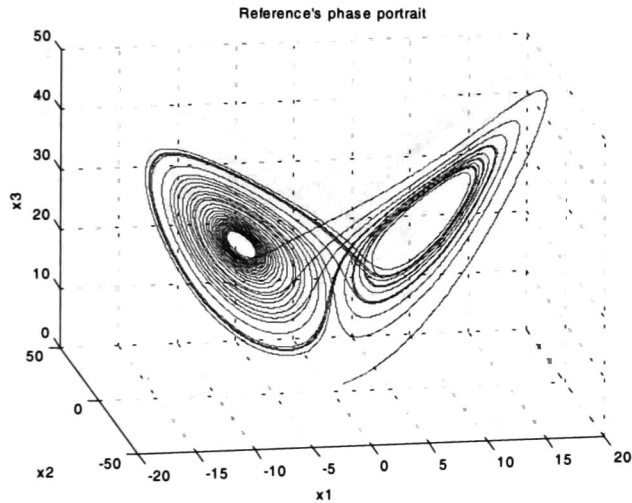


Fig. 4. Reference phase portrait

- [5] K. Suykens, L. Vandewalle and R. de Moor, *Artificial Neural Networks for Modelling and Control of Nonlinear Systems*, Kluwer Academic Publishers, Boston, USA, 1996.
- [6] G. A. Rovitahkis and M. A. Christodoulou, *Adaptive Control with Recurrent High-Order Neural Networks*, Springer Verlag, New York, USA, 2000.
- [7] A. Isidori, *Nonlinear Control Systems*, 3rd Ed., Springer Verlag, New York, USA, 1995.
- [8] D. J. Hill and P. Moylan, "The Stability of nonlinear dissipative systems", *IEEE Trans. on Auto. Contr.*, vol. 21, 708-711, 1996.
- [9] H. W. Knobloch, A. Isidori and D. Flockerzi, *Topics in Control Theory*, Birkhauser, Boston, USA, 1993.
- [10] T. Basar and P. Bernhard, *H-Infinity Optimal Control and Related Minimax Design Problems*, Birkhauser, Boston, USA, 1995.
- [11] M. Krstic and H. Deng, *Stablization of Nonlinear Uncertain Systems*, Springer Verlag, New York, USA, 1998.
- [12] E. N. Sanchez, J. P. Perez and G. Chen, "Using dynamic neural control to generate chaos: An inverse optimal control approach", *Int. J. of Bifurcation and Chaos*, to appear, 2001.
- [13] E. B. Kosmatopoulos *et al.*, "Dynamical neural networks that ensure exponential identification error convergence", *Neural Networks*, vol. 1, no. 2, pp 299-314, 1997.
- [14] H. Khalil, *Nonlinear Systems*, 2nd Ed., Prentice Hall, Upper Saddle River, NJ, USA, 1996.
- [15] E. N. Sanchez and J. P. Perez, "Input-to-state stability analysis for dynamic neural networks", *IEEE Trans. Circuits Syst. I*, vol. 46, pp 1395-1398, 1999.

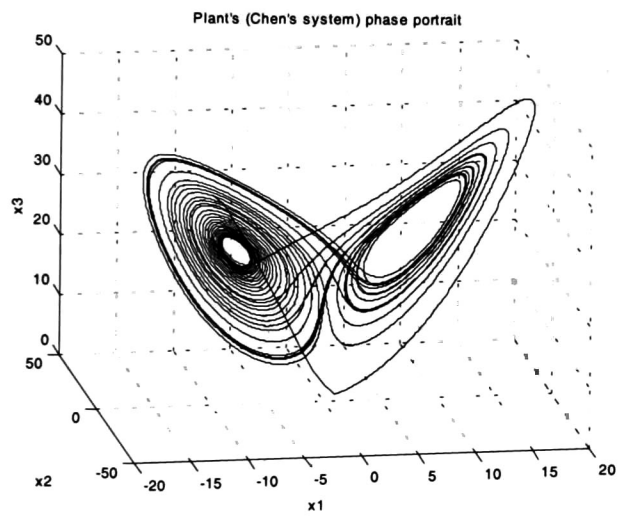


Fig. 5. Plant phase portrait



## RECURRENT NEURAL CONTROL FOR ROBOT TRAJECTORY TRACKING

Edgar N. Sanchez \* Jose P. Perez \*\* Luis J. Ricalde \*

\* CINVESTAV, Unidad Guadalajara, Apartado Postal 31-430,  
Plaza La Luna, Guadalajara, Jalisco C.P. 45091, Mexico, e-mail:  
sanchez@gdl.cinvestav.mx

\*\* CINVESTAV, Unidad Guadalajara, on doctoral studies leave  
from, Universidad Autonoma de Nuevo Leon, Monterrey, Nuevo  
Leon, Mexico

**Abstract:** This paper extends the results previously obtained for trajectory tracking of unknown plants using recurrent neural networks. The proposed controller structure is composed of a neural identifier and a control law defined by using the inverse optimal control approach, which has been improved so that less inputs than states are needed. The proposed new control scheme is applied to the control a robotic manipulator model.

**Keywords:** Neural networks, Trajectory tracking, Adaptive control, Lyapunov function, Stability analysis.

### 1. INTRODUCTION

Since the seminal paper [Narendra and Parthasarathy, 1990], there has been intensive studies in applying neural networks to identification and control of nonlinear systems. In this endeavor, many approaches use feedforward structures [Gupta and Rao, 1994],[Hunt *et al.*, 1995]. Lately, dynamic neural networks were developed, enabling more efficient modeling of complex dynamics [Poznyak *et al.*, 1999]. Two recent books, [Suykens *et al.*, 1996] and [Rovithakis and Christodoulou, 2000], have reviewed a broad spectrum of applications of dynamic neural networks for nonlinear systems identification and control. In [Suykens *et al.*, 1996], in particular, off-line learning is used, while in [Rovithakis and Christodoulou, 2000] on-line learning is employed, for adaptive identification and control,

where stability of closed-loop systems is analyzed using the Lyapunov function method. In [Rovithakis and Christodoulou, 2000], the trajectory tracking problem is studied, by means of reducing the system to a linear model following problem, in which an application to DC electric motors is discussed.

On the other hand, control methods that are applicable to general nonlinear systems have seen rapid development since the early 1980's. Main approaches include, for example, the use of differential geometry theory [Isidori, 1995]. Recently, the passivity approach has generated increasing interest for synthesizing control laws, also for general nonlinear systems [Hill and Moylan, 1996]. An important problem in this approach is how to achieve robust nonlinear control in the presence of unmodelled dynamics and external disturbances. In this direction, worth mentioning is the so-called  $H_\infty$  nonlinear

control approach [Basar and Bernhard, 1995]. One major difficulty with this approach today, alongside its possible system structural instability, seems to be the requirement of solving some resulting partial differential equations. In order to alleviate this computational problem, the so-called inverse optimal control technique was recently developed, based on the input-to-state stability concept [Krstic and Deng, 1998].

A control law for generating chaos in a dynamic neural network was designed in [Sanchez *et al.*, 2001(a)]. In [Sanchez *et al.*, 2001 (b)] and [Sanchez *et al.*, 2001 (c)], this methodology was modified for stabilization and trajectory tracking of an unknown chaotic dynamical system. The proposed adaptive control scheme is composed of a dynamic neural identifier and a controller, where the former is used to build an on-line model for the unknown plant and the latter, to ensure the unknown plant to track the reference trajectory. In this paper, we further improve the design by using less inputs than states in the control. The approach is based on the methodology developed in [Sanchez *et al.*, 2001(b)] and [Sanchez *et al.*, 2001(c)], in which the control law is optimal with respect to a well-defined cost functional. Trajectory tracking for a robotic manipulator model is demonstrated as an application example.

## 2. MATHEMATICAL PRELIMINARIES

### 2.1 A High-Order Recurrent Neural Network

In [Kosmatopoulos *et al.*, 1997], high-order recurrent neural networks (HORNN) are defined as

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

where  $x_i$  is the  $i$ th neuron state,  $L$  is the number of high-order connections,  $\{I_1, I_2, \dots, I_L\}$  is a collection of non-ordered subsets of  $\{1, 2, \dots, m+n\}$ ,  $a_i > 0$ ,  $w_{ik}$  are the adjustable weights of the neural network,  $d_j(k)$  are nonnegative integers, and  $y$  is a vector defined as  $y = [y_1, \dots, y_n, y_{n+1}, \dots, y_{n+m}]^T$

$= [S(x_1), \dots, S(x_n), S(u_1), \dots, S(u_m)]^T$ , with  $u = [u_1, u_2, \dots, u_m]^T$  being the input to the neural network, and  $S(\cdot)$  is a smooth sigmoid function formulated by  $S(x) = \frac{1}{1 + \exp(-\beta x)} + \varepsilon$ . For the sigmoid,  $\beta$  is a positive constant and  $\varepsilon$  is a small positive real number. Here,  $S(x) \in [\varepsilon, \varepsilon + 1]$ .

Clearly, system (1) allows the inclusion of high-order terms. By defining a new vector,

$$z(x, u) = [z_1(x, u), \dots, z_L(x, u)]^T \\ = \left[ \prod_{j \in I_1} y_j^{d_j(1)}, \dots, \prod_{j \in I_L} y_j^{d_j(L)} \right]^T \quad \text{system (1)}$$

can be rewritten as

$$\dot{x}_i = -a_i x_i + \sum_{k=1}^L w_{ik} z_k(x, u), \quad i = 1, \dots, n \\ \dot{x}_i = -a_i x_i + w_i^T z_i(x, u), \quad i = 1, \dots, n \quad (2)$$

where  $w_i = [w_{i,1} \dots w_{i,L}]^T$

In this paper, we assume that the input enters the neural network directly. We are particularly interested in less inputs than states, since this is a more efficient structure. For this purpose, we rewrite system (1) as

$$\dot{x}_i = -a_i x_i + w_i^T z_i(x) + \Omega u_i, \quad \Omega \in R^{n \times m} \quad (3)$$

We reformulate (2) in a matrix form and obtain

$$\dot{x} = Ax + Wz(x) + \Omega u \quad (4)$$

where  $x \in \mathbb{R}^n$ ,  $W^* \in \mathbb{R}^{n \times L}$ ,  $z(x) \in \mathbb{R}^L$ ,  $u \in \mathbb{R}^m$ , and  $A = -\lambda I$ , with  $\lambda > 0$ . In the following, we use the following slight modification of the HORNN:

$$\dot{x} = Ax + W\Gamma z(x) + \Omega u$$

where  $\Gamma \in \mathbb{R}^{L \times L}$  is a diagonal matrix defined by

$$\Gamma = \text{diag} \{ \gamma_1, \gamma_2, \dots, \gamma_{L-1}, \gamma_L \}$$

## 3. MODELLING THE UNKNOWN NONLINEAR PLANT

The unknown nonlinear plant is modeled as

$$x_p = f_p(x_p) + g_p(x_p)u \quad (5)$$

We propose to modelize the unknown nonlinear plant by a dynamic neural network, in the following form:

$$\dot{x}_p = \dot{x} + w_{per} \\ = Ax + W^* \Gamma z(x) + (x - x_p) + \Omega u \quad (6)$$

where  $x_p, x, z(x) \in \mathbb{R}^n$ ,  $W^*, \Gamma \in \mathbb{R}^{n \times n}$ ,  $u \in \mathbb{R}^{m \times 1}$ ,  $\Omega \in R^{n \times m}$ , and  $w_{per} = x - x_p$  represents the modelling error.  $W^*$  are the unknown values of the neural network which minimize the modelling error.

#### 4. TRAJECTORY TRACKING ANALYSIS

We proceed to analyze the tracking error between the unknown plant modeled by (6) and the reference trajectory given by

$$\dot{x}_r = f_r(x_r, u_r), \quad x_r \in \mathbb{R}^n \quad (7)$$

we define the tracking error and its time derivative as

$$e = x_p - x_r \quad (8)$$

$$\dot{e} = \dot{x}_p - \dot{x}_r \quad (9)$$

$$= Ax + W^* \Gamma z(x) + w_{per} + \Omega u - f_r(x_r, u_r)$$

Now, we proceed to add and subtract the terms  $\hat{W} \Gamma z(x_r)$ ,  $Ae$ , and  $\Omega \alpha_r(t, \hat{W})$ , so that

$$\begin{aligned} \dot{e} &= Ae + W^* \Gamma z(x) + \Omega u + \\ &(-f_r(x_r, u_r) + Ax_r + \hat{W} \Gamma z(x_r) + x_r - x_p \\ &+ \Omega \alpha_r(t, \hat{W})) - Ae - \hat{W} \Gamma z(x_r) \\ &- \Omega \alpha_r(t, \hat{W}) - Ax_r - x_r + x + Ax \end{aligned} \quad (10)$$

Note that the plant will track the reference signal even in the presence of uncertainties, if there exists a function  $\alpha_r(t, \hat{W})$  such that

$$\alpha_r(t, \hat{W}) = (\Omega^T \Omega)^{-1} \Omega^T (f_r(x_r, u_r) - Ax_r - \hat{W} \Gamma z(x_r) - (x_r - x_p)) \quad (11)$$

Next, assume that (11) holds, and define  $\tilde{W} = W^* - \hat{W}$ ,  $\tilde{u} = u - \alpha_r(t, \hat{W})$ , so that (10) is reduced to

$$\begin{aligned} \dot{e} &= Ae + \tilde{W} \Gamma z(x) + \hat{W} \Gamma (z(x) - z(x_r)) \\ &+ (A + I)(x - x_r) - Ae + \Omega \tilde{u} \\ &= Ae + \tilde{W} \Gamma z(x) + \hat{W} \Gamma (z(x) - z(x_p)) \\ &+ z(x_p) - z(x_r) \\ &+ (A + I)(x - x_p + x_p - x_r) - Ae + \Omega \tilde{u} \end{aligned} \quad (12)$$

$$\text{Then, by defining} \quad \tilde{u} = u_1 + u_2 \quad (13)$$

with

$$u_1 = (\Omega^T \Omega)^{-1} \Omega^T (-\hat{W} \Gamma (z(x) - z(x_p)) - (A + I)(x - x_p)) \quad (14)$$

Equation (12) reduces to

$$\begin{aligned} \dot{e} &= Ae + \tilde{W} \Gamma z(x) + \hat{W} \Gamma z(x_p) - z(x_r) \\ &+ (A + I)(x_p - x_r) - Ae + \Omega u_2 \end{aligned} \quad (15)$$

Moreover, by taking into account that  $e = x_p - x_r$ , equation (15) reduces to the following equation:

$$\begin{aligned} \dot{e} &= (A + I)e + \tilde{W} \Gamma z(x) \\ &+ \hat{W} \Gamma (z(x_p) - z(x_r)) + \Omega u_2 \end{aligned} \quad (16)$$

Therefore, the tracking problem reduces to a stabilization problem for the error dynamics (16). To solve this problem, we next apply the inverse optimal control approach.

#### 5. TRACKING ERROR STABILIZATION

Once (16) is obtained, we consider its stabilization. Note that  $(e, \tilde{W}) = (0, 0)$  is an equilibrium point of the undisturbed autonomous system. For stability analysis, we define the candidate Lyapunov function as

$$V = \frac{1}{2} \|e\|^2 + \frac{1}{2} \text{tr} \left\{ \tilde{W}^T \tilde{W} \right\} \quad (17)$$

Its time derivative, along the trajectories of (16), is

$$\begin{aligned} \dot{V} &= e^T (A + I)e + e^T \tilde{W} \Gamma z(x) \\ &+ e^T \hat{W} \Gamma (z(x_p) - z(x_r)) \\ &+ e^T \Omega u_2 + \text{tr} \left\{ \dot{\tilde{W}}^T \tilde{W} \right\} \end{aligned} \quad (18)$$

As in [Rovithakis and Christodoulou, 2000], we propose the following learning law:

$$\text{tr} \left\{ \dot{\tilde{W}}^T \tilde{W} \right\} = -e^T \tilde{W} \Gamma z(x) \quad (19)$$

$$\begin{aligned} \dot{\hat{w}}_{i,j} &= e_i(\gamma_j z(x_j)) \\ i &= 1, 2, \dots, n, \quad j = 1, 2, \dots, L. \end{aligned} \quad (20)$$

Then, we substitute (19) into (18), to obtain

$$\begin{aligned} \dot{V} &= -(\lambda - 1)e^T e \\ &+ e^T \hat{W} \Gamma (z(x_p) - z(x_r)) \\ &+ e^T \Omega u_2 \end{aligned} \quad (21)$$

$$L_f V = -(\lambda - 1)e^T e + e^T \hat{W} \Gamma (z(x_p) - z(x_r))$$

$$L_g V = e^T \Omega$$

Next, we consider the following inequality, proved in [Poznyak, et al. 1999]:

$$X^T Y + Y^T X \leq X^T \Lambda X + Y^T \Lambda^{-1} Y \quad (22)$$



which holds for all matrices  $X, Y \in \mathfrak{R}^{n \times k}$  and  $\Lambda \in \mathfrak{R}^{n \times n}$  with  $\Lambda = \Lambda^\top > 0$ . Applying it to the second term of (21), we obtain

$$\begin{aligned} \dot{V} &\leq -(\lambda - 1)e^\top e + \frac{1}{2}e^\top e \\ &\quad + \frac{1}{2}\|\hat{W}\|^2\|\Gamma\|^2\|z(x_p) - z(x_r)\|^2 + e^\top \Omega u_2 \end{aligned}$$

where  $\|\hat{W}\|$ ,  $\|\Gamma\|$  are the Euclidean norm of  $\hat{W}$ ,  $\Gamma$ , respectively. Taking in account  $x_p = e + x_r$  and denoting  $\phi_z = z(x_p) - z(x_r) = z(e + x_r) - z(x_r)$ , we have

$$\begin{aligned} \dot{V} &\leq -(\lambda - 1)e^\top e + \frac{1}{2}e^\top e \\ &\quad + \frac{1}{2}L_{\phi_z}^2\|\hat{W}\|^2\|\Gamma\|^2\|e\|^2 \\ &\quad + e^\top \Omega u_2 \\ \dot{V} &\leq -(\lambda - 1)e^\top e \\ &\quad + e^\top \left( \frac{1}{2} + \frac{1}{2}L_{\phi_z}^2\|\hat{W}\|^2\|\Gamma\|^2 \right) e \\ &\quad + e^\top \Omega u_2 \end{aligned} \quad (23)$$

where  $L_{\phi_z}^2$  is the Lipschitz constant of  $\phi_z$ .

To this end, we define the following control law:

$$\begin{aligned} u_2 &= -\mu(\Omega^\top \Omega)^{-1} \Omega^\top (1 + L_{\phi_z}^2\|\hat{W}\|^2\|\Gamma\|^2)e \quad (24) \\ \mu &> \frac{1}{2} \\ &\triangleq -\beta \left( R(e, \hat{W}) \right)^{-1} (L_g V)^\top \\ &\quad \beta \left( R(e, \hat{W}) \right)^{-1} \\ &= \mu (\Omega^\top \Omega)^{-1} \Omega^\top \left( 1 + L_{\phi_z}^2\|\hat{W}\|^2\|\Gamma\|^2 \right) \end{aligned}$$

with scalars  $R(e, \hat{W}) > 0, \beta > 0$ . Now, substituting (24) in (23) yields

$$\begin{aligned} \dot{V} &= -(\lambda - 1)e^\top e \quad (25) \\ &\quad - \left( \mu - \frac{1}{2} \right) e^\top \left( 1 + L_{\phi_z}^2\|\hat{W}\|^2\|\Gamma\|^2 \right) e \\ &< 0, \quad \forall e, \hat{W} \neq 0 \end{aligned}$$

Finally, the control law, which affects the plant and the neural network, is given by

$$\begin{aligned} u &= \tilde{u} + \alpha_r(t, \hat{W}) \quad (26) \\ &= (\Omega^\top \Omega)^{-1} \Omega^\top (-\hat{W}\Gamma(z(x) - z(x_p)) \\ &\quad - (A + I)(x - x_p) - \mu \left( 1 + L_{\phi_z}^2\|\hat{W}\|^2\|\Gamma\|^2 \right) e \end{aligned}$$

$$+ f_r(x_r, u_r) - Ax_r - \hat{W}\Gamma z(x_r) - x_r + x_p)$$

This control law gives asymptotic stability of the error dynamics and thus ensures the tracking to the reference signal.

## 6. OPTIMIZATION WITH RESPECT TO A COST FUNCTIONAL

Once the problem of finding the control law (24) is formulated based on the inverse optimal control approach, which stabilizes (16), we can proceed to formulate a cost functional defined by

$$\begin{aligned} J(\tilde{u}) &= \lim_{t \rightarrow \infty} \quad (27) \\ &2\beta V + \int_0^t \left( l(e, \hat{W}) + u_2^\top R(e, \hat{W})u_2 \right) dt \end{aligned}$$

where the Lyapunov function solves the Hamilton-Jacobi-Bellman family of partial derivative equations parametrized with  $\beta > 0$  as follows:

$$\begin{aligned} l(e, \hat{W}) + 2\beta L_f V \quad (28) \\ -\beta^2 L_g V R(e, \hat{W})^{-1} L_g V^\top = 0 \end{aligned}$$

Note that  $2\beta V$  in (27) is bounded when  $t \rightarrow \infty$ , since by (25)  $V$  is decreasing and bounded from below by  $V(0)$ . Then,  $\lim_{t \rightarrow \infty} V(t)$  exists and is finite. Recall that in [Krstic and Deng.,1998], we need  $l(e, \hat{W})$  to be positive definite and radially unbounded with respect to  $e$ . Here, from (28) we have

$$\begin{aligned} l(e, \hat{W}) &= -2\beta L_f V \quad (29) \\ &\quad + \beta^2 L_g V R(e, \hat{W})^{-1} L_g V^\top \end{aligned}$$

Substituting (24) into (29) and then applying (22) to the second term on the right side of  $L_f V$ , we have

$$\begin{aligned} l(e, \hat{W}) &\geq (\lambda - 1)\|e\|^2 \\ &\quad + (\mu - 1) \left( 1 + L_{\phi_z}^2\|\Gamma\|^2\|\hat{W}\|^2 \right) \|e\|^2 \end{aligned}$$

Since we select  $\lambda > 1$  and  $\mu > 1$ , we know that  $l(e, \hat{W})$  satisfies the condition of being positive definite and radially unbounded. Hence, (27) is a cost functional. It is easy to verify that, in (27),

$$l(e, \hat{W}, \hat{W}_1) + \tilde{u}^\top R(e, \hat{W}, \hat{W}_1)\tilde{u} = -2\beta \dot{V}$$

with optimal value  $J^* = 2\beta V(0)$ . This is achieved by the control law (24).

## 7. SIMULATION RESULTS ON ROBOTIC TRAJECTORY TRACKING

In order to test the applicability of the proposed control scheme, we consider the trajectory tracking problem for a robot manipulator model.

In the absence of friction, the dynamics of a 2-link rigid robot arm with torque control input can be written as

$$D(q)\ddot{q} + C(q, \dot{q})\dot{q} + G(q) = \tau$$

where  $q = [q_1 \ q_2]^T$ ,  $\dot{q} = [\dot{q}_1 \ \dot{q}_2]^T$ ,  $\ddot{q} = [\ddot{q}_1 \ \ddot{q}_2]^T$

$$D(q) = \begin{bmatrix} D_{11} & D_{12} \\ D_{21} & D_{22} \end{bmatrix},$$

$$D_{11} = m_1 l_{c1}^2 + m_2 (l_1^2 + l_{c2}^2 + 2l_1 l_{c2} \cos(q_2)) + I_1 + I_2$$

$$D_{12} = m_2 (l_{c2}^2 + l_1 l_{c2} \cos(q_2)) + I_2$$

$$D_{21} = D_{12} \quad D_{22} = m_2 l_{c2}^2 + I_2$$

$$C = \begin{bmatrix} -2m_2 l_1 l_{c2} \dot{q}_2 \sin(q_2) & -m_2 l_1 l_{c2} \dot{q}_2 \sin(q_2) \\ m_2 l_1 l_{c2} \dot{q}_1 \sin(q_2) & 0 \end{bmatrix}$$

$$G(q) = \begin{bmatrix} G_{11} \\ G_{21} \end{bmatrix} \quad \tau = [\tau_1 \ \tau_2]^T$$

$$G_{11} = m_1 g l_{c1} \cos(q_1) + m_2 g (l_1 \cos(q_1) + l_{c2} \cos(q_1 + q_2))$$

$$G_{21} = m_2 g l_{c2} \cos(q_1 + q_2)$$

$q_1, q_2 \rightarrow$  Angular position of link  $i$

$l_1, l_2 \rightarrow$  Length of link  $i$

$l_{c1}, l_{c2} \rightarrow$  Position of the center of gravity of link  $i$

$I_1, I_2 \rightarrow$  Inertia of link  $i$

$g \rightarrow$  Gravitational acceleration ( $9.81 \text{ m/s}^2$ )

where  $D(q)$  is a positive definite and symmetric inertia matrix,  $C(q, \dot{q})$  is the matrix containing the effect of centripetal and Coriolis forces, and  $G(q)$  contains the gravitational torques.

We consider the example studied in [Llama *et al.*, 2000], here as the nonlinear unknown plant. We try to force this manipulator to track a reference signal given by

$$q_{r1} = 1.57 + 0.78(1 - e^{-2t^3}) + 0.17(1 - e^{-2t^3}) \sin w_1 t$$

$$q_{r2} = 1.57 + 1.04(1 - e^{-1.8t^3}) + 2.18(1 - e^{-1.8t^3}) \sin w_2 t$$

where  $w_1$  and  $w_2$  are the frequencies of the desired trajectories for link 1 and link 2, respectively. This trajectory requires large velocity and acceleration

from the manipulator. In the simulation,  $w_1 = 15 \text{ rad/s}$  and  $w_2 = 3.5 \text{ rad/s}$  were used. We select the initial position at  $q_1 = -90^\circ$  and  $q_2 = 0^\circ$ , which correspond to the inferior equilibrium point. The parameters for this robot model used in [Llama *et al.*, 2000] are the following:

$$m_1 = 23.902 \text{ kg.}$$

$$m_2 = 1.285 \text{ kg.}$$

$$l_1 = 0.45 \text{ m.}$$

$$l_2 = 0.45 \text{ m.}$$

$$l_{c1} = 0.091 \text{ m.}$$

$$l_{c2} = 0.048 \text{ m.}$$

$$I_1 = 1.226 \text{ kg.m}^2$$

$$I_2 = 0.093 \text{ kg.m}^2$$

$$g = 9.81 \text{ m/s}^2$$

We use the following dynamical neural network to modelize the same system:

$$\dot{x}_p = Ax + W^* \Gamma z(x) + (x - x_p) + \Omega u$$

with  $A = -150I$ ,  $\Gamma = 0.5I$ ,  $k = 0.45$ ,  $I \in R^{4 \times 4}$

$$z(x) = \tanh(kx) \quad , \quad \Omega = \begin{pmatrix} 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{pmatrix}^T$$

For the control law (26), we select  $\mu = 85$ . The time evolution for the angles and torques applied to the links are shown in Figs. 1–4. As can be seen, trajectory tracking is successful.

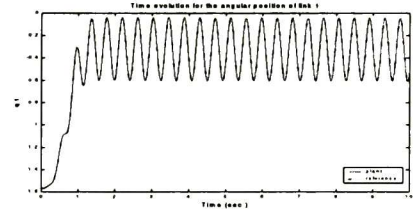


Fig. 1. Time evolution for the position of link 1

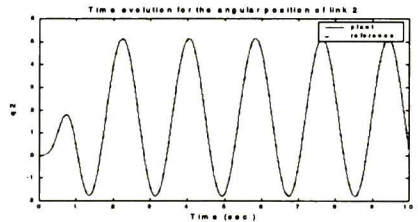


Fig. 2. Time evolution for the position of link 2

## 8. CONCLUSIONS

We have extended the designed previously developed in [Sanchez *et al.*, 2001 (b)] and [Sanchez *et al.*, 2001 (c)] for the adaptive trajectory tracking



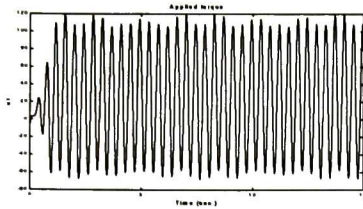


Fig. 3. Torque applied to link 1

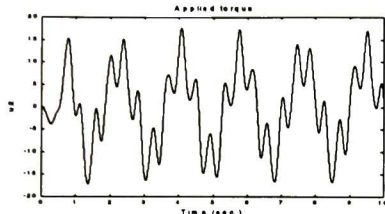
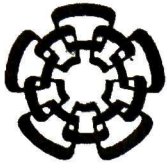


Fig. 4. Torque applied to link 2

control problem, based on the inverse optimal control approach. We have further relaxed the condition of having the same number of inputs and states, so as to allow for less inputs than states, in the control law. The proposed scheme consists of a neural identifier and a controller, in which the former builds an on-line model of the unknown plant and the latter is optimal with respect to a meaningful cost functional. A robot model is used to verify the design for trajectory tracking, with satisfactory performance, where the plant model is assumed to be unknown. Research will continue to implement the scheme in real time and to further test it in a laboratory environment.

## 9. REFERENCES

- [1] Basar T. and P. Bernhard, *H-Infinity Optimal Control and Related Minimax Design Problems*, Birkhauser, Boston, USA, (1995).
- [2] Krstic M. and H. Deng, *Stabilization of Nonlinear Uncertain Systems*, Springer Verlag, New York, USA, (1998).
- [3] Gupta M. M. and D. H. Rao (Eds.), *Neuro-Control Systems, Theory and Applications*, IEEE Press, Piscataway, N.J., USA, (1994).
- [4] Hill D. J. and P. Moylan "The Stability of nonlinear dissipative systems", *IEEE Trans. on Auto. Contr.*, vol. 21, 708-711, (1996).
- [5] Hunt K., G. Irwin and K. Warwick (Eds.), *Neural Networks Engineering in Dynamic Control Systems*, Springer Verlag, New York, USA, (1995).
- [6] Isidori A. , *Nonlinear Control Systems*, 3rd Ed., Springer Verlag, New York, USA, (1995).
- [7] Khalil H., *Nonlinear Systems*, 2nd Ed., Prentice Hall, Upper Saddle River, NJ, USA, (1996).
- [8] Kosmatopoulos E. B. et al., "Dynamical neural networks that ensure exponential identification error convergence", *Neural Networks*, vol. 1, no. 2, pp 299-314, (1997).
- [9] Llama M. A., R. Kelly, V. Santibañez, "Stable Computed torque of Robot manipulators via fuzzy self-tuning", *IEEE Trans. on Systems, Man, and Cybernetics*, Vol. 30, 143-150, (Feb. 2000)
- [10] Narendra K. S. and K. Parthasarathy, "Identification and control of dynamical systems using neural networks", *IEEE Trans. on Neural Networks*, vol. 1, no. 1, pp 4-27, (1990).
- [11] Poznyak A. S., W. Yu, E. N. Sanchez, and J. P. Perez, "Nonlinear adaptive trajectory tracking using dynamic neural networks" *IEEE Trans. on Neural Networks*, vol. 10, no 6, pp 1402-1411, (1999).
- [12] Rovitahkis G. A. and M. A. Christodoulou, *Adaptive Control with Recurrent High-Order Neural Networks*, Springer Verlag, New York, USA,(2000).
- [13] Sanchez E. N. and J. P. Perez, "Input-to-state stability analysis for dynamic neural networks", *IEEE Trans. Circuits Syst. I*, vol. 46, pp 1395-1398, (1999).
- [14] Sanchez E. N. , J. P. Perez and G. Chen, "Using dynamic neural control to generate chaos: An inverse optimal control approach" *Int. J. of Bifurcation and Chaos*, vol. 11, no. 3, pp. 857-863 (2001a).
- [15] Sanchez E. N. J. P. Perez, L. Ricalde and G. Chen, "Trajectory tracking via adaptive neural control", *Proc. of IEEE Int. Symp. on Int. Cont.*, Mexico City, pp. 286-289, ( 2001b).
- [16] Sanchez E. N. J. P. Perez, L. Ricalde and G. Chen "Chaos production and synchronization via adaptive neural control", to appear in *Proc. of IEEE Conf. on Dec. and Cont.*, Orlando, FL, USA,(Dec. 4 - 7, 2001c).
- [17] Suykens K., L. Vandewalle and R. de Moor, *Artificial Neural Networks for Modelling and Control of Nonlinear Systems*, Kluwer Academic Publishers, Boston, USA,(1996).



**Cinvestav**

**Centro de Investigación y de Estudios  
Avanzados del IPN**

**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 ADAPTABLE PARA SEGUIMIENTO DE TRAYECTORIAS del(a) C. Luis Josué RICALDE CASTELLANOS el día 14 de Diciembre de 2001

Dr. Edgar Nelson Sanchez  
Camperos  
Investigador Cinvestav 3B  
CINVESTAV GDL  
Guadalajara

Dr. Alexander Georgievich  
Loukianov  
Investigador Cinvestav 3A  
CINVESTAV GDL  
Guadalajara

Dr. Juan Manuel Ramírez  
Arredondo  
Investigador Cinvestav 3A  
CINVESTAV GDL  
Guadalajara

Dr. Fernando Lara Rojo  
Profesor Investigador  
Instituto Tecnológico de  
Estudios Superiores de  
Occidente  
Guadalajara





CINVESTAV  
BIBLIOTECA CENTRAL



SSIT000003919