

CENTRO DE INVESTIGACIÓN Y DE ESTUDIOS
AVANZADOS DEL IPN

DEPARTAMENTO DE INGENIERÍA ELÉCTRICA
SECCIÓN DE CONTROL AUTOMÁTICO

**Control adaptable del sistema no-lineal TORA usando
redes neuronales dinámicas**

TESIS QUE PRESENTA EL

Ing. Marco Antonio Moreno Armendariz*

PARA OBTENER EL GRADO DE

MAESTRO EN CIENCIAS EN INGENIERÍA ELÉCTRICA

DIRECTORES DE TESIS:

**Dr. Wen Yu Liu
Dr. Alexander S. Pozniak**

México, D.F., 13 de agosto de 1999

* Becario del CONACyT

Contenido

1	Introducción	9
1.1	<i>El problema de identificación y control</i>	10
1.2	<i>Motivación del trabajo</i>	10
1.3	<i>Estructura del trabajo</i>	11
2	Redes Neuronales	14
2.1	<i>Introducción</i>	14
2.2	<i>Modelo de una Neurona</i>	15
2.3	<i>Estructuras de las redes neuronales</i>	18
2.3.1	<i>Redes de alimentación hacia adelante de una capa</i>	19
2.3.2	<i>Redes de alimentación hacia adelante multicapa</i>	21
2.3.3	<i>Redes de funciones radiales básicas</i>	22
2.3.4	<i>Redes neuronales dinámicas</i>	22
2.4	<i>Redes Neuronales en Identificación</i>	24
2.4.1	<i>Estrategias de Identificación</i>	25
2.4.2	<i>Aproximación de Perceptrón Dinámico Multicapa</i>	27
2.5	<i>Redes Neuronales en Control</i>	34
2.5.1	<i>Estructuras de Control</i>	36
2.5.2	<i>Control Adaptable</i>	38
3	Modelado del sistema no-lineal TORA	40

3.1	<i>Introducción</i>	40
3.2	<i>Modelo Lagrangiano</i>	40
3.3	<i>Modelo Normalizado</i>	41
3.4	<i>Modelo de Primer Orden</i>	42
3.5	<i>Modelo en Cascada</i>	43
4	Identificación de Sistemas No-lineales	51
4.1	<i>Introducción</i>	51
4.2	<i>Redes Neuronales Dinámicas de una capa para Identificación de Sistemas No-lineales</i>	52
4.2.1	<i>Estructura de la red neuronal de una capa</i>	52
4.2.2	<i>Preliminares</i>	53
4.2.3	<i>Regla de Actualización de Pesos</i>	55
4.2.4	<i>Análisis de Estabilidad</i>	55
4.2.5	<i>Simulaciones</i>	58
4.3	<i>Estimación de los estados internos de un sistema no-lineal</i>	61
4.3.1	<i>Introducción</i>	61
4.3.2	<i>Observador de modo deslizante</i>	62
4.3.3	<i>Diagrama de diseño</i>	63
4.3.4	<i>Análisis de Estabilidad</i>	65
4.3.5	<i>Simulaciones</i>	66
4.4	<i>Redes Neuronales Dinámicas Multicapa para Identificación de Sistemas No-lineales con medición parcial de sus estados</i>	72
4.4.1	<i>Planteamiento del Problema</i>	72
4.4.2	<i>Estructura de la red neuronal dinámica multicapa</i>	73
4.4.3	<i>Preliminares</i>	74
4.4.4	<i>Regla de Actualización de Pesos</i>	76
4.4.5	<i>Análisis de Estabilidad</i>	77
4.4.6	<i>Simulaciones</i>	81

5	Control Adaptable usando Redes Neuronales Dinámicas	84
5.1	<i>Introducción</i>	84
5.2	<i>Control Adaptable Directo</i>	85
5.2.1	<i>Planteamiento del Problema</i>	85
5.2.2	<i>Estructura de la red neuronal dinámica</i>	86
5.2.3	<i>Diseño del Neuro-controlador</i>	86
5.2.4	<i>Preliminares</i>	88
5.2.5	<i>Regla de actualización de pesos</i>	90
5.2.6	<i>Análisis de Estabilidad</i>	91
5.2.7	<i>Simulaciones</i>	95
5.3	<i>Control Adaptable Indirecto</i>	99
5.3.1	<i>Planteamiento del Problema</i>	99
5.3.2	<i>Estructura de la red neuronal dinámica</i>	99
5.3.3	<i>Preliminares</i>	100
5.3.4	<i>Regla de Actualización de Pesos</i>	102
5.3.5	<i>Análisis de Estabilidad</i>	102
5.3.6	<i>Diseño del Neuro Controlador</i>	106
5.3.7	<i>Simulaciones</i>	109
6	Conclusiones	114
A	Matemático	121
B	Diagramas de Simulink	123

Indice de Figuras

2-1	Esquema de una neurona biológica.	16
2-2	Modelo no-lineal de una neurona.	16
2-3	Tabla de funciones No-lineales $g(x)$	18
2-4	Perceptrón multicapa.	21
2-5	Esquema de una red neuronal dinámica.	23
2-6	Sistemas dinámicos y estáticos equivalentes.	24
2-7	Esquema de identificación de una red dinámica.	26
2-8	Estructura general del perceptrón dinámico multicapa.	28
2-9	Relación entre las redes neuronales y el control.	36
2-10	Esquema general del control directo.	37
2-11	Esquema general del control indirecto.	38
3-1	Oscilador Traslacional con Actuador Rotacional (TORA).	41
4-1	Identificación usando una RND con una capa del estado q	59
4-2	Identificación usando una RND con una capa del estado \dot{q}	60
4-3	Identificación usando una RND con una capa del estado θ	60
4-4	Identificación usando una RND con una capa del estado $\dot{\theta}$	61
4-5	Diagrama de diseño del observador.	64
4-6	Resultados de observación para el modo deslizante del estado \dot{q}	70
4-7	Resultados de observación para el modo deslizante del estado \dot{q}	71
4-8	Resultados de observación usando el compensador del estado \dot{q}	71

4-9	Resultados de observación usando el compesador del estado $\dot{\theta}$	72
4-10	Estructura general de una red neuronal dinámica.	74
4-11	Identificación con estimación del estado q	82
4-12	Identificación con estimación del estado \dot{q}	82
4-13	Identificación con estimación del estado θ	83
4-14	Identificación con estimación del estado $\dot{\theta}$	83
5-1	Comportamiento en tiempo de la ley de control (5.4).	96
5-2	Control Directo para el estado q	97
5-3	Control Directo para el estado \dot{q}	97
5-4	Control Directo para el estado θ	98
5-5	Control Directo para el estado $\dot{\theta}$	98
5-6	Identificación usando una RND multicapa de estado q	109
5-7	Identificación usando una RND multicapa de estado \dot{q}	110
5-8	Identificación usando una RND multicapa de estado θ	110
5-9	Identificación usando una RND multicapa de estado $\dot{\theta}$	111
5-10	Comportamiento en tiempo de la ley de control (5.59).	111
5-11	Control Indirecto para el estado q	112
5-12	Control Indirecto para el estado \dot{q}	112
5-13	Control Indirecto para el estado θ	112
5-14	Control Indirecto para el estado $\dot{\theta}$	113
B-1	Diagrama Principal de la RND de una capa.	123
B-2	Diagrama Principal de los observadores usados para estimar los estados. .	124
B-3	Diagrama Principal de Identificación usando la estimación de 2 estados. .	124
B-4	Diagrama Principal del Control Directo.	125
B-5	Diagrama Principal del Control Indirecto.	125

Notación

$:=$	significa “igual por definición”;
$x_t \in \mathfrak{R}^n$	vector de estado del sistema no-lineal en el tiempo $t \in \mathfrak{R}^+ := \{t : t \geq 0\}$;
$\hat{x}_t \in \mathfrak{R}^n$	estado de la red neuronal dinámica;
$\bar{x}_t \in \mathfrak{R}^n$	estado del observador de modo deslizante;
$u_t \in \mathfrak{R}$	acción de control dado;
$y_t \in \mathfrak{R}$	vector de la salida;
$f(x_t, u_t, t) : \mathfrak{R}^{n+2} \rightarrow \mathfrak{R}^n$	vector de la función no-lineal que describe la dinámica del sistema;
$C \in \mathfrak{R}^{n \times m}$	matriz de salida desconocida;
$A \in \mathfrak{R}^{n \times n}$	matriz Hurwitz (estable);
$a \in \mathfrak{R}$	con $a < 0$;
$W_{1,t} \in \mathfrak{R}^{n \times m}$ y $W_{2,t} \in \mathfrak{R}^{n \times m}$	matrices de los pesos (f, g respectivamente);
W_1^* y W_2^*	valores fijos de $W_{1,t}$ y $W_{2,t}$;
\bar{W}_1 y \bar{W}_2	cotas superiores de $W_{1,t}$ y $W_{2,t}$;
\tilde{W}_1 y \tilde{W}_2	estimados del error de $W_{1,t}$ y $W_{2,t}$;
$k_i \in \mathfrak{R}$ y $K_i \in \mathfrak{R}^{n \times n}$	ganancias del algoritmo de aprendizaje;
$\phi(\cdot)$	función diagonal matricial;
$\sigma(\cdot)$	función vectorial n-dimensional;
Δ_t	error de identificación;
r	error de seguimiento;
\tilde{f}	error de modelado;
$\ \cdot\ $	norma Euclidiana para vectores y para cualquier matriz A se define como: $\ A\ := \sqrt{\lambda_{\max}(A^T A)}$
	λ_{\max} es el máximo eigenvalor de la matriz A
$\ \cdot\ _W$	norma euclidiana del vector $x \in \mathfrak{R}$:
	$\ x\ _W := x^T W x$

Capítulo 1

Introducción

Indudablemente, desde que ha habido un fuerte renacimiento en la última década, las redes neuronales artificiales (RNA) están jugando cada vez un papel más importante en la Ingeniería. Por algunos años, éstas han sido una considerable promesa para aplicaciones en control no-lineal.

En su mayor extensión, la aplicación de las RNA al control automático es usualmente para aproximar el modelo de la planta [10] y, en base a este modelo, diseñar la ley de control. La principal estructura de las RNA en uso es la red neuronal estática (RNE) [15][13]: éstas procesan la información entrada/salida, lo cual, da como resultado un mapeo algebraico no-lineal.

En base a la capacidad de las RNE para aproximar cualquier función continua no-lineal, una extensión natural es aproximar el comportamiento entrada/salida de los sistemas no-lineales usando redes neuronales dinámicas (RND): su proceso de la información está descrito por ecuaciones diferenciales en tiempo continuo o por ecuaciones en diferencias en tiempo discreto. Existen resultados en este tipo de redes, pero requieren de condiciones muy restrictivas como: estabilidad en lazo abierto o que el tiempo sea finito[36].

Esta tesis busca familiarizar al lector con el nuevo campo de las RND aplicadas al control no-lineal. Por lo cual, se presenta un análisis sistemático para la identificación,

estimación de estados, y seguimiento de trayectorias para sistemas no-lineales. La herramienta principal para el análisis es el *enfoque tipo Lyapunov*.

1.1 *El problema de identificación y control*

En esta tesis se plantea como objetivo general, proponer una herramienta que ayude en el campo de la identificación y en el caso del control para una clase de sistemas no-lineales, como lo es el sistema no-lineal conocido como: Oscilador Traslacional con Actuador Rotacional (TORA) para el cual se propone un control adaptable directo e indirecto. Se asume que el sistema es de la forma:

$$\begin{aligned}\dot{x}_1 &= x_2 \\ \dot{x}_2 &= x_3 \\ &\vdots \\ \dot{x}_n &= f(\mathbf{x}) + g(\mathbf{x})u \\ y &= x_1\end{aligned}\tag{1.1}$$

Las funciones f y g son desconocidas, por lo cual se propone una RND para identificar el sistema y diseñar una ley de control: $u \in \mathfrak{R}^1$.

1.2 *Motivación del trabajo*

Debido a la complejidad y variedad de sistemas no-lineales, no existe una sola metodología que resuelva los problemas de identificación y control para esta clase de sistemas, por lo que, en esta tesis se propone utilizar las redes neuronales dinámicas para resolver el problema de identificación y de control para la clase de sistemas no-lineales (1.1). Dado que éstas han demostrado su gran capacidad como aproximadores [10], resultan ser una herramienta adecuada para la solución de estos problemas.

1.3 *Estructura del trabajo*

La estructura de la tesis que se presenta, consiste en dos partes principales:

1. **El identificador** : Una RND es usada para reconstruir el modelo de la planta; así, la RND se utiliza como identificador. Se considera 2 casos:
 - (a) Dádo un sistema no-lineal, se tiene acceso a la medición de la salida del sistema; así la RND se usa para identificar al sistema.
 - (b) Dádo un sistema no-lineal, se cuenta con la medición parcial de los estados, por lo que se implementa observadores de modos deslizantes para estimar algunos de los estados y con base en este resultado se utiliza la RND para identificar al sistema.
2. **El controlador**: Una RND es usada para diseñar una ley de control que resuelva el problema de seguimiento de una trayectoria dada. Se consideran dos casos:
 - (a) Dado un sistema no-lineal, se diseña un neuro-controlador, sin tener ningun tipo de información previa por lo que el aprendizaje de la RND, se lleva a cabo en línea.
 - (b) Dado un sistema no-lineal, se usa una RND para identificar al sistema y con base al resultado de esta identificación se diseña el neuro controlador.

En general, se obtuvieron 4 nuevos resultados que se encuentran en los siguientes capítulos, cuyo contenido es el siguiente:

Capítulo 2: *Redes Neuronales*. Se presenta un breve panorama del desarrollo y estructuras de las redes neuronales. Primero se muestra la estructura básica de las redes neuronales biológicas (RNB). Después, se discuten diferentes estructuras de las RNA como los son: de una capa, multicapa, funciones radiales básicas, además estableciendo la diferencia entre las RNE y las RND. Luego, se presenta un panorama general de las RNA en la identificación, mostrando las estrategias más comúnmente usadas, el primer

resultado que se presenta en la tesis que es la aproximación del perceptrón multicapa dinámico, por último, se hace mención del algoritmo de propagación hacia atrás (back-propagation) y la aplicación de las RNA en control.

Capítulo 3: *Sistema TORA*. Los resultados teóricos obtenidos en este trabajo, se ejemplifican mediante el sistema no-lineal que se muestra en la figura 3-1 que representa un oscilador traslacional con actuador rotacional (TORA) [6]. Fue introducido por primera vez en [45]. Se eligió este sistema ya que fue propuesto como un sistema no-lineal estándar que permitirá realizar pruebas de comparación entre diferentes metodologías de control. Se presenta el modelo Lagrangiano, el cual se normaliza para luego ser reescrito como un sistema de primer orden, por último se presenta un cambio de variables de estado con el cual se obtiene un modelo en cascada el cual se usará para ejemplificar los resultados teóricos, por medio de simulaciones.

Capítulo 4: *Identificación de sistemas no-lineales*. Se presenta la aplicación de las RND en el campo de la identificación En este caso, para sistemas no-lineales. La capacidad de estas en este proceso, justifica su aplicación al control no-lineal, en donde, los modelos matemáticos de este tipo de sistemas son muy complejos o no es posible contar con toda la información del mismo.

Primero, se presenta una RND de una capa, su algoritmo de aprendizaje y el análisis de estabilidad. Luego, se introduce un observador de modo deslizante, para resolver el problema cuando no se tiene acceso completo a todos los estados del sistema no-lineal, para en base a esta estimación, realizar la identificación con una RND multicapa, junto con su algoritmo de aprendizaje y su análisis de estabilidad. El análisis de estabilidad, como el diagrama de diseño para este tipo de observador es el segundo resultado obtenido en este trabajo. Este trabajo fue sujeto a revisión y aceptado en el 38th IEEE Conference on Decision and Control (CDC'99).

Capítulo 5: *Control Adaptable Directo*. Aquí, se plantea el problema de seguimiento para un sistema no-lineal, para lo cual se define un error de seguimiento, en base a este, se propone una RND multicapa como neuro controlador, el cual resuelve este problema. Se

presenta el algoritmo de aprendizaje y el análisis de estabilidad; éste es el tercer resultado obtenido en la tesis. Este trabajo fue sujeto a revisión y aceptado en International Joint Conference on Neural Networks (IJCNN'99).

Control Adaptable Indirecto. Al igual que en el capítulo anterior, se plantea el problema de seguimiento para un sistema no-lineal, sin embargo, dado que se busca implementar un control indirecto el análisis se divide en 2 partes: La primera es; la identificación del sistema no lineal con una red neuronal multicapa, de la cual, se presenta su algoritmo de aprendizaje y el análisis de estabilidad. En la segunda, se define el error de seguimiento y se propone el neuro controlador que resuelve el problema, este es el cuarto resultado obtenido en la tesis. Este trabajo fue sujeto a revisión y aceptado el la 5^{ta} Conferencia de Ingeniería Eléctrica (CIE'99).

Capítulo 2

Redes Neuronales

2.1 *Introducción*

La última meta de la ingeniería en control es implementar sistemas automáticos los cuales pudieran operar con un mayor grado de independencia de las acciones del humano en un ambiente no estructurado y con incertidumbres [12]. Tal sistema puede ser nombrado como *autónomo* o *inteligente*. Pudiera necesitar sólo ser presentado como una meta y podría alcanzarse este objetivo con una iteración continua con el ambiente a través de una retroalimentación acerca de su comportamiento. Podría continuar adaptándose y ejecutando tareas con una gran eficiencia en condiciones impredecibles y cambiantes. Esto sería muy útil cuando la iteración directa del humano pudiera ser peligrosa, propensa a fallas, o imposible.

Los sistemas *biológicos* son posibles esqueletos para el diseño de tales sistemas autónomos. Ellos proveen muchas pistas para el desarrollo de aprendizaje *robusto* (altamente estable) y algoritmos adaptables. Los sistemas biológicos procesan la información, de forma diferente que los esquemas de control convencionales; éstos no están basados en ningún modelo y son muy eficientes para tratar con incertidumbres y complejidad. Estos no requieren del desarrollo de un modelo matemático para ejecutar tareas complejas. Ciertamente, éstos puede aprender a ejecutar nuevas tareas y adaptarse fácilmente a cambios en

el ambiente. Si los principios fundamentales de la computación encajaron en los sistemas nerviosos, es comprensible que una generación totalmente nueva de métodos de control pudieran ser desarrollados mas allá de las capacidades de las presentes técnicas basadas en un modelo matemático explícito.

Un sistema de control tiene la habilidad de aprender si adquiere información durante la operación, acerca de comportamientos desconocidos de la planta y su ambiente tal que la ejecución completa es mejorada. Con este enriquecimiento del controlador con el aprendizaje, es posible expandir la región de operación y finalmente la implementación de sistemas autónomos.

Una clase de modelos que tiene la potencialidad de implementar este aprendizaje son las redes neuronales artificiales. Ciertamente, la morfología neuronal del sistema nervioso es mucho más compleja. No obstante, una analogía simplificada puede ser desarrollada, la cual podría ser utilizada en aplicaciones de ingeniería. Basándose en esta comprensión simplificada, las estructuras de las redes neuronales artificiales pueden ser desarrolladas.

2.2 Modelo de una Neurona

Una red neuronal artificial (RNA) [15] es un elemento capaz de procesar gran cantidad de información de forma paralela y distribuida, inspirada de las redes neuronales biológicas (ver Fig.2-1), las cuales pueden almacenar conocimiento experimental y tenerlo disponible para su uso [13]. Esta tiene algunas similitudes con el cerebro, como lo son:

1. El conocimiento es adquirido a través del proceso de aprendizaje.
2. La conectividad entre neuronas es llamada pesos sinápticos y son utilizados para almacenar el conocimiento.

El procedimiento para el proceso de aprendizaje es conocido como el algoritmo de aprendizaje. Su función es modificar los pesos sinápticos de las redes para alcanzar una

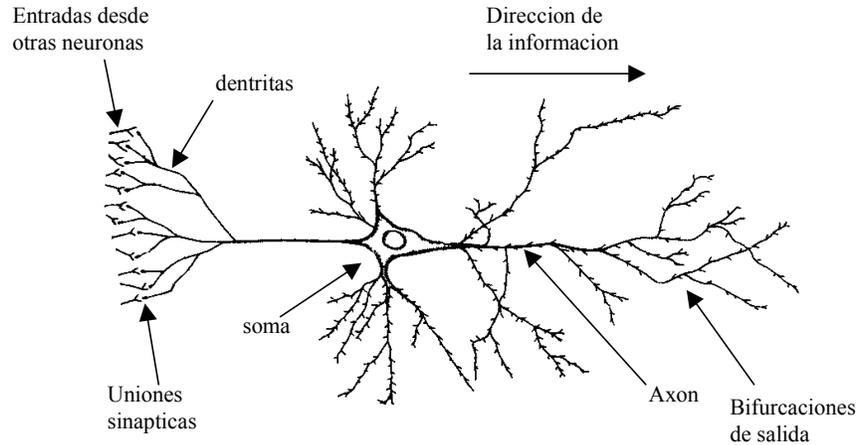


Figura 2-1: Esquema de una neurona biológica.

meta preestablecida. La modificación de los pesos provee el método tradicional para el diseño e implementación de las redes neuronales.

La neurona es la unidad fundamental para la operación de la red neuronal. La Fig. 2-2 muestra el esquema de una neurona.

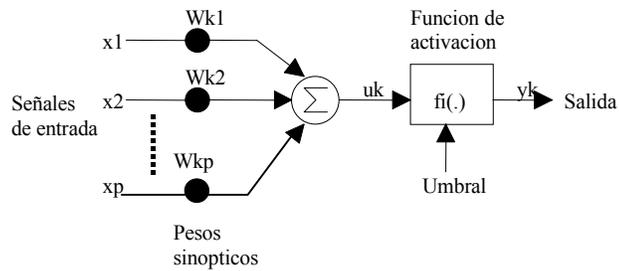


Figura 2-2: Modelo no-lineal de una neurona.

Existen 3 elementos básicos de la RNA:

1. Un conjunto de uniones de sinápsis, con cada elemento caracterizado por su propio peso.
2. Un sumador el cual suma los componentes de la señal de entrada, multiplicados por su respectivo peso sináptico.

3. Una función de activación no-lineal que transforma la salida del sumador en la entrada de la siguiente neurona.

Un umbral externo es también aplicado para reducir la entrada a la función de activación. En términos matemáticos, la *i-ésima* neurona puede ser descrita como:

$$\begin{aligned} u_i &= \sum_{j=1}^n w_{ij} x_j \\ y_i &= \varphi(u_i - \rho_i) \end{aligned} \tag{2.1}$$

donde:

x_j *j-ésimo* componente de la entrada.

w_{ij} peso de la conexión entre la *j-ésima* componente de la entrada y la *i-ésima* neurona .

u_i salida del sumador.

ρ_i umbral.

$\varphi(\cdot)$ función de activación no-lineal.

y_i salida de la *i-ésima* neurona.

La función de activación no-lineal, es denotada por $g(\cdot)$ y genera el elemento de la salida y_i , recibiendo como entrada x_i :

$$y_i = g(x_i) \tag{2.2}$$

Una clasificación de este tipo de funciones es:

1. Diferenciable y No-diferenciable
2. Tipo pulso y Tipo escalón
3. Positiva y Promedio cero

La clasificación (1) se distingue por tener funciones suaves y discontinuas. Las funciones suaves son necesarias para algunos algoritmos de adaptación como el de propa-

Nombre	Formula	Caracteristicas
Umbral	+1 si $x > 0$ sino 0	No-diferenciable, tipo escalon, positiva
Umbral	+1 si $x > 0$ sino -1	No-diferenciable, tipo escalon, prom. cero
Sigmoide	$1 / (1 + e^{-x})$	Diferenciable, tipo escalon, positiva
Tangente hiperbolica	$\tanh(x)$	Diferenciable, tipo escalon, prom. cero
Gaussiana	$e^{-(x \wedge 2 / \sigma \wedge 2)}$	Diferenciable, tipo pulso

Figura 2-3: Tabla de funciones No-lineales $g(x)$.

gación hacia atrás, mientras que las funciones discontinuas por ejemplo, la funciones de umbral son necesarias para generar una salida binaria.

La clasificación (2) se distingue por tener funciones, las cuales sólo tienen un valor significativo de salida cuando las entradas estan cerca del cero, dado que las funciones sólo cambian significativamente alrededor del cero.

La clasificación (3) se refiere a las funciones tipo escalón. Las funciones positivas que cambian de 0 en $-\infty$ a 1 en ∞ . Las funciones de promedio cero cambian de -1 en $-\infty$ a 1 en ∞ .

Algunas funciones estandar pueden observarse en la Fig. 2-3.

2.3 *Estructuras de las redes neuronales*

La forma como las neuronas de una red neuronal están interconectadas determina su estructura. Para propósitos de identificación y control, las estructuras más usadas son:

1. Redes de alimentación hacia adelante de una capa.
2. Redes de alimentación hacia adelante multicapa.
3. Redes de funciones radiales básicas.

4. Redes neuronales dinámicas.

2.3.1 *Redes de alimentación hacia adelante de una capa*

Esta es la forma mas simple de una red neuronal. Esta tiene sólo una capa de neuronas. La más conocida es llamada *Perceptrón*. Básicamente, ésta consta de una neurona con pesos sinápticos ajustables y una función de umbral.

El algoritmo de aprendizaje que ajusta los pesos de estas redes neuronales apareció por primera vez en [34], [35]. Ahí, es probado que los vectores de información, usados para entrenar el perceptrón son tomados de dos clases lineales separables, entonces el algoritmo de aprendizaje del perceptrón converge y define una superficie de decisión: un hiperplano que separa las dos clases. La respectiva prueba de convergencia es conocida como el teorema de convergencia del perceptrón.

El perceptrón básico es el llamado modelo de McCulloch-Pitts [22], en el cual la función de activación $\varphi(\cdot)$ es de límites extremos.

El propósito del perceptrón es el de clasificar la señal de entrada, con componentes: u_1, u_2, \dots, u_n en una o dos clases: C_1 o C_2 . La regla de decisión para la clasificación es asignar el punto correspondiente a una entrada u_1, u_2, \dots, u_n a la clase C_1 si la salida del perceptrón y es igual a $+1$ y la clase C_2 si es -1 .

Usualmente el umbral ρ es tratado como un peso sináptico conectado a una entrada fijada en -1 , así el vector de entrada es definido como:

$$(u(k))^T = (-1, u_1(k), u_2(k), \dots, u_n(k)) \quad (2.3)$$

donde k es la k -ésima entrada.

El sumador que produce la salida es calculado por:

$$v(k) = w^T(k) \quad u(k) = u^T(k) \quad w(k) \quad (2.4)$$

donde w es el vector de los pesos sinápticos.

Para cualquier k , en el espacio n -dimensional, la ecuación $w^T u$, con coordenadas u_1, u_2, \dots, u_n define un hiperplano que separa las entradas en dos clases: C_1 y C_2 . Si estas clases son linealmente separables, entonces existe un vector w tal que:

$$\begin{aligned} w^T u &\geq 0, \quad \forall u \in C_1 \\ &y \\ w^T u &< 0, \quad \forall u \in C_2 \end{aligned} \tag{2.5}$$

El algoritmo de aprendizaje adapta el vector de los pesos como sigue:

$$\begin{aligned} 1. - \quad w(k+1) &= w(k) && \text{si } w^T u \geq 0, \quad \forall u \in C_1 \\ & && \text{o } \text{si } w^T u < 0, \quad \text{para } u \in C_2 \\ 2. - \quad w(k+1) &= w(k) - \eta(k) u(k) && \text{si } w^T u \geq 0, \quad \text{para } u \in C_2 \\ & && \text{o} \\ & && \text{si } w^T u < 0, \quad \text{para } u \in C_1 \\ & && w(k+1) = w(k) + \eta(k) u(k) \end{aligned} \tag{2.6}$$

La convergencia de este algoritmo puede ser demostrada utilizando un argumento por contradicción (ver [?]).

Si se observa únicamente la salida $v(k)$ y se define el error como:

$$e(k) = d(k) - v(k) \tag{2.7}$$

El algoritmo de mínimos cuadrados (MC) puede ser aplicado para minimizar el error. La obtención de este algoritmo está basado en el método del gradiente descendiente. El resultado de la ley de aprendizaje para los pesos es la siguiente:

$$w(k+1) = w(k) + \eta e(k) u(k) \tag{2.8}$$

Puesto que el error depende linealmente de los pesos, este algoritmo asegura la ob-

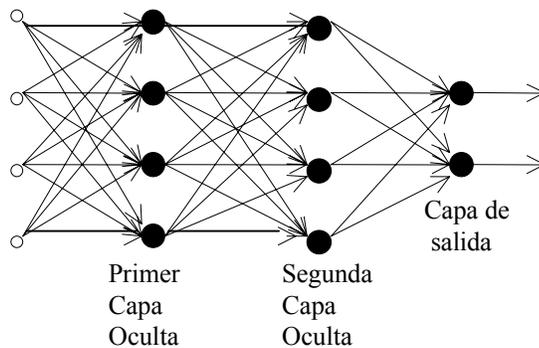


Figura 2-4: Perceptrón multicapa.

tención de un mínimo global.

2.3.2 *Redes de alimentación hacia adelante multicapa*

Estas se distinguen por la presencia de una o más capas ocultas, ver Fig. 2-4 cuyos nodos computacionales se llaman neuronas ocultas. Típicamente, las neuronas en cada capa tienen como señales de entrada, las señales de salida de la capa precedente. Si cada neurona en cada capa es conectada con todas las neuronas de las capas adyacentes, entonces la red neuronal es llamada totalmente conectada, en el caso opuesto, es llamada parcialmente conectada.

El perceptrón multicapa tiene las siguientes tres características:

1. La función de activación para cada neurona es suave a oposición a la de límites extremos usada en el perceptrón de una sola capa. Usualmente, esta función no-lineal es una sigmoide definida como:

$$\varphi_i(v_i) = \frac{1}{1 + e^{-v_i}} \quad (2.9)$$

2. La red está compuesta por una o más capas ocultas de neuronas.
3. La red presenta un alto grado de conectividad.

El preceptrón multicapa obtiene su poder computacional a través de la combinación de estas características y su habilidad de aprender de la experiencia. Sin embargo, la presencia de no-linealidades distribuidas y la alta conectividad de la red hacen el análisis teórico difícil de realizar.

2.3.3 *Redes de funciones radiales básicas*

Este tipo de redes neuronales tiene tres clases totalmente diferentes de capas:

1. La capa artificial de entrada de nodos de entrada.
2. La capa oculta, con un alto número de nodos (neuronas). Cada uno de estos nodos ejecuta una transformación no-lineal a la entrada, denotadas funciones radiales básicas.
3. La capa de salida, la cual es una combinación lineal de las salidas de la neuronas ocultas.

Las funciones radiales básicas fueron por primera vez introducidas para la solución de problemas de interpolación multivariables; los más recientes trabajos en este tema son examinados en [30]. La primera aplicación de las funciones radiales básicas en el diseño de redes neuronales fue reportado en [4]. Especiales contribuciones a la teoría, diseño, y aplicación de estas funciones a las redes neuronales se encuentran en [24], [33], [28].

2.3.4 *Redes neuronales dinámicas*

Este tipo de redes se distinguen de las redes neuronales estáticas en el que éstas tienen al menos un ciclo de retroalimentación. Estos ciclos involucran el uso del tiempo discreto, y de bifurcaciones compuestas por elementos de una unidad de retraso. Esta unidad se denota por q^{-1} , tal que $u(k-1) = q^{-1} u(k)$, con k indicando el k -ésimo muestreo en el tiempo. La ecuación de las redes neuronales dinámicas sería:

Tiempo discreto:

$$y(k+1) = g[y(k), y(k+1), \dots, y(k-n); u(k), u(k+1), \dots, u(k-m)] \quad (2.10)$$

Tiempo Continuo

$$\dot{\hat{x}} = A \hat{x} + w_1 \sigma(v_1 x) + w_2 \phi(v_2 x) u \quad (2.11)$$

donde $\sigma(\cdot)$ y $\phi(\cdot)$ son funciones sigmoideas, ver Fig.2-5.

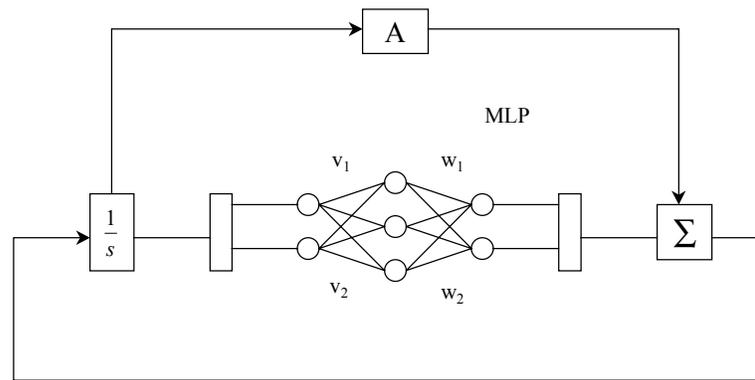


Figura 2-5: Esquema de una red neuronal dinámica.

Los ciclos de retroalimentación resultan en un comportamiento dinámico no-lineal debido a la función de activación no-lineal de las neuronas. De ahí que el término de las redes neuronales dinámicas (RND) describe mejor este tipo de estructura de las redes neuronales. Debido a estos hechos, las llamaremos redes neuronales dinámicas.

Estas pueden ofrecer grandes ventajas computacionales. De hecho, es bien sabido que un sumador lineal estático finito es equivalente a un sistema lineal retroalimentado de un solo polo, como se ve en la Fig.2-6.

De la Fig. 2-6, el sumador de salida es:

$$v(k) = u(k) + u(k-1) + \dots + u(k-n) = \sum_{i=0}^n u(k-i), \quad n \rightarrow \infty \quad (2.12)$$

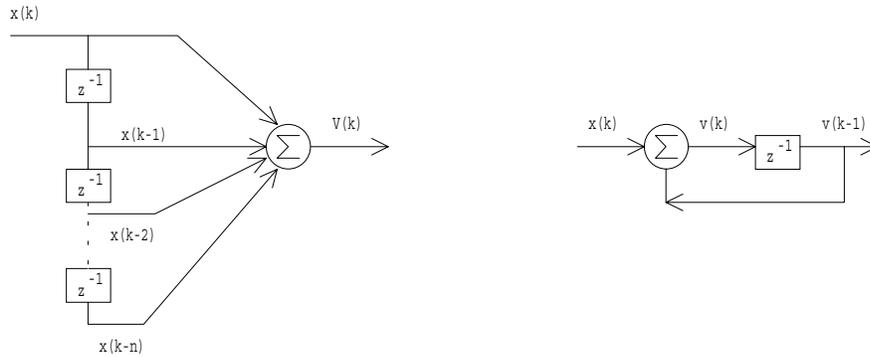


Figura 2-6: Sistemas dinámicos y estáticos equivalentes.

El sistema lineal es descrito por:

$$\begin{aligned}
 v(k) &= v(k-1) + u(k) \\
 \frac{v(k)}{u(k)} &= \frac{1}{1-q^{-1}} = 1 + q^{-1} + q^{-2} \dots q^{-n} \\
 v(k) &= u(k) + u(k-1) + \dots u(k-n)
 \end{aligned}
 \tag{2.13}$$

Es claro que las dos estructuras son equivalentes, pero desde el punto de vista computacional, el sistema con retroalimentación es equivalente a una muy grande, posiblemente infinita, estructura estática. Esta propiedad es muy interesante para identificación y control, y abre el camino para las aplicaciones de las redes neuronales dinámicas en estos campos.

2.4 *Redes Neuronales en Identificación*

Las redes neuronales tienen el potencial de ser aplicadas para modelar sistemas no-lineales. Una pregunta importante es si el sistema es identificable [27], i.e, si puede ser el sistema dinámico en consideración, adecuadamente representado sin ser necesario dar un modelo estructurado particular. La **Identificabilidad** en redes neuronales está relacionada a la unicidad de los pesos y si dos redes con diferentes parámetros pueden producir un comportamiento idéntico entrada/salida. Resultados en este tema se encuentra

en [40] para redes neuronales estáticas, y en [2] para las dinámicas.

Para representar los sistemas no-lineales con redes neuronales, una aproximación correcta es incrementar las entradas de la redes con señales correspondientes a sus entradas y salidas. Asumiendo que el sistema no-lineal está descrito por ver [15]:

$$y(k+1) = g[y(k), y(k+1), \dots, y(k-n); u(k), u(k+1), \dots, u(k-m)] \quad (2.14)$$

$$y, u \in \mathfrak{R}, m \leq n$$

Este modelo no considera las perturbaciones directamente. Para un método que incluya las perturbaciones, ver [7]. Casos especiales de este modelo son considerados en [3].

Una aproximación obvia del modelo del sistema es seleccionar la estructura de entrada/salida de la red neuronal para que sea la misma que tiene el sistema. Denotando la salida de la red neuronal como y_{nn} , existen dos posibles estrategias.

2.4.1 Estrategias de Identificación

Existen 2 tipos de modelos:

Modelo serie paralelo: En este caso, las salidas del sistema y son usadas como entradas de la red neuronal.

$$y_{nn}(k+1) = g[y(k), y(k+1), \dots, y(k-n); u(k), u(k+1), \dots, u(k-m)] \quad (2.15)$$

Dado que no existe recursividad en la ecuación, esta corresponde a una red neuronal estática como la que se muestra en la Fig.2-7.

Modelo paralelo: En este modelo las salidas pasadas de la red neuronal son usadas como componentes de la entrada de la red.

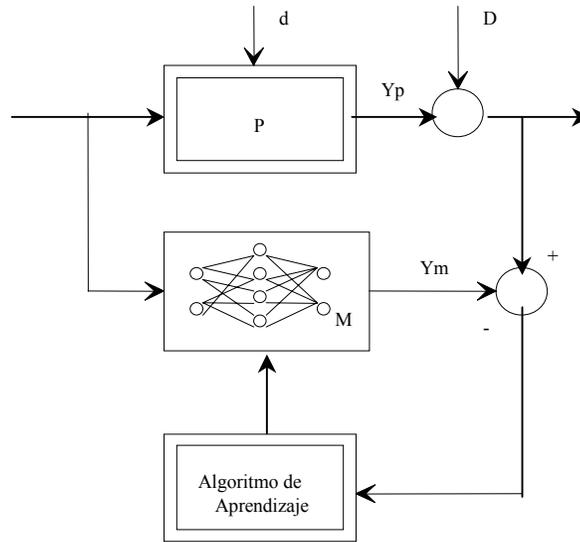


Figura 2-7: Esquema de identificación de una red dinámica.

$$y_{nn}(k+1) = g [y_{nn}(k), y_{nn}(k+1), \dots, y(k-n); u(k), u(k+1), \dots, u(k-m)] \quad (2.16)$$

Dado que existe una recursividad de y_{nn} en esta ecuación, esta corresponde a una red neuronal dinámica.

Observación 2.1 La regla de actualización de pesos de la estructura serie paralelo [36, 37] y la estructura paralelo son muy similares, la única diferencia es que la serie paralelo obtiene a la matriz de ganancia P de la solución de la siguiente ecuación de Lyapunov

$$PA + A^T P = -Q. \quad (2.17)$$

La paralela obtiene a la matriz de ganancia P de la solución de la ecuación de Riccati.

2.4.2 Aproximación de Perceptrón Dinámico Multicapa

Considere el siguiente perceptrón dinámico multicapa (PDM) en tiempo continuo.

$$\dot{\hat{x}}_t = A\hat{x}_t + W_{1,t}\sigma(V_{1,t}\hat{x}_t) + W_{2,t}\phi(V_{2,t}\hat{x}_t)u_t \quad (2.18)$$

donde $\forall t \in [0, \infty)$, el vector $\hat{x}_t \in \mathfrak{R}^n$ es el estado de la red neuronal, $u_t \in \mathfrak{R}^k$ es la entrada. Si $k < m$, $u_i = 0$, para $i = k+1, \dots, m$. La matriz $A \in \mathfrak{R}^{n \times n}$ es una matriz estable, la cual será especificada despues. Las matrices $W_{1,t} \in \mathfrak{R}^{n \times m}$ y $W_{2,t} \in \mathfrak{R}^{n \times m}$ son los pesos de las capas de salida. $V_1 \in \mathfrak{R}^{m \times n}$ y $V_2 \in \mathfrak{R}^{m \times n}$ son los pesos de las capas ocultas. $\sigma(\cdot) \in \mathfrak{R}^m$ son funciones vectoriales sigmoidales, $\phi(\cdot)$ es $\mathfrak{R}^{m \times m}$ una matriz diagonal, i.e.,

$$\phi(\cdot) = \text{diag}[\phi_1(V_{2,t}\hat{x}_t)_1 \cdots \phi_m(V_{2,t}\hat{x}_t)_m]. \quad (2.19)$$

Los elementos de $\sigma_i(\cdot)$ (como lo es $\phi_i(\cdot)$) son usualmente funciones sigmoidales, i.e.,

$$\sigma_i(x_i) = \frac{a_i}{1 + e^{-b_i x_i}} - c_i. \quad (2.20)$$

La estructura del perceptrón dinámico se muestra en la Fig.2-8. Las redes neuronales dinámicas convecionales (RND) son las redes neuronales que corresponden al caso:

$$m = n \quad \text{y} \quad V_1 = V_2 = I. \quad (2.21)$$

Usando el PDM (2.18) para identificar el siguiente sistema no-lineal.

$$\dot{x}_t = f(x_t, u_t), \quad \forall t \in [0, \infty). \quad (2.22)$$

donde $x_t \in \mathfrak{R}^n$, $u_t \in \mathfrak{R}^m$, $f: \mathfrak{R}^{n+m} \rightarrow \mathfrak{R}^n$ es un campo vectorial suave continuo definido en un conjunto compacto $\Theta \in \mathfrak{R}^{n+m}$. Se asume que $f(x_t, u_t, t)$ satisface la condición de Lipschitz, i.e., existe una constante $l > 0$ tal que para todo par de puntos $(x_{1,t}, u_t)$,

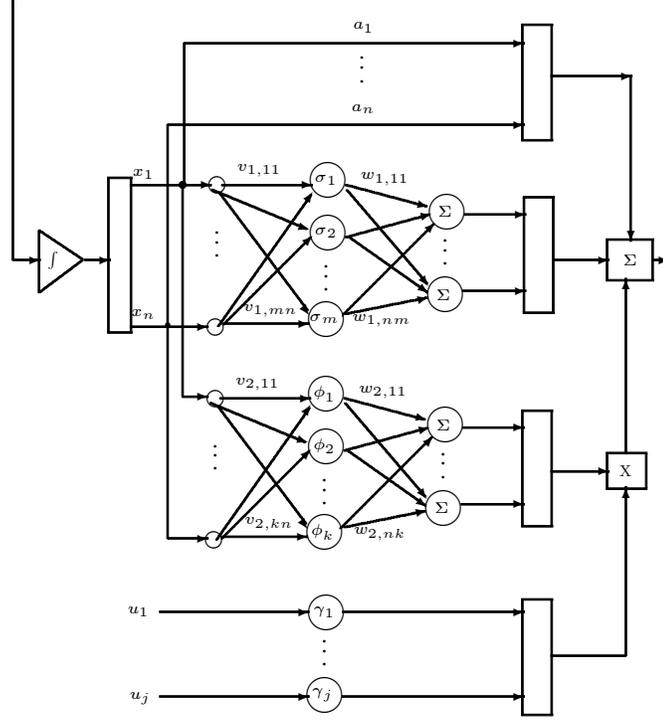


Figura 2-8: Estructura general del perceptrón dinámico multicapa.

$$(x_{2,t}, u_t) \in \Theta$$

$$|f(x_{1,t}, u_t) - f(x_{2,t}, u_t)| \leq l |x_{1,t} - x_{2,t}| \quad (2.23)$$

Problema 2.1 *El problema de identificación del sistema dinámico consiste en determinar si existe un perceptrón dinámico multicapa (PDM) tal que (2.18) pueda aproximar el comportamiento entrada-salida del sistema dinámico (2.22).*

Si se define $W_t := [W_{1,t} \ W_{2,t}]$, $\Phi_{V_t}(\hat{x}_t, u_t) := [\sigma(V_{1,t}\hat{x}_t) \ \phi(V_{2,t}\hat{x}_t)u_t]^T$, así (2.18) y (2.22) quedarían

$$\begin{aligned} \hat{\dot{x}}_t &= A\hat{x}_t + W_t\Phi_{V_t}(\hat{x}_t, u_t) \\ \dot{x}_t &= Ax_t + G(x_t, u_t) \end{aligned} \quad (2.24)$$

donde $G(x_t, u_t) := f(x_t, u_t) - Ax_t$. Obsérvese $W_t\Phi_{V_t}(\hat{x}_t, u_t)$ es una PDM en tiempo continuo, del teorema Stone-Weierstrass si la capa oculta $(V_{1,t} \ V_{2,t})$ es lo suficientemente grande (m es suficientemente grande), $W_t\Phi_{V_t}(x_t, u_t)$ puede aproximarse $G(x_t, u_t)$ con

cualquier grado de exactitud para todo $(x_t, u_t) \in \Theta$. Por ello existe $W_t = W^* := [W_1^* \ W_2^*]$ y $V_t = V^* := [V_1^* \ V_2^*]$, tal que

$$\sup_{(\hat{x}, u_t) \in \Theta} |G(x_t, u_t) - W^* \Phi_{V^*}(x_t, u_t)| \leq \delta \quad (2.25)$$

donde δ es cualquier constante pequeña. Basado en las suposiciones anteriores se obtiene el siguiente nuevo resultado.

Teorema 2.1 *Sea el sistema dinámico (2.22) y el perceptrón dinámico multicapa (2.18) inician en el mismo estado $x_0 = \hat{x}_0$. Para cualquier $\varepsilon > 0$ y un tiempo finito $T > 0$, existe un entero m tal que el estado del PDM \hat{x}_t con pesos $[W_1^* \ W_2^* \ V_1^* \ V_2^*]$ satisface*

$$\sup_{0 \leq t \leq T} |x_t - \hat{x}_t| < \varepsilon \quad (2.26)$$

Prueba. Se define el error de identificación como:

$$\Delta_t := x_t - \hat{x}_t \quad (2.27)$$

De (2.24) se sabe que existe un entero m y los pesos $[W_1^* \ W_2^* \ V_1^* \ V_2^*]$ tal que

$$\begin{aligned} \dot{\hat{x}}_t &= A\hat{x}_t + W^* \Phi_{V^*}(\hat{x}_t, u_t) \\ \dot{\Delta}_t &= A\Delta_t + G(x_t, u_t) - W^* \Phi_{V^*}(\hat{x}_t, u_t) \end{aligned} \quad (2.28)$$

Dado que $x_0 = \hat{x}_0$, se tiene que $\Delta_0 = 0$. La solución de la ecuación diferencial (2.28) es

$$\begin{aligned} \Delta_t &= \int_0^t e^{A(t-\tau)} [G(x_\tau, u_\tau) - W^* \Phi_{V^*}(\hat{x}_\tau, u_\tau)] d\tau \\ &= \int_0^t e^{A(t-\tau)} [W^* \Phi_{V^*}(x_\tau, u_\tau) - W^* \Phi_{V^*}(\hat{x}_\tau, u_\tau)] d\tau \\ &\quad + \int_0^t e^{A(t-\tau)} [G(x_\tau, u_\tau) - W^* \Phi_{V^*}(x_\tau, u_\tau)] d\tau \end{aligned} \quad (2.29)$$

Desde que A es una matriz estable, existe una constante positiva α tal que

$$\|e^{At}\| \leq e^{-\alpha t}, \quad \alpha > 0$$

Asi (2.29) se convierte

$$\begin{aligned} |\Delta_t| &\leq \int_0^t \|e^{A(t-\tau)}\| \|W^*\| |\Phi_{V^*}(x_\tau, u_\tau) - \Phi_{V^*}(\hat{x}_\tau, u_\tau)| d\tau \\ &\quad + \int_0^t \|e^{A(t-\tau)}\| |G(x_\tau, u_\tau) - W^* \Phi_{V^*}(x_\tau, u_\tau)| d\tau \end{aligned}$$

Usando la condición de Lipschitz (2.23) y el teorema de aproximación de una funcion (2.25) se tiene

$$|\Delta_t| \leq \int_0^t e^{-\alpha(t-\tau)} L |x_\tau - \hat{x}_\tau| d\tau + \int_0^t e^{-\alpha(t-\tau)} \delta d\tau = \frac{\delta}{\alpha} (1 - e^{-\alpha t}) + \int_0^t e^{-\alpha(t-\tau)} L |\Delta_t| d\tau$$

donde $L := l \|W^*\|$. Dado que $e^{-\alpha t} \leq 1$, para $\alpha t > 0$. Utilizando la desigualdad de Gronwall-Bellman [18], se tiene

$$\begin{aligned} |\Delta_t| &\leq \frac{\delta}{\alpha} + L \frac{\delta}{\alpha} \int_0^t (1 - e^{-\alpha s}) e^{-\alpha(t-\tau)} e^{\int_\tau^t L e^{-\alpha(t-s)} ds} d\tau \\ &= \frac{\delta}{\alpha} + L \frac{\delta}{\alpha} \int_0^t (1 - e^{-\alpha s}) e^{-\alpha(t-\tau)} e^{\frac{L}{\alpha}(1 - e^{-\alpha(t-\tau)})} d\tau \end{aligned}$$

Aplicando la siguiente desigualdad

$$1 - e^{-x} \leq x, \quad x \geq 0$$

se tiene

$$\begin{aligned} |\Delta_t| &\leq \frac{\delta}{\alpha} + L \frac{\delta}{\alpha} \int_0^t (1 - e^{-\alpha s}) e^{-\alpha(t-\tau)} e^{L(t-\tau)} d\tau \\ &= \frac{\delta}{\alpha} + L \frac{\delta}{\alpha} \int_0^t (1 - e^{-\alpha s}) (1 - e^{-\alpha \tau}) e^{(L-\alpha)(t-\tau)} d\tau \end{aligned}$$

$$\begin{aligned}
&= \frac{\delta}{\alpha} + \frac{(\alpha - L)e^{-\alpha t} - \alpha e^{-(\alpha-L)t} + L}{\alpha(\alpha - L)}\delta \\
&= \frac{\delta}{\alpha} + \left[\frac{1}{\alpha}e^{-\alpha t} - \frac{e^{-(\alpha-L)t}}{(\alpha - L)} + \frac{L}{\alpha(\alpha - L)} \right] \delta \\
&\leq \frac{\delta}{\alpha} + \left[\frac{1}{\alpha} - \frac{e^{(\alpha-L)t}}{(\alpha - L)} + \frac{L}{\alpha(\alpha - L)} \right] \delta \\
&= \frac{\delta}{\alpha} + \frac{\delta}{\alpha - L} [1 - e^{-(\alpha-L)t}] \\
&\leq \frac{\delta}{\alpha} + \frac{\delta}{\alpha - L} = \frac{(2\alpha - L)}{\alpha(\alpha - L)}\delta
\end{aligned}$$

Si se define

$$\varepsilon := \frac{(2\alpha - L)}{\alpha(\alpha - L)}\delta$$

y se selecciona $\alpha \geq L$, así $\varepsilon > 0$, por lo tanto $|\Delta_t| = |x_t - \hat{x}_t| \leq \varepsilon$. Del teorema de aproximación de una función (2.25) se conoce que δ puede ser cualquier constante pequeña. Dado que α y L son constantes, así ε puede ser tan bien cualquier constante pequeña. Esto es (2.26). ■

Observación 2.2 Si la condición inicial de la red neuronal y el sistema dinámico son distintos, $x_0 \neq \hat{x}_0$, La solución de la ecuación diferencial (2.28) es

$$\Delta_t = \int_0^t e^{A(t-\tau)} [G(x_\tau, u_\tau) - W^* \Phi_{V^*}(\hat{x}_\tau, u_\tau)] d\tau + \Delta_0 e^{At}$$

El término adicional $\Delta_0 e^{At}$ que influye en el resultado final es

$$|\Delta_t| \leq \frac{(2\alpha - L)}{\alpha(\alpha - L)}\delta + \frac{1}{L}e^{-(\alpha-L)t} |\Delta_0|.$$

Dado que $(\alpha - L)t > 0$, l y Δ_0 son constantes

$$\lim_{t \rightarrow \infty} \frac{1}{L} e^{-(\alpha-L)t} |\Delta_0| = 0$$

Así la influencia de la condición inicial del PDM decae exponencialmente a cero

Algoritmo de propagación hacia atrás (Backpropagation)

El algoritmo de aprendizaje usado para ajustar los pesos sinápticos del perceptrón multicapa estático es conocido como *backpropagation*. La idea básica fue por primera vez descrita en [15]. Este algoritmo provee un método eficiente para el entrenamiento del perceptrón multi-capas.

El error en la salida de la j -ésima neurona de la capa de salida, está dado por:

$$e_j(k) = d_j(k) - y_j(k) \quad (2.30)$$

donde:

d_j es la salida deseada.

y_j es la neurona de salida.

k indica la k -ésima muestra.

La suma instantánea del cuadrado de los errores de salida está dado por:

$$\mathcal{E}(k) = \frac{1}{2} \sum_{j=1}^l e_j(k) \quad (2.31)$$

donde l es el número de neuronas de la capa de salida.

El error cuadrático promedio se obtiene a sumar $\mathcal{E}(k)$ para todas las muestras (una iteración) y normalizarlo con respecto al tamaño de la iteración.

$$\mathcal{E}_{av} = \frac{1}{N} \sum_{k=1}^N \mathcal{E}(k) \quad (2.32)$$

con N como el número de muestras, que forman cada iteración.

Usando el gradiente descendiente, los pesos de la conexiones entre la i -ésima neurona y la j -ésima neurona se actualiza como:

$$\Delta w_{ji}(k) = w_{ji}(k+1) - w_{ji}(k) = -\eta \frac{\partial \mathcal{E}(k)}{\partial w_{ji}(k)}. \quad (2.33)$$

La corrección que realiza $\Delta w_{ji}(k)$ es conocida como la regla delta, donde el término $\frac{\partial \mathcal{E}(k)}{\partial w_{ji}(k)}$ puede ser calculado como:

$$\frac{\partial \mathcal{E}(k)}{\partial w_{ji}(k)} = \frac{\partial \mathcal{E}(k)}{\partial e_j(k)} \frac{\partial e_j(k)}{\partial y_j(k)} \frac{\partial y_j(k)}{\partial v_j(k)} \frac{\partial v_j(k)}{\partial w_{ji}(k)}. \quad (2.34)$$

Las derivadas parciales estan dadas por:

$$\begin{aligned} \frac{\partial \mathcal{E}(k)}{\partial e_j(k)} &= e_j(k), & \frac{\partial e_j(k)}{\partial y_j(k)} &= -1, \\ \frac{\partial y_j(k)}{\partial v_j(k)} &= \varphi'_j(v_j(k)) & \text{con } \varphi'_j(\beta) &= \frac{\partial \varphi_j(\beta)}{\partial \beta} \\ & & \frac{\partial v_j(k)}{\partial w_{ji}(k)} &= y_i(k). \end{aligned} \quad (2.35)$$

Asi la regla delta puede ser reescrita como:

$$\Delta w_{ji}(k) = \eta \delta_j(k) y_i(k) \quad (2.36)$$

con:

$$\delta_j(k) = -\frac{\partial \mathcal{E}(k)}{\partial y_j(k)} \frac{\partial y_j(k)}{\partial v_j(k)}. \quad (2.37)$$

Dos casos pueden ser distinguidos: la j -ésima neurona está en la capa de salida o se encuentra en la capa oculta.

Caso 1: Si la j -ésima neurona está localizada en la capa de salida es correcto calcular $\delta_j(k)$.

Caso 2: Si la j -ésima neurona está localizada en la capa oculta, no hay una respuesta deseada para esta neurona. Así, la señal de error tiene que ser derivada recursiva-

mente en términos del error para todas la neuronas a la que esté conectada. En este caso es posible establecer la siguiente ecuación:

$$\delta_j(k) = \frac{\partial y_j(k)}{\partial v_j(k)} \sum_{n=1}^m \delta_n(k) w_{nj}(k) \quad (2.38)$$

donde n indica la n -ésima neurona a la que la j -ésima neurona esta conectada y m es el número total de estas neuronas.

Este es un resumen del algoritmo tomado de [13], donde la derivación completa es presentada. Este algoritmo se ha convertido en el mas popular para el entrenamiento del perceptrón multicapa. Es muy eficiente y tiene la capacidad de clasificar información que debe ser separada de forma no-lineal. El algoritmo es una técnica de gradiente, implementando sólo la minimización en una sola dirección, la cual puede no ser la óptima. Asi, no es posible demostrar la convergencia del algoritmo a un mínimo global.

Como podemos ver las redes neuronales nos permiten obtener un mapeo entrada-salida del comportamiento del sistema, con lo cual no es necesario obtener un modelo matematico del mismo, esto, puede ser una ventaja al tratar con sistemas muy complejos de los cuales es muy dificil obtener su modelo matemático. Por otra parte las redes neuronales nos permiten implementar un controlador para un sistema a partir de la informacion obtenida de por la red.

2.5 *Redes Neuronales en Control*

Con una referencia específica en las redes neuronales en control las siguientes características y propiedades son importantes:

1. Sistemas no-lineales: Las redes neuronales son una gran promesa en el campo de los problemas del control no-lineal. Esta proviene de su teórica habilidad para aproximar mapeos no-lineales arbitrarios.

2. Procesamiento Distribuido Paralelo: Las redes neuronales tienen una estructura sumamente paralela que les permiten inmediatamente realizar una implementación paralela. Con tal implementación puede esperarse alcanzar un mayor grado de tolerancia a errores que los esquemas convencionales.
3. Implementación en Hardware: Este punto está muy relacionado con el punto anterior. Las redes no sólo pueden ser implementadas en paralelo, algunos vendedores han introducido recientemente implementaciones utilizando un hardware dedicado VLSI. Este proporciona velocidad adicional y incrementa el número de redes que pueden ser implementadas.
4. Aprendizaje y Adaptación: Las redes neuronales son entrenadas utilizando datos pasados del sistema en estudio. Una red propiamente entrenada tiene la habilidad de generalizar cuando se le introducen datos diferentes a los utilizados para su entrenamiento. Las redes pueden ser además adaptadas en línea.
5. Fusión de Datos: Las redes neuronales pueden operar simultáneamente con datos cualitativos y cuantitativos en este punto las redes se encuentran en algún lugar en medio del campo entre los sistemas tradicionales de ingeniería (Datos cuantitativos) y técnicas de procesamiento del campo de la inteligencia artificial (datos simbólicos).
6. Sistemas multivariados: Las redes neuronales naturalmente procesan muchas entradas y tiene muchas salidas; por lo que ellas pueden fácilmente ser aplicadas para sistemas multivariables.

Desde el punto de vista de la teoría de control la habilidad de las redes neuronales de tratar con los sistemas no-lineales es muy importante. La gran diversidad de los sistemas no-lineales es la primera razón del porque no una teoría general y sistemática aplicable para el diseño de control no-lineal no ha sido todavía desarrollada. Sin embargo, existen algunos métodos tradicionales para el análisis y síntesis de controladores no-lineales para

clases específicas de sistemas no-lineales, como lo son: métodos en el plano de fase, técnicas de linealización y funciones descriptivas.

Sin embargo, es la habilidad de las redes neuronales para representar mapeos no-lineales, y de modelar sistemas no-lineales, la característica más importante para usar a las redes neuronales en la realización de controladores no-lineales. Una vista de las relaciones generales entre los campos de la ciencia del control y las redes neuronales es mostrada en la Fig .2-9. Los cuadros blancos son utilizados cuando una similitud obvia no exista.

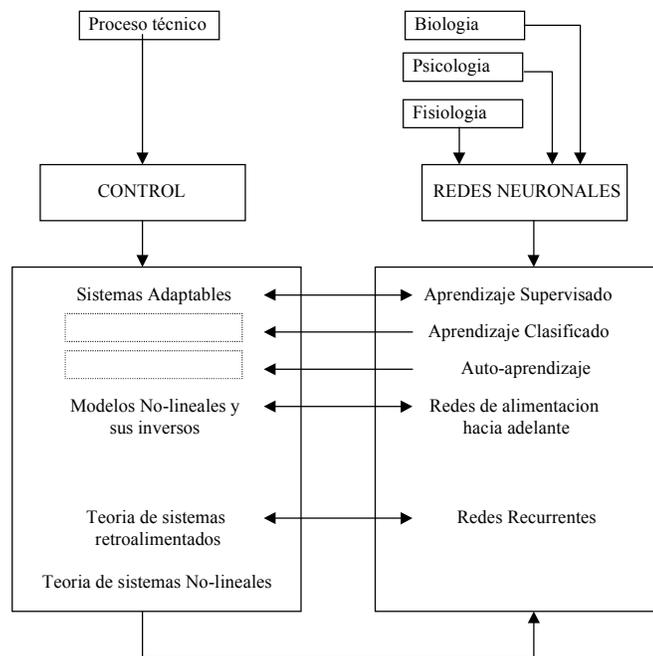


Figura 2-9: Relación entre las redes neuronales y el control.

2.5.1 Estructuras de Control

Una gran cantidad de estructuras de control han sido propuestas. Para una revisión reciente y completa, ver [1]. En la literatura de arquitecturas de redes neuronales aplicadas en el control han sido propuestas y usadas una gran cantidad de estructuras de control;

aquí se muestran dos de las más importantes

Control Directo

En esta estructura [41], el neuro-controlador es entrenado sin tener el modelo de la planta. En este esquema, ver Fig. (2-10) se busca que el resultado del lazo cerrado que contiene a la planta y al sistema de como resultado un mapeo identidad. La acción de control busca que la planta siga la señal de referencia x_d .

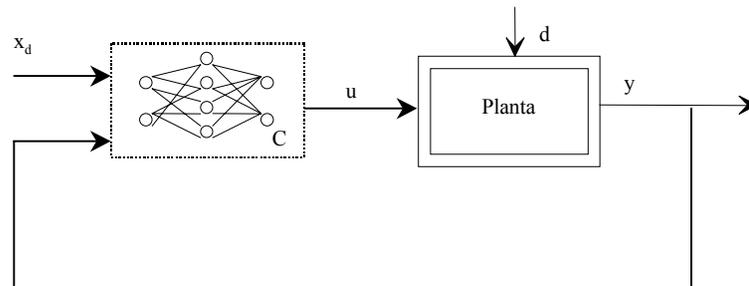


Figura 2-10: Esquema general del control directo.

Control Indirecto

En esta estructura [41], primero un neuro-identificador estima el comportamiento de la planta, usando como información únicamente la salida del sistema, luego el neuro controlador es diseñado en base al modelo identificado, ver Fig. (2-11). La acción de control busca que la planta siga la señal de referencia x_d .

Observación 2.3 *Notese que en las figs. (2-10), (2-11) dado que se tratan de RND, solo es necesario usar la información dada por la salida de la planta (y) y no se requiere de los pares entrada/salida como el caso de las RNE.*

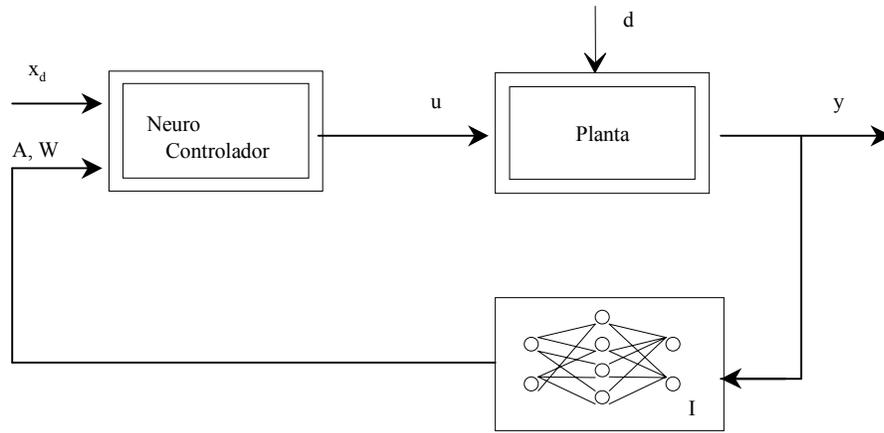


Figura 2-11: Esquema general del control indirecto.

2.5.2 *Control Adaptable*

Existe una diferencia entre el control adaptable directo y indirecto en el contexto de esta estructura. En los métodos indirectos, primero el sistema es identificado desde las medidas entrada/salida de la planta y después el controlador es adaptado basándose en el modelo identificado. En los métodos directos el controlador es aplicado sin realizar una identificación previa del sistema, por lo que esta se lleva a cabo en línea. Las redes neuronales pueden ser utilizadas en ambos métodos.

Parte del control adaptable neuronal es basado en el método indirecto. En este método primero, un modelo del sistema es obtenido por la red neuronal en línea utilizando medidas de la planta y entonces una de las estructuras de control citadas anteriormente es implementada encima de este modelo adaptable neuronal. Uno de los primeros resultados en redes neuronales adaptables no-lineales es [3], donde en base a modelos neuronales específicos, el controlador basado en el modelo adaptativo indirecto es implementado. En [38], un control adaptable IMC es desarrollado usando redes neuronales RNFB. Aplicaciones para control de robots, donde las RNE son usadas para estimar adaptativamente parte de la dinámica del robot son presentadas en [21]. Basado en un modelo identificado del sistema en línea por una RND, explicada anteriormente, en [36], la ley de control es

construida para asegurar el seguimiento del modelo de referencia lineal.

Estimando un control neuronal adaptable directo, en [39], un controlador adaptable es desarrollado usando redes neuronales RBF. En [41], una combinacion de ambos métodos es usado; de hecho, un modelo de una RND es adaptado en línea, también como un neuro-controlador dinámico.

Capítulo 3

Modelado del sistema no-lineal TORA

3.1 *Introducción*

El sistema mostrado en la Fig. 3-1 representa un oscilador traslacional con actuador rotacional (TORA) [6]. Fue introducido por primera vez en [45], siendo utilizado como un sistema no-lineal estándar en un número reciente de trabajos [6], [45] con el propósito de comparación y realizar pruebas de comparación entre diferentes metodologías de control.

3.2 *Modelo Lagrangiano*

El oscilador consiste en un carro de masa M conectado con una pared fija por un resorte lineal con una constante de rigidez k . El carro está limitado a una dimensión de movimiento. El actuador que está unido al carro tiene una masa m y un momento de inercia I desde su centro de masa, que está localizado a una distancia e desde el punto en el cual la masa rota. El movimiento ocurre en el plano horizontal, por lo que la fuerza gravitacional no requiere ser considerada (ver Fig. 3-1), donde N denota el control del torque aplicada a la masa m , y F es una fuerza de perturbación sobre el carro.

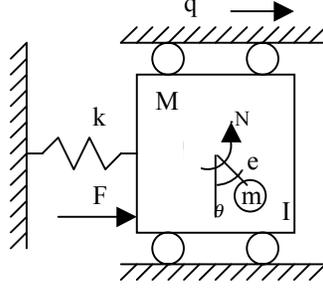


Figura 3-1: Oscilador Traslacional con Actuador Rotacional (TORA).

Sea q y \dot{q} la posición traslacional y la velocidad del carro, y sea θ y $\dot{\theta}$ la posición angular y la velocidad angular de masa m , donde $\theta = 0$ es perpendicular al movimiento del carro, y $\theta = 90^\circ$ está alineado con la dirección positiva de q . Las ecuaciones del movimiento están dadas por:

$$\begin{aligned} (M + m) \ddot{q} + kq &= -me \left(\ddot{\theta} \cos \theta - \dot{\theta}^2 \sin \theta \right) + F \\ (I + me^2) \ddot{\theta} &= -me \ddot{q} \cos \theta + N \end{aligned} \quad (3.1)$$

3.3 Modelo Normalizado

Usando las normalizaciones [45]:

$$\begin{aligned} \xi &\triangleq \sqrt{\frac{M + m}{I + me^2}} q, & \tau &\triangleq \sqrt{\frac{k}{M + m}} t \\ u &\triangleq \frac{M + m}{k(I + me^2)} N, & w &\triangleq \frac{1}{k} \sqrt{\frac{M + m}{I + me^2}} F, \end{aligned}$$

Las ecuaciones de movimiento se convierten en:

$$\begin{aligned}\ddot{\xi} + \xi &= \varepsilon \left(\dot{\theta}^2 \sin \theta - \ddot{\theta} \cos \theta \right) + w \\ \ddot{\theta} &= -\varepsilon \ddot{\xi} \cos \theta + u\end{aligned}\quad (3.2)$$

donde ξ es la posición normalizada de la posición del carro, y w y u representan una perturbación adimensional, el control del torque, respectivamente. En las ecuaciones normalizadas, el símbolo (\cdot) representa la diferenciación con respecto al tiempo normalizado τ . La unión entre el movimiento traslacional y el rotacional esta representado por el parámetro ε el cual es definido por (un valor típico $\varepsilon = 0.1$ es utilizado en las simulaciones):

$$\varepsilon \triangleq \frac{me}{\sqrt{(I + me^2)(M + m)}} \quad (3.3)$$

3.4 Modelo de Primer Orden

Tomando $x = [x_1, x_2, x_3, x_4]^T = [\xi, \dot{\xi}, \theta, \dot{\theta}]^T$, las ecuaciones adimensionales del movimiento de primer orden estarían dadas por:

$$\dot{x} = f(x) + g(x)u + d(x)w, \quad (3.4)$$

donde

$$f(x) = \begin{bmatrix} x_2 \\ \frac{-x_1 + \varepsilon x_4^2 \sin x_3}{1 - \varepsilon^2 \cos^2 x_3} \\ x_4 \\ \frac{\varepsilon \cos x_3 (x_1 - \varepsilon x_4^2 \sin x_3)}{1 - \varepsilon^2 \cos^2 x_3} \end{bmatrix} = \begin{bmatrix} f_1 \\ f_2 \\ f_3 \\ f_4 \end{bmatrix}, \quad g(x) = \begin{bmatrix} 0 \\ \frac{-\varepsilon \cos x_3}{1 - \varepsilon^2 \cos^2 x_3} \\ 0 \\ \frac{1}{1 - \varepsilon^2 \cos^2 x_3} \end{bmatrix} = \begin{bmatrix} 0 \\ g_1 \\ 0 \\ g_2 \end{bmatrix} \quad (3.5)$$

$$d(x) = \begin{bmatrix} 0 \\ \frac{1}{1 - \varepsilon^2 \cos^2 x_3} \\ 0 \\ \frac{-\varepsilon \cos x_3}{1 - \varepsilon^2 \cos^2 x_3} \end{bmatrix}$$

3.5 Modelo en Cascada

Definición 3.1 *Un sistema no-lineal*

$$\begin{aligned} \dot{x} &= f(x) + g(x)u \\ y &= h(x) \end{aligned} \tag{3.6}$$

donde $f : D \rightarrow \mathfrak{R}^n, g : D \rightarrow \mathfrak{R}^n$ y $h : D \rightarrow \mathfrak{R}$ son suficientemente suaves en el dominio $D \subset \mathfrak{R}^n$, se dice que tienen un grado relativo r , $1 \leq r \leq n$, en una region $D_o \subset D$ Si

$$\frac{\partial \psi_i}{\partial x} g(x) = 0, i = 1, 2, \dots, r-1; \frac{\partial \psi_r}{\partial x} g(x) \neq 0$$

para todo $x \in D_o$ donde $\psi_1(x) = h(x)$ y $\psi_{i+1}(x) = \frac{\partial \psi_i}{\partial x} f(x), i = 1, 2, \dots, r-1$

Si el sistema (3.6) tiene grado relativo r , entonces es linealizable entrada-salida. Si tiene grado relativo n , entonces es linealizable entrada-estado y entrada-salida.

Ahora se verificará esta condición para el modelo TORA usando (3.4), (3.5) y $y = h(x) = x_3$:

$$1. - L_g h(x)$$

$$L_g h(x) = \frac{\partial h}{\partial x} g(x) = \begin{bmatrix} 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} 0 \\ g_1 \\ 0 \\ g_2 \end{bmatrix} = 0$$

2. - $L_g L_f h(x)$

$$L_f h(x) = \frac{\partial h}{\partial x} f(x) \begin{bmatrix} 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} f_1 \\ f_2 \\ f_3 \\ f_4 \end{bmatrix} = f_3 = x_4$$

$$L_g L_f h(x) = \frac{\partial(L_f h)}{\partial x} f(x) = \begin{bmatrix} 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 0 \\ g_1 \\ 0 \\ g_2 \end{bmatrix} = g_2 \neq 0$$

La planta tiene un grado relativo $r=2$, que es más pequeño que el orden de la planta $n=4$, por lo que es linealizable entrada-salida. Para transformar (3.4) en la forma de cascada con la parte lineal de máxima dimensión, se aplica el siguiente procedimiento:

$$L_g z_i = 0, \quad L_{\text{ad}_f g(x)} z_i = 0, \quad i = 1, 2$$

donde

$$\text{ad}_f g(x) = \frac{\partial g}{\partial x} f - \frac{\partial f}{\partial x} g$$

$$\frac{\partial g}{\partial x} f = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & \varepsilon \frac{\sin x_3}{1-\varepsilon^2 \cos^2 x_3} + 2\varepsilon^3 \frac{\cos^2 x_3}{(1-\varepsilon^2 \cos^2 x_3)^2} \sin x_3 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & -\frac{2}{(1-\varepsilon^2 \cos^2 x_3)^2} \varepsilon^2 \cos x_3 \sin x_3 & 0 \end{bmatrix} \begin{bmatrix} x_2 \\ \frac{-x_1 + \varepsilon x_4^2 \sin x_3}{1-\varepsilon^2 \cos^2 x_3} \\ x_4 \\ \frac{\varepsilon \cos x_3 (x_1 - \varepsilon x_4^2 \sin x_3)}{1-\varepsilon^2 \cos^2 x_3} \end{bmatrix} =$$

$$\begin{bmatrix} 0 \\ \left(\varepsilon \frac{\sin x_3}{1-\varepsilon^2 \cos^2 x_3} + 2\varepsilon^3 \frac{\cos^2 x_3}{(1-\varepsilon^2 \cos^2 x_3)^2} \sin x_3 \right) x_4 \\ 0 \\ -\frac{2}{(1-\varepsilon^2 \cos^2 x_3)^2} \varepsilon^2 (\cos x_3 \sin x_3) x_4 \end{bmatrix} = \begin{bmatrix} 0 \\ \varepsilon (\sin x_3) (1 + \varepsilon^2 \cos^2 x_3) \frac{x_4}{1-2\varepsilon^2 \cos^2 x_3 + \varepsilon^4 \cos^4 x_3} \\ 0 \\ -2\varepsilon^2 (\cos x_3 \sin x_3) \frac{x_4}{1-2\varepsilon^2 \cos^2 x_3 + \varepsilon^4 \cos^4 x_3} \end{bmatrix}$$

$$\frac{\partial f}{\partial x} g = \begin{bmatrix} 0 & 1 & 0 & 0 \\ -\frac{1}{1-\varepsilon^2 \cos^2 x_3} & 0 & F_1 & 2\varepsilon x_4 \frac{\sin x_3}{1-\varepsilon^2 \cos^2 x_3} \\ 0 & 0 & 0 & 1 \\ \varepsilon \frac{\cos x_3}{1-\varepsilon^2 \cos^2 x_3} & 0 & F_2 & -2\varepsilon^2 x_4 (\cos x_3) \frac{\sin x_3}{1-\varepsilon^2 \cos^2 x_3} \end{bmatrix} \begin{bmatrix} 0 \\ \frac{-\varepsilon \cos x_3}{1-\varepsilon^2 \cos^2 x_3} \\ 0 \\ \frac{1}{1-\varepsilon^2 \cos^2 x_3} \end{bmatrix} =$$

$$\begin{bmatrix} -\varepsilon \frac{\cos x_3}{1-\varepsilon^2 \cos^2 x_3} \\ 2\varepsilon x_4 \frac{\sin x_3}{(1-\varepsilon^2 \cos^2 x_3)^2} \\ \frac{1}{1-\varepsilon^2 \cos^2 x_3} \\ -2\varepsilon^2 x_4 (\cos x_3) \frac{\sin x_3}{(1-\varepsilon^2 \cos^2 x_3)^2} \end{bmatrix} = \begin{bmatrix} \varepsilon \frac{\cos x_3}{-1+\varepsilon^2 \cos^2 x_3} \\ 2\varepsilon x_4 \frac{\sin x_3}{1-2\varepsilon^2 \cos^2 x_3 + \varepsilon^4 \cos^4 x_3} \\ -\frac{1}{-1+\varepsilon^2 \cos^2 x_3} \\ -2\varepsilon^2 (\cos x_3 \sin x_3) \frac{x_4}{1-2\varepsilon^2 \cos^2 x_3 + \varepsilon^4 \cos^4 x_3} \end{bmatrix}$$

$$\text{donde } F_1 = \varepsilon x_4^2 \frac{\cos x_3}{1-\varepsilon^2 \cos^2 x_3} - 2 \frac{-x_1 + \varepsilon x_4^2 \sin x_3}{(1-\varepsilon^2 \cos^2 x_3)^2} \varepsilon^2 \cos x_3 \sin x_3,$$

$$F_2 = \frac{-\varepsilon x_1 \sin x_3 + \varepsilon^2 x_4^2 \sin^2 x_3 - \varepsilon^2 x_4^2 \cos^2 x_3}{1-\varepsilon^2 \cos^2 x_3} - 2 \frac{\varepsilon x_1 \cos x_3 - \varepsilon^2 x_4^2 \cos x_3 \sin x_3}{(1-\varepsilon^2 \cos^2 x_3)^2} \varepsilon^2 \cos x_3 \sin x_3$$

$$\begin{aligned}
& \frac{\partial g}{\partial x} f - \frac{\partial f}{\partial x} g = \\
& \left[\begin{array}{c} 0 \\ \varepsilon (\sin x_3) (1 + \varepsilon^2 \cos^2 x_3) \frac{x_4}{1 - 2\varepsilon^2 \cos^2 x_3 + \varepsilon^4 \cos^4 x_3} \\ 0 \\ -2\varepsilon^2 (\cos x_3 \sin x_3) \frac{x_4}{1 - 2\varepsilon^2 \cos^2 x_3 + \varepsilon^4 \cos^4 x_3} \end{array} \right] - \\
& \left[\begin{array}{c} \varepsilon \frac{\cos x_3}{-1 + \varepsilon^2 \cos^2 x_3} \\ 2\varepsilon x_4 \frac{\sin x_3}{1 - 2\varepsilon^2 \cos^2 x_3 + \varepsilon^4 \cos^4 x_3} \\ -\frac{1}{-1 + \varepsilon^2 \cos^2 x_3} \\ -2\varepsilon^2 (\cos x_3 \sin x_3) \frac{x_4}{1 - 2\varepsilon^2 \cos^2 x_3 + \varepsilon^4 \cos^4 x_3} \end{array} \right] = \\
& \left[\begin{array}{c} -\varepsilon \frac{\cos x_3}{-1 + \varepsilon^2 \cos^2 x_3} \\ \varepsilon (\sin x_3) (1 + \varepsilon^2 \cos^2 x_3) \frac{x_4}{1 - 2\varepsilon^2 \cos^2 x_3 + \varepsilon^4 \cos^4 x_3} - 2\varepsilon x_4 \frac{\sin x_3}{1 - 2\varepsilon^2 \cos^2 x_3 + \varepsilon^4 \cos^4 x_3} \\ \frac{1}{-1 + \varepsilon^2 \cos^2 x_3} \\ 0 \end{array} \right] = \\
& \left[\begin{array}{c} -\varepsilon \frac{\cos x_3}{-1 + \varepsilon^2 \cos^2 x_3} \\ \varepsilon x_4 \frac{\sin x_3}{-1 + \varepsilon^2 \cos^2 x_3} \\ \frac{1}{-1 + \varepsilon^2 \cos^2 x_3} \\ 0 \end{array} \right]
\end{aligned}$$

Un conjunto de ecuaciones para las ecuaciones anteriores es:

$$z_3 = x_1 + \varepsilon \sin x_3$$

$$z_4 = x_2 + \varepsilon x_4 \cos x_3$$

Verificando:

$$L_g z_3 = \frac{\partial z_3}{\partial x} g = \begin{bmatrix} 1 & 0 & \varepsilon \cos x_3 & 0 \end{bmatrix} \begin{bmatrix} 0 \\ \frac{-\varepsilon \cos x_3}{1 - \varepsilon^2 \cos^2 x_3} \\ 0 \\ \frac{1}{1 - \varepsilon^2 \cos^2 x_3} \end{bmatrix} = 0$$

$$L_g z_4 = \frac{\partial z_4}{\partial x} g = \begin{bmatrix} 0 & 1 & -\varepsilon x_4 \sin x_3 & \varepsilon \cos x_3 \end{bmatrix} \begin{bmatrix} 0 \\ \frac{-\varepsilon \cos x_3}{1 - \varepsilon^2 \cos^2 x_3} \\ 0 \\ \frac{1}{1 - \varepsilon^2 \cos^2 x_3} \end{bmatrix} = 0$$

$$L_{\text{ad}_f g(x)} z_3 = \frac{\partial z_3}{\partial x} \text{ad}_f g(x) = \begin{bmatrix} 1 & 0 & \varepsilon \cos x_3 & 0 \end{bmatrix} \begin{bmatrix} -\varepsilon \frac{\cos x_3}{-1 + \varepsilon^2 \cos^2 x_3} \\ \varepsilon x_4 \frac{\sin x_3}{-1 + \varepsilon^2 \cos^2 x_3} \\ \frac{1}{-1 + \varepsilon^2 \cos^2 x_3} \\ 0 \end{bmatrix} = 0$$

$$L_{\text{ad}_f g(x)} z_4 = \frac{\partial z_4}{\partial x} \text{ad}_f g(x) = \begin{bmatrix} 0 & 1 & -\varepsilon x_4 \sin x_3 & \varepsilon \cos x_3 \end{bmatrix} \begin{bmatrix} -\varepsilon \frac{\cos x_3}{-1 + \varepsilon^2 \cos^2 x_3} \\ \varepsilon x_4 \frac{\sin x_3}{-1 + \varepsilon^2 \cos^2 x_3} \\ \frac{1}{-1 + \varepsilon^2 \cos^2 x_3} \\ 0 \end{bmatrix} = 0$$

Las nuevas variables de estado en la parte linealizada del sistema transformado pueden ser obtenidas eligiendo $z_1(x)$ tal que:

$$L_g z_1 = 0, \quad L_{\text{ad}_f g(x)} z_1 \neq 0$$

y con los vectores:

$$\left(\frac{\partial z_1}{\partial x}\right)^T, \left(\frac{\partial z_3}{\partial x}\right)^T, \left(\frac{\partial z_4}{\partial x}\right)^T$$

sean linealmente independientes [23]. La solución mas simple a estas condiciones es:

$$z_2 = L_f z_3 = \frac{\partial z_3}{\partial x} f = \begin{bmatrix} 0 & 0 & 1 & 0 \end{bmatrix} \begin{matrix} z_1 = x_3 \\ \left[\begin{array}{c} x_2 \\ \frac{-x_1 + \varepsilon x_4^2 \sin x_3}{1 - \varepsilon^2 \cos^2 x_3} \\ x_4 \\ \frac{\varepsilon x_1 \cos x_3 - \varepsilon^2 x_4^2 \cos x_3 \sin x_3}{1 - \varepsilon^2 \cos^2 x_3} \end{array} \right] \end{matrix} = x_4$$

verificando:

$$\left(\frac{\partial z_1}{\partial x}\right)^T, \left(\frac{\partial z_3}{\partial x}\right)^T, \left(\frac{\partial z_4}{\partial x}\right)^T = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 1 & \varepsilon \cos x_3 & -\varepsilon x_4 \sin x_3 \\ 0 & 0 & \varepsilon \cos x_3 \end{bmatrix}, \text{rank} : 3$$

$$L_g z_1 = \frac{\partial z_1}{\partial x} g = \begin{bmatrix} 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} 0 \\ \frac{-\varepsilon \cos x_3}{1 - \varepsilon^2 \cos^2 x_3} \\ 0 \\ \frac{1}{1 - \varepsilon^2 \cos^2 x_3} \end{bmatrix} = 0$$

$$L_{\text{ad}_f g(x)} z_1 = \frac{\partial z_1}{\partial x} \text{ad}_f g(x) = \begin{bmatrix} 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} -\varepsilon \frac{\cos x_3}{-1 + \varepsilon^2 \cos^2 x_3} \\ \varepsilon x_4 \frac{\sin x_3}{-1 + \varepsilon^2 \cos^2 x_3} \\ \frac{1}{-1 + \varepsilon^2 \cos^2 x_3} \\ 0 \end{bmatrix} = \frac{1}{-1 + \varepsilon^2 \cos^2 x_3} \neq 0$$

Así, las nuevas variables de estado son:

$$x_1 = z_3 - \varepsilon \sin x_3$$

$$x_2 = z_4 - \varepsilon x_4 \cos x_3$$

$$x_3 = z_1$$

$$x_4 = z_2$$

Se calcula el sistema transformado:

$$z_1 = h(x) \implies \dot{z}_1 = \frac{\partial h}{\partial x} \dot{x} = L_f h(x) + L_g h(x)u = x_4$$

$$z_2 = L_f h(x) \implies \dot{z}_2 = \frac{\partial L_f h(x)}{\partial x} \dot{x} = L_f^2 h(x) + L_g L_f h(x)u = f_4 + g_2 u$$

$$z_3 = L_f z_3(x) \implies \dot{z}_3 = \frac{\partial z_3}{\partial x} f =$$

$$= \begin{bmatrix} 1 & 0 & \varepsilon \cos x_3 & 0 \end{bmatrix} \begin{bmatrix} x_2 \\ \frac{-x_1 + \varepsilon x_4^2 \sin x_3}{1 - \varepsilon^2 \cos^2 x_3} \\ x_4 \\ \frac{\varepsilon x_1 \cos x_3 - \varepsilon^2 x_4^2 \cos x_3 \sin x_3}{1 - \varepsilon^2 \cos^2 x_3} \end{bmatrix}$$

$$= x_2 + \varepsilon (\cos x_3) x_4$$

$$z_4 = L_f z_4(x) \implies \dot{z}_4 = \frac{\partial z_4}{\partial x} f =$$

$$= \begin{bmatrix} 0 & 1 & -\varepsilon x_4 \sin x_3 & \varepsilon \cos x_3 \end{bmatrix} \begin{bmatrix} x_2 \\ \frac{-x_1 + \varepsilon x_4^2 \sin x_3}{1 - \varepsilon^2 \cos^2 x_3} \\ x_4 \\ \frac{\varepsilon x_1 \cos x_3 - \varepsilon^2 x_4^2 \cos x_3 \sin x_3}{1 - \varepsilon^2 \cos^2 x_3} \end{bmatrix}$$

$$= f_2 + f_3 (-\varepsilon x_4 \sin x_3) + f_4 (\varepsilon \cos x_3) = -x_1$$

El sistema se reescribe de la siguiente forma:

$$\begin{aligned}
 \dot{z}_1 &= z_2 \\
 \dot{z}_2 &= \frac{\varepsilon \cos z_1}{1 - \varepsilon^2 \cos^2 z_1} (z_3 - \varepsilon(1 + z_4^2) \sin z_1 - F) + \frac{1}{1 - \varepsilon^2 \cos^2 z_1} u \\
 \dot{z}_3 &= z_4 \\
 \dot{z}_4 &= -z_3 + \varepsilon \sin z_1 + F \\
 y &= z_2
 \end{aligned} \tag{3.7}$$

En el resto del trabajo, ver [16], se asume que no existen perturbaciones, i.e. $F = 0$.

Observación 3.1 *El modelo Lagrangiano, Normalizado y el de primer orden son utilizados para la identificación de este sistema usando RND. El modelo en cascada se obtuvo para implementar una acción de control usando este modelo via redes neuronales dinámicas.*

Capítulo 4

Identificación de Sistemas No-lineales

4.1 *Introducción*

La identificación es uno de los puntos esenciales en la teoría de control en el caso cuando no se cuenta con la información completa del sistema. Una solución efectiva es utilizar RNA, pues han mostrado ser una herramienta muy útil para identificar sistemas no-lineales complejos [10], aun cuando la planta es considerada como una “caja negra”. Los neuro-identificadores pueden ser clasificados como estáticos y dinámicos [3]. La mayoría de las publicaciones que presentan la identificación de sistemas no-lineales usan RNE, como por ejemplo el perceptrón multicapa, el cual es implementado para la aproximación de funciones no-lineales que se encuentran en el lado derecho de la ecuación que representa la dinámica del modelo [42]. La principal desventaja de las RNE es que la regla de actualización de los pesos utiliza información local contenida en la estructura de los datos que se le presentan, por otra parte, la función de aproximación es sensible a los datos utilizados para su entrenamiento [13]. Las RND pueden superar exitosamente esta desventaja dado que su estructura incorpora una retroalimentación [19] [36] [48].

En el caso cuando no se cuenta con la medición de todos los estados del sistema, lo

cual puede ser un grave problema cuando la planta tiene una gran cantidad de estados internos. Para resolver este problema existen al menos dos formas:

1. Usar una RNA que contenga un observador tipo Luenberger [17] [49], donde la RNA debe tener la misma dimension de los estados de la planta. Dado que solo se cuenta el error de la salida, los pesos correspondientes a los estados internos no cambian cuando se aplica la regla de aprendizaje. Asi, se requiere añadir otros terminos, como un termino de retraso [49] o un termino lineal [17].
2. Utilizar el “principio de separación”, i.e, un observador de los estados y un sistema de identificación son tratados de forma separada. La dificultad radica en que ninguna otra informacion de la planta puede ser utilizada, por lo que un observador no-lineal libre de la estructura del sistema se requiere, como lo es un observador de alta ganancia [26]. Sin embargo, el observador de alta ganancia requiere que el sistema no-lineal tenga una forma linealizada, esta condición no es aceptable cuando se busca identificar un sistema que se considera como una “caja negro”. Por lo que un observador de modo deslizante [14] es usado.

En este trabajo, se utiliza la segunda forma para resolver este problema, dado que no se busca tratar con términos como los retrasos y el capítulo trata en su mayor parte al problema de identificación.

4.2 Redes Neuronales Dinámicas de una capa para Identificación de Sistemas No-lineales

4.2.1 Estructura de la red neuronal de una capa

El sistema no-lineal a ser identificado es de la forma:

$$\dot{x}_t = f(x_t, u_t, t), \quad x_t \in \mathbb{R}^n, u_t \in \mathbb{R}^m \quad (4.1)$$

Diferentes redes neuronales se proponen en [36, 37, 19, 31], en esta seccion se analizará la siguiente red neuronal de una capa [48]:

$$\dot{\hat{x}}_t = A\hat{x}_t + W_{1,t}\sigma(\hat{x}_t) + W_{2,t}\phi(\hat{x}_t) u_t \quad (4.2)$$

donde $\hat{x}_t \in \mathfrak{R}^n$ son los estados de la red neuronal, $u_t \in \mathfrak{R}^n$ es el control dado, $W_{1,t} \in \mathfrak{R}^{n \times n}$, $W_{2,t} \in \mathfrak{R}^{n \times n}$ son las matrices de pesos, $A \in \mathfrak{R}^{n \times n}$ es una matriz estable. $\sigma(\cdot) \in \mathfrak{R}^n$ es una funcion vectorial sigmoideal, $\phi(\cdot) \in \mathfrak{R}^{n \times m}$ es una matriz diagonal, i.e.

$$\phi(\hat{x}_t) = \text{diag}(\phi_1(\hat{x}_1) \cdots \phi_n(\hat{x}_n))$$

Los elementos $\sigma_i(\cdot)$ (como los de $\phi_i(\cdot)$) son usualmente funciones sigmoideales, i.e.

$$\sigma_i(x_i) = \frac{a_i}{1 + e^{-b_i x_i}} - c_i.$$

Esta red neuronal dinámica es la mas simple, ya que no contiene capas ocultas.

4.2.2 Preliminares

Considerese las siguientes suposiciones:

Suposición 4.1 (A1) : *El control u_t es seleccionado como acotado, entonces :*

$$u_t \leq \bar{u}$$

En el caso general, cuando la RND (4.2) no puede identificar exactamente el sistema no-lineal dado (4.1), el sistema no-lineal (4.1) puede ser representado como:

$$\dot{x}_n = ax_n + W_1^* \sigma(\mathbf{x}) + W_2^* \phi(x) u_t + \tilde{f}_t. \quad (4.3)$$

donde \tilde{f}_t es el error de modelado, W_1^* , W_2^* , V_1^* y V_2^* son matrices constantes. Dado que las funciones sigmoideales σ y ϕ son uniformemente acotadas, es razonable asumir que

las dinámicas no-modeladas \tilde{f}_t , satisfacen la siguiente suposición:

Suposición 4.2 (A2) *Existe una constante positiva $\bar{\eta}$ tal que*

$$\left\| \tilde{f}_t \right\|_{\Delta_f}^2 \leq \bar{\eta}, \quad \Lambda_f = \Lambda_f^T > 0$$

Definición 4.1 *Sea el error de identificación:*

$$\Delta_t := \mathbf{x} - \hat{\mathbf{x}}$$

Suposición 4.3 (A3) *Es claro que las funciones sigmoideas, comúnmente usadas en las redes neuronales satisfacen la condición de Lipschitz, basado en el Lema 1 (ver Apéndice) se concluye que:*

$$\tilde{\sigma}_t^T \Lambda_1 \tilde{\sigma}_t \leq \Delta_t^T \Lambda_\sigma \Delta_t, \quad \left(\tilde{\phi}_t u_t \right)^T \Lambda_2 \left(\tilde{\phi}_t u_t \right) \leq \bar{u}^2 \Delta_t^T \Lambda_\phi \Delta_t$$

donde

$$\begin{aligned} \tilde{\sigma}_t &:= \sigma(x_t) - \sigma(\hat{x}_t), & \tilde{\phi}_t &:= \phi(x_t) - \phi(\hat{x}_t) \\ \tilde{W}_{1,t} &:= W_1^* - W_{1,t}, & \tilde{W}_{2,t} &:= W_2^* - W_{2,t} \end{aligned}$$

Λ_σ y Λ_ϕ son matrices definidas positivas.

Se conoce [44] que si la matriz A es estable, el par $(A, R^{1/2})$ es controlable, el par $(Q^{1/2}, A)$ es observable, y la *condición especial local de frecuencia* o su matriz equivalente:

$$A^T R^{-1} A - Q \geq \frac{1}{4} [A^T R^{-1} - R^{-1} A] R [A^T R^{-1} - R^{-1} A]^T \geq 0 \quad (4.4)$$

se cumple, entonces la matriz de Riccati

$$A^T P + PA + PRP + Q = 0 \quad (4.5)$$

tiene una solución positiva. Siguiendo este hecho, se introduce adicionalmente la siguiente suposición

Suposición 4.4 (A4) Para una matriz estable A dada, existe una matriz estrictamente positiva Q_1 tal que la ecuación de la matriz de Riccati (4.5) con:

$$R = 2\bar{W}_1 + 2\bar{W}_2 + \Lambda_f^{-1}, \quad \bar{W}_1 := W_1^{*T} \Lambda_1^{-1} W_1^*, \quad \bar{W}_2 := W_2^{*T} \Lambda_2^{-1} W_2^*$$

$$Q = Q_1 + \Lambda_\sigma + \bar{u}^2 \Lambda_\phi,$$

tiene una solución positiva. Estas condiciones se cumplen fácilmente si se selecciona A como una matriz diagonal estable.

4.2.3 Regla de Actualización de Pesos

La regla dinámica de actualización de los pesos dada por:

$$\begin{aligned} \dot{W}_{1,t} &= -s_t [k_1 P \Delta_t \sigma(\hat{x}_t)^T] \\ \dot{W}_{2,t} &= -s_t [k_2 P \Delta_t u_t \phi(\hat{x}_t)^T] \end{aligned} \quad (4.6)$$

donde $k_i \in \mathfrak{R}^1$ ($i = 1, 2$), P es la solución de la ecuación matricial de Riccati dada por (4.5). Las condiciones iniciales son $W_{1,0} = W_1^*$, $W_{2,0} = W_2^*$. s_t es la función de zona muerta definida como:

$$s_t := \left[1 - \frac{\mu}{\|P^{1/2} \Delta\|} \right]_+, \quad [z]_+ = \begin{cases} z & z \geq 0 \\ 0 & z < 0 \end{cases} \quad (4.7)$$

$$\mu = \frac{\bar{\eta}}{\lambda_{\min} \left(P^{-\frac{1}{2}} Q_1 P^{-\frac{1}{2}} \right)} \quad (4.8)$$

4.2.4 Análisis de Estabilidad

Teorema 4.1 Se considera el sistema no-lineal (4.1) y la red neuronal de una capa (4.2). Si se cumplen las suposiciones A1-A4 y los pesos de esta red se ajustan de acuerdo a (4.6). Los siguientes hechos se cumplen:

$$\Delta_t, W_{1,t}, W_{2,t} \in L^\infty \quad (4.9)$$

el error de identificación Δ_t satisface el siguiente desempeño de seguimiento:

$$\limsup_{T \rightarrow \infty} \frac{1}{T} \int_0^T \Delta_t^T Q_1 \Delta_t s_t dt \leq \bar{\eta} \quad (4.10)$$

Prueba. De (4.2) y (4.3) la ecuación del error puede ser expresada como:

$$\dot{\Delta}_t = A\Delta_t + \widetilde{W}_{1,t}\sigma + \widetilde{W}_{2,t}\phi u_t + W_1^* \tilde{\sigma}_t + W_2^* \tilde{\phi}_t u_t + \tilde{f}_t. \quad (4.11)$$

donde $\sigma, \phi, \widetilde{W}_{1,t}, \widetilde{W}_{2,t}$ se definen en A3. Sea la función candidata de Lyapunov:

$$V_t := \left[\left\| P^{\frac{1}{2}} \Delta_t \right\| - \mu \right]_+^2 + \text{tr} \left[\widetilde{W}_{1,t}^T K_1^{-1} \widetilde{W}_{1,t} \right] + \text{tr} \left[\widetilde{W}_{2,t}^T K_2^{-1} \widetilde{W}_{2,t} \right] \quad (4.12)$$

donde $P = P^T > 0$, entonces con base en el Lema 2 (ver Apéndice), se calcula la derivada de V_t :

$$\begin{aligned} \dot{V}_t &\leq 2 \left[\left\| P^{\frac{1}{2}} \Delta_t \right\| - \mu \right]_+ \frac{(P^{\frac{1}{2}} \Delta_t)^T}{\left\| P^{\frac{1}{2}} \Delta_t \right\|} P^{\frac{1}{2}} \dot{\Delta}_t + 2 \text{tr} \left[\dot{\widetilde{W}}_{1,t}^T K_1^{-1} \widetilde{W}_{1,t} \right] + 2 \text{tr} \left[\dot{\widetilde{W}}_{2,t}^T K_2^{-1} \widetilde{W}_{2,t} \right] \\ &= 2 \left[1 - \mu \left\| P^{\frac{1}{2}} \Delta_t \right\|^{-1} \right]_+ \Delta_t^T P \dot{\Delta}_t + 2 \text{tr} \left[\dot{\widetilde{W}}_{1,t}^T K_1^{-1} \widetilde{W}_{1,t} \right] + 2 \text{tr} \left[\dot{\widetilde{W}}_{2,t}^T K_2^{-1} \widetilde{W}_{2,t} \right] \end{aligned} \quad (4.13)$$

Si se define s_t como en (4.7), entonces (4.13) queda como:

$$\dot{V}_t \leq s_t 2 \Delta_t^T P \dot{\Delta}_t + 2 \text{tr} \left[\dot{\widetilde{W}}_{1,t}^T K_1^{-1} \widetilde{W}_{1,t} \right] + 2 \text{tr} \left[\dot{\widetilde{W}}_{2,t}^T K_2^{-1} \widetilde{W}_{2,t} \right]$$

De (4.11) se tiene:

$$\begin{aligned} 2 \Delta_t^T P \dot{\Delta}_t &= 2 \Delta_t^T P A \Delta_t + \\ &2 \Delta_t^T P \left(\widetilde{W}_{1,t} \sigma + \widetilde{W}_{2,t} \phi u_t + W_1^* \tilde{\sigma}_t + W_2^* \tilde{\phi}_t u_t + \tilde{f}_t \right) \end{aligned} \quad (4.14)$$

Usando la desigualdad matricial:

$$X^T Y + (X^T Y)^T \leq X^T \Lambda^{-1} X + Y^T \Lambda Y \quad (4.15)$$

la cual es valida para cualquier $X, Y \in \mathfrak{R}^{n \times k}$ y para cualquier matriz definida positiva $0 < \Lambda = \Lambda^T \in \mathfrak{R}^{n \times n}$, de **A2** el error de modelado $2\Delta_t^T P \tilde{f}_t$ puede ser estimado como:

$$2\Delta_t^T P \tilde{f}_t \leq \Delta_t^T P \Lambda_f^{-1} P \Delta_t + \tilde{f}_t^T \Lambda_f \tilde{f}_t \leq \Delta_t^T P \Lambda_f^{-1} P \Delta_t + \bar{\eta} \quad (4.16)$$

Usando **A3** los términos $\Delta_t^T P \left(W_1^* \tilde{\sigma}_t + W_2^* \tilde{\phi}_t u_t \right)$ en (4.14) pueden ser reescritos como:

$$\begin{aligned} 2\Delta_t^T P W_1^* \tilde{\sigma}_t &\leq \Delta_t^T P W_1^* \Lambda_1^{-1} W_1^{*T} P \Delta_t + \tilde{\sigma}_t^T \Lambda_1 \tilde{\sigma}_t \leq \Delta_t^T (P \bar{W}_1 P + \Lambda_\sigma) \Delta_t \\ 2\Delta_t^T P W_2^* \tilde{\phi}_t u_t &\leq \Delta_t^T (P \bar{W}_2 P + \bar{u}^2 \Lambda_\phi) \Delta_t. \end{aligned} \quad (4.17)$$

Dado que $s_t \geq 0$ y usando (4.14), (4.17), (4.16). La ecuación (4.13) se reescribe como

$$\dot{V}_t \leq s_t \Delta_t^T L \Delta_t + L_{w1} + L_{w2} - \Delta_t^T Q_1 \Delta_t s_t + \bar{\eta} s_t \quad (4.18)$$

donde

$$\begin{aligned} L &= A^T P + P A + P R P + Q, \\ L_{w1} &= 2tr \left[\begin{array}{c} \cdot^T \\ \tilde{W}_{1,t} \end{array} k_1^{-1} \tilde{W}_{1,t} \right] + 2\Delta_t^T P \tilde{W}_{1,t} \sigma s_t \\ L_{w2} &= 2tr \left[\begin{array}{c} \cdot^T \\ \tilde{W}_{2,t} \end{array} k_2^{-1} \tilde{W}_{2,t} \right] + 2\Delta_t^T P \tilde{W}_{2,t} u_t s_t \end{aligned}$$

Usando **A4** y la regla de actualizacion (4.6) y dado que

$$\dot{\tilde{W}}_{1,t} = -\dot{W}_{1,t}, \quad \dot{\tilde{W}}_{2,t} = -\dot{W}_{2,t}$$

se obtiene:

$$L_{w1} = 0, \quad L_{w2} = 0$$

Finalmente, de (4.18) se obtiene que:

$$\dot{V}_t \leq -s_t [\Delta_t^T Q_0 \Delta_t - \bar{\eta}] \quad (4.19)$$

El lado derecho de esta desigualdad puede ser estimada de la siguiente forma:

$$\dot{V}_t \leq -s_t \lambda_{\min} \left(P^{\frac{1}{2}} Q_1 P^{-\frac{1}{2}} \right) \left(\left\| P^{\frac{1}{2}} \Delta_t \right\|^2 - \mu \right) \leq 0 \quad (4.20)$$

donde μ es definido como en (4.8). Así, V_t es acotada, por lo que (4.9) es probada.

Desde que $0 \leq s_t \leq 1$, de (4.20) se tiene:

$$\dot{V}_t \leq -\Delta_t^T Q_1 \Delta_t s_t + \bar{\eta} s_t \leq -\Delta_t^T Q_0 \Delta_t s_t + \bar{\eta}$$

Integrando (4.19) de 0 hasta T se obtiene:

$$V_T - V_0 \leq - \int_0^T \Delta^T Q_0 \Delta s_t dt + \bar{\eta} T$$

así,

$$\int_0^T \Delta^T Q_1 \Delta s_t dt \leq V_0 - V_T + \bar{\eta} T \leq V_0 + \bar{\eta} T$$

Dado que $W_{1,0} = W_1^*$ y $W_{2,0} = W_2^*$, V_0 están acotadas, (4.10) es probada. ■

Observación 4.1 (C1) *La estructura (4.2) de la RND de una capa es mas sencilla y al ser utilizada para identificar un sistema, para un sistema no-lineal dado (caso particular) esta RND puede llegar a obtener mejores resultados que otras estructuras mas complejas (RND multicapa), entonces puede ser utilizada para obtener la acción de control propuesta en el diseño de un control indirecto.*

4.2.5 Simulaciones

Problema 4.1 *Identificar los 4 estados del sistema no lineal TORA (3.7), usando una red neuronal dinámica de una capa (4.2).*

Solución 4.1 Se usa la red neuronal dinámica de una capa (4.3), junto con la regla de actualización de pesos (4.6) y $W_{1,t} \in R^{4 \times 4}$, $W_{2,t} \in R^{4 \times 4}$, $\sigma(x) = \frac{2}{(1+e^{-2x})} - 0.5$. Las condiciones iniciales para las matrices de los pesos son:

$$W_{1,0} = W_{2,0} = \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 \end{bmatrix}$$

con $\bar{\eta} = 0.2$, $k_{1,2} := 1006$, para la identificación de sistema. La simulaciones se llevaron acabo usando el paquete matlab y el simulink, el diagrama general se muestra en la figura B-1 . Los resultados obtenidos de identificación de los 4 estados se muestran en las figuras 4-1, 4-2, 4-3, 4-4.

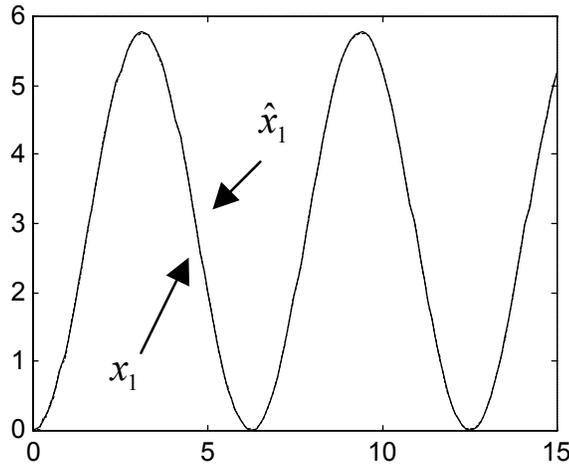


Figura 4-1: Identificación usando una RND con una capa del estado q .

Observación 4.2 Como se puede ver en las figuras anteriores, en el caso particular del sistema no lineal TORA, el resultado de identificación obtenido por la RND de una capa, es muy satisfactorio, lo cual, puede ser muy útil en el diseño de un control indirecto.

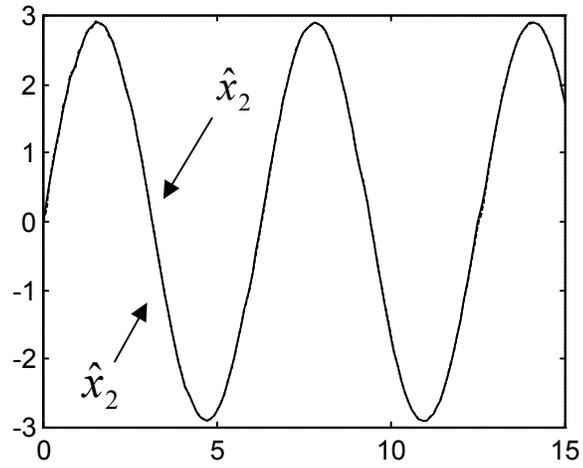


Figura 4-2: Identificación usando una RND con una capa del estado \dot{q} .

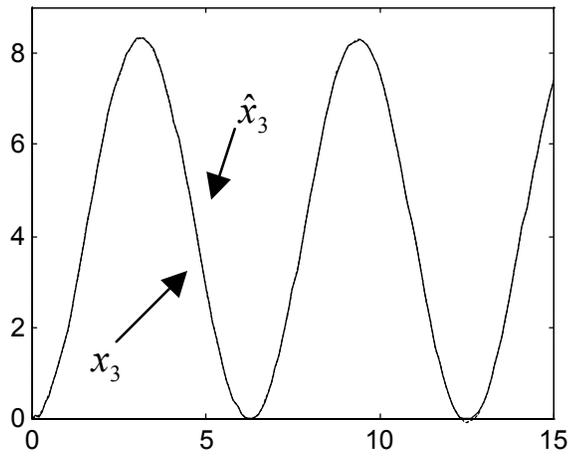


Figura 4-3: Identificación usando una RND con una capa del estado θ .

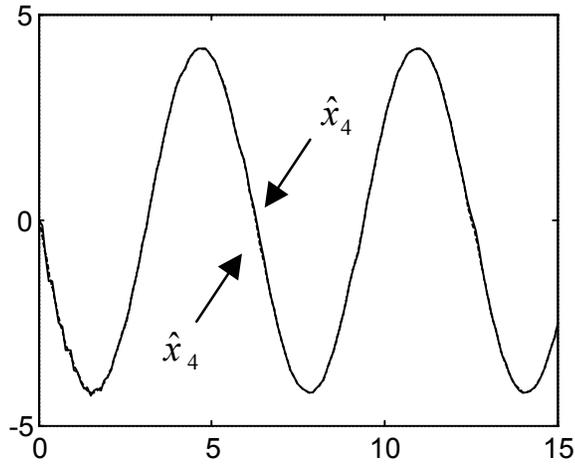


Figura 4-4: Identificación usando una RND con una capa del estado $\dot{\theta}$.

Observación 4.3 *El rango de ajuste de las ganancias $k_{1,2}$ están entre $[500, 1300]$ por lo que la sintonización fue fácil hasta obtener el valor final de 1006.*

4.3 *Estimación de los estados internos de un sistema no-lineal*

4.3.1 *Introducción*

Existen algunas publicaciones usan la suposición de tener completa accesibilidad a los estados del sistema [15]. Pero en realidad esto no es siempre valido. En el caso cuando sólo la entrada y la salida es medible, una red neuronal compleja puede relacionar los pares entrada-salida. Si el sistema no-lineal tiene muchos estados internos, el mapeo entrada-salida no es suficiente para modelar totalmente el sistema no-lineal. Por lo que se puede utilizar un observador para estimar los estados internos, entonces poder utilizar todos los estados para identificar el sistema no-lineal.

Muchos de los observadores no-lineales están basados en la estructura de la planta. En

el caso cuando sólo se cuenta con los pares entrada/salida, este es un problema bastante difícil que no ha recibido mucha atención, en comparación a los trabajos que se han enfocado en la modificación de las redes neuronales. Por lo cual en esta sección, se propone la combinación de un observador no-lineal libre de la estructura del sistema [18] con una RND para atacar este problema, obteniendo un buen resultado en la identificación del sistema no-lineal del cual no se tiene completo acceso a sus estados.

4.3.2 Observador de modo deslizante

En general (3.4) puede ser escrito como.

$$\begin{aligned}\dot{x}_t &= f(x_t, u_t) \\ y_t &= Cx_t\end{aligned}\tag{4.21}$$

donde $x_t \in \mathfrak{R}^n$ es el estado de la planta, $u_t \in \mathfrak{R}$ es el control de entrada, $y_t \in \mathfrak{R}$ es la salida medible. La clase de sistemas no-lineales descritos de esta forma pueden tratarse con la siguiente suposición. Transformando el sistema (4.21) en la forma normal:

$$\begin{aligned}\dot{x}_t &= Ax_t + F(x_t, u_t) \\ y_t &= Cx_t\end{aligned}\tag{4.22}$$

donde $F(x_t, u_t) := f(x_t, u_t) - Ax_t$, A es una matrix especial tal que el par (A, C) es observable. Ahora, se construye un observador independiente del modelo del sistema: Un observador de modo deslizante como:

$$\begin{aligned}\dot{\bar{x}}_t &= A\bar{x}_t + S(\bar{x}_t, e) - Ke_t \\ \bar{y}_t &= C\bar{x}_t\end{aligned}\tag{4.23}$$

Definición 4.2 Sea el error de salida:

$$e_t = \bar{y}_t - y_t = C\Delta_t = C(\bar{x}_t - x_t)\tag{4.24}$$

donde Δ_t es el error de observación

$S(\bar{x}_t, e_t)$ es seleccionado como:

$$S(\bar{x}_t, e_t) = -\frac{P^{-1}C^T C \Delta_t}{\|C \Delta_t\|} \rho, \quad (4.25)$$

Así, la derivada del error de observación es:

$$\begin{aligned} \dot{\Delta}_t &= A \Delta_t + S(\bar{x}_t, e_t) - K C \Delta_t - F(x_t, u_t) \\ &= (A - K C) \Delta_t + S(\bar{x}_t, e_t) - F(x_t, u_t) \\ &= A_0 \Delta_t + S(\bar{x}_t, e_t) - F(x_t, u_t) \end{aligned}$$

donde $A_0 := A - K C$. Dado que (A, C) es observable, esto significa que existe K tal que A_0 es estable. Así, existe $Q > 0$ tal que la siguiente ecuación algebraica de Lyapunov tiene una solución $P > 0$.

$$A_0^T P + P A_0 = -Q, \quad Q = Q^T > 0$$

Suposición 4.5 (A1) Para la función no-lineal $f(x_t, u_t)$, se asume que satisface:

$$f(x_t, u_t) - A x_t = -P^{-1} C^T h(x_t, u_t)$$

donde $h(x_t, u_t)$ está acotada por: $\|h(x_t, u_t)\| < \rho$, $\rho > 0$.

4.3.3 Diagrama de diseño

Algoritmo 4.1 El proceso de diseño de la dinámica de un observador de este tipo se muestra en la Fig.4-5

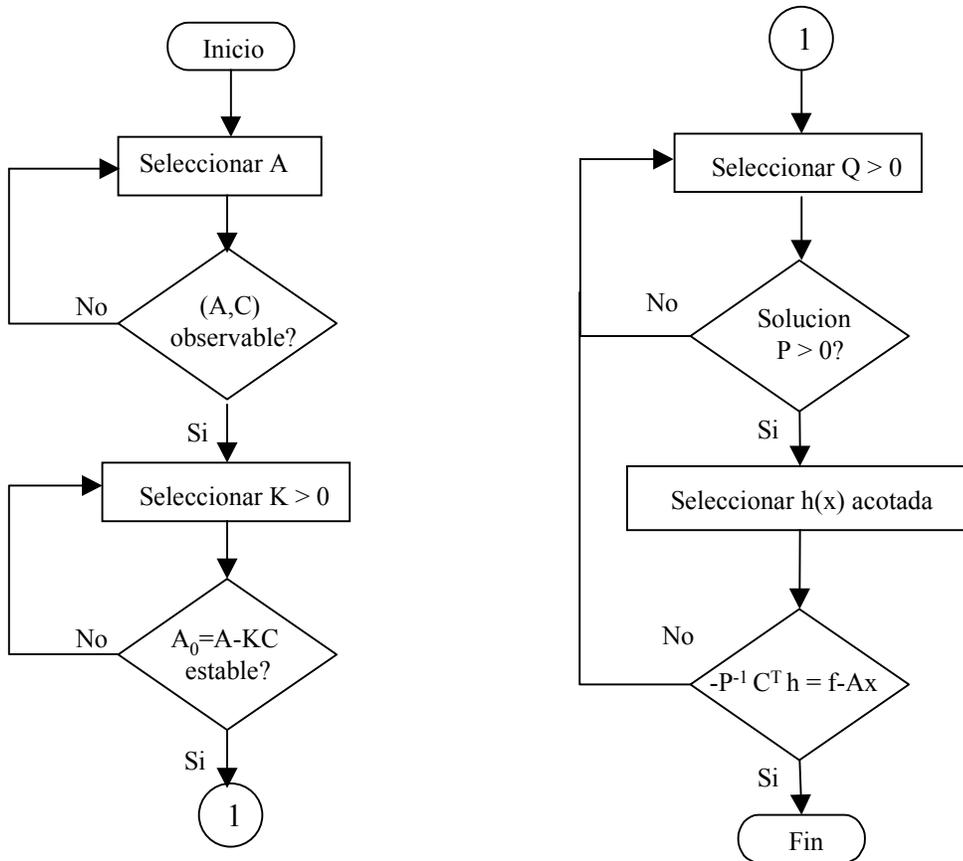


Figura 4-5: Diagrama de diseño del observador.

4.3.4 Análisis de Estabilidad

Teorema 4.2 *Bajo la suposición A1, el error de observación entre el observador de modo deslizante (4.23) y el sistema no-lineal (4.21) es asintóticamente estable.*

$$\lim_{t \rightarrow \infty} \Delta_t = 0 \quad (4.26)$$

Prueba. Se considera la siguiente función candidata de Lyapunov :

$$V_t = \Delta_t^T P \Delta_t$$

Calculando su derivada

$$\dot{V}_t = \Delta_t^T (A_0^T P + P A_0) \Delta_t + 2 \Delta_t^T P [S(\bar{x}_t, e_t) - F(x_t, u_t)].$$

Usando **A1**

$$F(x_t, u_t) = -P^{-1} C^T h(x_t, u_t), \quad \|h(x_t, u_t)\| < \rho.$$

y aplicando (4.25), se tiene

$$\begin{aligned} \dot{V}_t &= -\Delta_t^T Q \Delta_t + 2 \Delta_t^T C^T h(x_t, u_t) - 2 \frac{\Delta_t^T C^T C \Delta_t}{\|C \Delta_t\|} \rho \\ &= -\Delta_t^T Q \Delta_t + 2 \Delta_t^T C^T h(x_t, u_t) - 2 \|C \Delta_t\| \rho \\ &\leq -\Delta_t^T Q \Delta_t + 2 \|C \Delta_t\| (\|h(x_t, u_t)\| - \rho) < 0 \end{aligned}$$

dado que $\dot{V}_t < 0$ se obtiene que el error de observación Δ es asintóticamente estable, así

$$\lim_{t \rightarrow \infty} \Delta_t = 0. \quad \blacksquare$$

Observación 4.4 *El observador de alta ganancia (4.23) no depende de la planta no-lineal (4.21) o de (3.4), sólo se requiere la salida y .*

Observación 4.5 *Para eliminar el castaño que se presenta en las dinámicas no mode-*

ladas de alta frecuencia. El siguiente compensador acotado es usado

$$S(\bar{x}_t, e_t) = -\frac{P^{-1}C^T C \Delta_t}{\|C \Delta_t\|} \rho = \begin{cases} -\rho P^{-1}C^T \text{sign}(C \Delta_t) & |C \Delta_t| \geq \delta \\ -\rho P^{-1}C^T C \Delta_t / \delta & |C \Delta_t| < \delta \end{cases} \quad (4.27)$$

Este ofrece una aproximación continua de las discontinuidades que se presentan al usar el observador de modo deslizante, así el compensador acotado garantiza que el error de observación permaneciera dentro de una vecindad del origen [9].

4.3.5 Simulaciones

Ejemplo 4.1 Se aplicará el algoritmo de diseño al sistema TORA

1.- Seleccionar A , y checar que el par (A, C) sea observable .

$$A = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}, \quad C = \begin{bmatrix} 1 & 0 \end{bmatrix}$$

Para verificar que el par (A, C) sea observable, se debe calcular la siguiente matriz, la cual debe tener rango completo:

$$\begin{aligned} \text{rango} \begin{bmatrix} C \\ CA \end{bmatrix} &= 2 \\ C &= \begin{bmatrix} 1 & 0 \end{bmatrix}, \quad CA = \begin{bmatrix} 1 & 0 \end{bmatrix} \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix} = \begin{bmatrix} 0 & 1 \end{bmatrix} \\ \text{rango} \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} &= 2 \end{aligned}$$

Por lo tanto, el par (A, C) es observable por lo tanto existe K tal que $A_0 = A - KC$ es estable

2.- Seleccionar $K > 0$ para que la matriz A_0 sea estable. Si tomamos $K = \begin{bmatrix} 2 \\ 2 \end{bmatrix}$

entonces.

$$A_0 = A - KC$$

$$A_0 = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix} - \left(\begin{bmatrix} 2 \\ 2 \end{bmatrix} \begin{bmatrix} 1 & 0 \end{bmatrix} \right)$$

$$A_0 = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix} - \begin{bmatrix} 2 & 0 \\ 2 & 0 \end{bmatrix}$$

$$A_0 = \begin{bmatrix} -2 & 1 \\ -1 & 0 \end{bmatrix}$$

Se verifica que la matriz A_0 resultante sea estable, obteniendo su polinomio característico y sus raíces, las cuales deben ser reales negativas:

$$\det |sI - A_0| = \det \left| \begin{bmatrix} s & 0 \\ 0 & s \end{bmatrix} - \begin{bmatrix} -2 & 1 \\ -1 & 0 \end{bmatrix} \right| = \det \left| \begin{bmatrix} s+2 & -1 \\ 1 & s \end{bmatrix} \right|$$

polinomio característico : $s^2 + 2s + 1$

raíces : $s_1 = -1, \quad s_2 = -1$

como las raíces tienen parte real negativa la matriz A_0 es estable.

3.- Seleccionar $Q > 0$ y obtener la solución P de la ecuación algebraica de Lyapunov

$$Q = \begin{bmatrix} 3 & 1 \\ 8 & 3 \end{bmatrix}, \quad A_0 = \begin{bmatrix} -2 & 1 \\ -1 & 0 \end{bmatrix}$$

$$A_0 P + P A_0^T = -Q$$

la solución es:

$$P = \begin{bmatrix} 1.5 & -0.25 \\ 3.25 & 0 \end{bmatrix}$$

y es única, entonces se verifica que sea definida positiva, utilizando la condición de Sylvester:

$$P = \begin{bmatrix} 1.5 & -0.25 \\ 3.25 & 0 \end{bmatrix} = \begin{bmatrix} P_1 & P_2 \\ P_3 & P_4 \end{bmatrix}$$

realizando el cálculo de los menores

$$P_1 = 1.5 > 0$$

$$\det |P| = P_1 P_4 - P_2 P_3 = 0 + 0.8125 = 0.8125 > 0$$

asi la matriz P es definida positiva..

4.- Obtener una funcion $h(x)$ acotada a partir de las matrices P y A :

$$-P^{-1}C^T h = f - Ax$$

Podemos reescribir el sistema de la forma:

$$\begin{aligned} \dot{x}_1 &= x_2 \\ \dot{x}_2 &= \frac{-x_1 + \varepsilon x_4^2 \sin x_3}{1 - \varepsilon^2 \cos^2 x_3} = f_1 \end{aligned} \tag{4.28}$$

x_3 y x_4 se consideran perturbaciones

$$\begin{aligned} \dot{x}_3 &= x_4 \\ \dot{x}_4 &= \frac{\varepsilon \cos x_3 (x_1 - \varepsilon x_4^2 \sin x_3)}{1 - \varepsilon^2 \cos^2 x_3} = f_2 \end{aligned} \tag{4.29}$$

x_1 y x_2 se consideran perturbaciones

5.- Se realiza el calculo de h para el sistema (4.28):

$$\begin{aligned}
 & -P^{-1}C^T h = f - Ax \\
 - \begin{bmatrix} 0 & 0.3077 \\ -4 & 1.8462 \end{bmatrix} \begin{bmatrix} 1 \\ 0 \end{bmatrix} h &= \begin{bmatrix} x_2 \\ f_1 \end{bmatrix} - \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} \\
 \begin{bmatrix} 0 \\ 4 \end{bmatrix} h &= \begin{bmatrix} x_2 \\ f_1 \end{bmatrix} - \begin{bmatrix} x_2 \\ x_1 \end{bmatrix} \\
 \begin{bmatrix} 0 \\ 4 h \end{bmatrix} &= \begin{bmatrix} 0 \\ f_1 - x_1 \end{bmatrix} \\
 h &= \frac{1}{4} [f_1 - x_1]
 \end{aligned}$$

Substituyendo f_1 :

$$\begin{aligned}
 h &= \frac{1}{4} \left[\frac{-x_1 + \varepsilon x_4^2 \sin x_3}{1 - \varepsilon^2 \cos^2 x_3} - x_1 \right] \\
 h &= -\frac{1}{4} \frac{-2x_1 + \varepsilon x_4^2 \sin x_3 + x_1 \varepsilon^2 \cos^2 x_3}{-1 + \varepsilon^2 \cos^2 x_3}
 \end{aligned}$$

Se realiza el calculo de h para el sistema (4.29):

$$\begin{aligned}
 & -P^{-1}C^T h = f - Ax \\
 - \begin{bmatrix} 0 & 0.3077 \\ -4 & 1.8462 \end{bmatrix} \begin{bmatrix} 1 \\ 0 \end{bmatrix} h &= \begin{bmatrix} x_4 \\ f_2 \end{bmatrix} - \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix} \begin{bmatrix} x_3 \\ x_4 \end{bmatrix} \\
 \begin{bmatrix} 0 \\ 4 \end{bmatrix} h &= \begin{bmatrix} x_4 \\ f_2 \end{bmatrix} - \begin{bmatrix} x_4 \\ x_3 \end{bmatrix} \\
 \begin{bmatrix} 0 \\ 4 h \end{bmatrix} &= \begin{bmatrix} 0 \\ f_2 - x_3 \end{bmatrix} \\
 h &= \frac{1}{4} [f_2 - x_3]
 \end{aligned}$$

Substituyendo f_2 :

$$h = \frac{1}{4} [f_2 - x_3]$$

$$h = \frac{1}{4} \left[\frac{\varepsilon \cos x_3 (x_1 - \varepsilon x_4^2 \sin x_3)}{1 - \varepsilon^2 \cos^2 x_3} - x_3 \right]$$

$$h = -\frac{1}{4} \frac{\varepsilon \cos (x_3 x_1 - x_3 \varepsilon x_4^2 \sin x_3) - x_3 + x_3 \varepsilon^2 \cos^2 x_3}{-1 + \varepsilon^2 \cos^2 x_3}$$

Una vez obtenido las funciones h , para el sistema TORA se plantea el siguiente problema:

Problema 4.2 Estimar los estados x_2 , x_4 que corresponden a los estados que describen las velocidad del carro \dot{q} y la velocidad angular $\dot{\theta}$ respectivamente del sistema no lineal TORA (3.2), usando un observador de modo deslizante.

Solución 4.2 Se usa un observador de modo deslizante (4.23), seleccionado $\rho = 10$, para realizar la estimación. Las simulaciones se llevaron a cabo usando el paquete matlab y el simulink, el diagrama general se muestra en la figura B-2. Los resultados de observación obtenidos de los 2 estados se muestran en las figuras 4-6, 4-7.

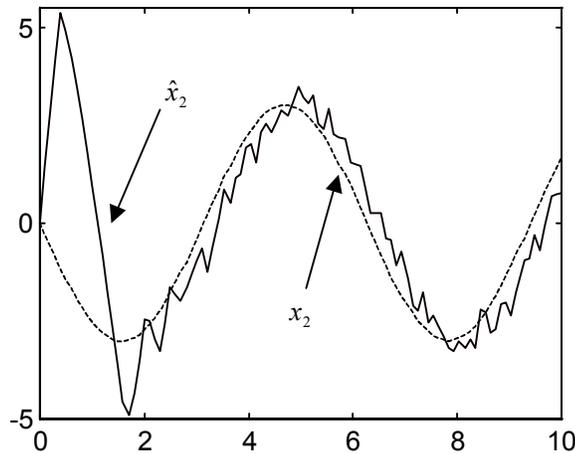


Figura 4-6: Resultados de observación para el modo deslizante del estado \dot{q} .

Observación 4.6 Como se puede ver en las figuras anteriores, el castaño provocado por la función signo, da como resultado una estimación muy pobre, por lo que se sustituye por

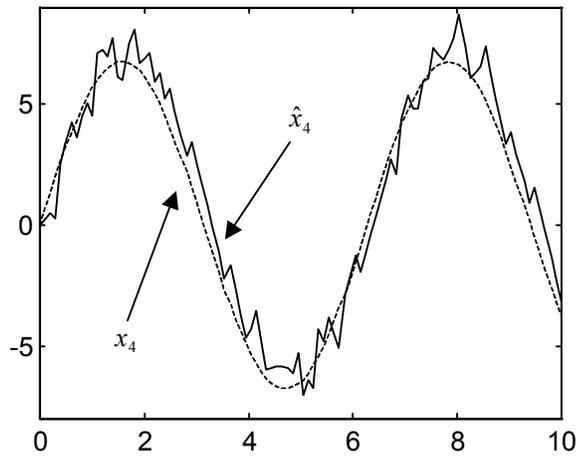


Figura 4-7: Resultados de observación para el modo deslizante del estado \dot{q} .

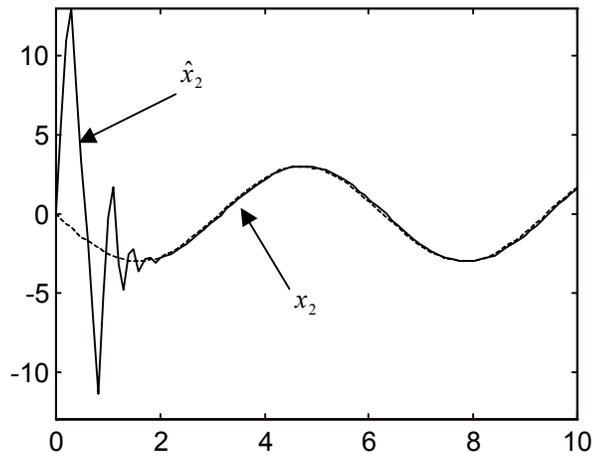


Figura 4-8: Resultados de observación usando el compensador del estado \dot{q} .

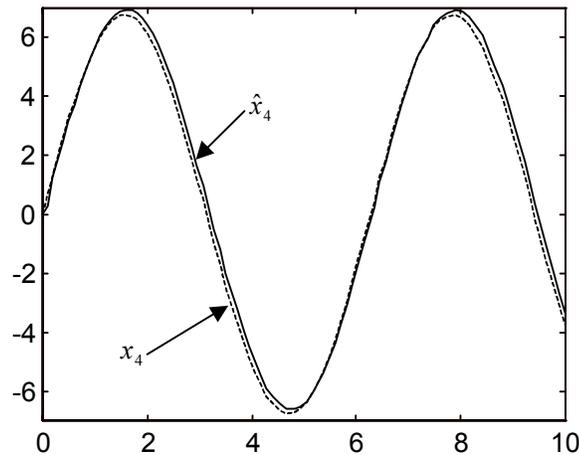


Figura 4-9: Resultados de observación usando el compesador del estado $\dot{\theta}$.

el compensador acotado (4.27). Los resultados de observación obtenidos de los 2 estados usando este compensador se muestran en las figuras , 4-8, 4-9.

Observación 4.7 El ajuste de la mayoría de las ganancias esta entre $[-10, -1]$, siendo -50 el valor mas grande utilizado.

4.4 *Redes Neuronales Dinámicas Multicapa para Identificación de Sistemas No-lineales con medición parcial de sus estados*

4.4.1 *Planteamiento del Problema*

De una entrada u y una salida y es posible identificar al sistema no-lineal (4.21) o (3.4) via redes neuronales dinámicas, sin embargo, este modelo neuronal puede sólo reflejar la relacion entrada/salida. En la sección anterior se muestró la estimación asintótica de los estados internos, si estos estados \bar{x}_i en (4.23) son usados para identificar el sistema,

el modelo de la RND podra aproximar completamente el sistema no-lineal.

Problema 4.3 *La meta de identificación es hacer que los estados del modelo neuronal (4.33) sigan a los de la planta real (4.21) o (3.4).*

Definición 4.3 *Sea la siguiente semi-norma:*

$$\|z\|_R^2 = \limsup_{\tau \rightarrow \infty} \frac{1}{\tau} \int_0^\tau z^T(t) R z(t) dt, \quad (4.30)$$

donde $R = R^T > 0$

Para resolver este problema, puede ser formulado con el siguiente criterio:

$$J_{\min} = \min_W J, \quad J = \|x - \hat{x}\|_R^2 \quad (4.31)$$

Asi, para cualquier $\eta > 0$, se tiene

$$J \leq (1 + \eta) \|x - \bar{x}\|_R^2 + (1 + \eta^{-1}) \|\bar{x} - \hat{x}\|_R^2. \quad (4.32)$$

el mínimo del término $\|x - \bar{x}\|_R^2$ ya ha sido resuelto en la sección anterior. Seleccionando $\tilde{R} = (1 + \eta^{-1}) R$, ahora podemos reformular nuestro objetivo de identificación como: minimizar el término $\|\bar{x} - \hat{x}\|_R^2$. Entonces

Definición 4.4 *Sea el error de identificación:*

$$\Delta := \bar{x} - \hat{x}.$$

4.4.2 Estructura de la red neuronal dinámica multicapa

Considere la siguiente RND, para identificar el sistema no-lineal:

$$\hat{x}_t = A\hat{x}_t + W_{1,t}\sigma(V_{1,t}\hat{x}_t) + u_t \quad (4.33)$$

donde $\forall t \in [0, \infty)$, el vector $\hat{x}_t \in \mathfrak{R}^n$ son los estados de la RND, $A \in \mathfrak{R}^{n \times n}$ es la matriz estable la cual puede ser seleccionada despues. Las matrices $W_{1,t} \in \mathfrak{R}^{n \times m}$ y $V_1 \in \mathfrak{R}^{m \times n}$ son los pesos que describen las conexiones entre las capas ocultas y la de salida, $u_t \in R^n$ es el campo vectorial de control dado. $\sigma(\cdot) \in R^m$ son las funciones vectoriales sigmoidales. Los elementos $\sigma_i(\cdot)$ son usualmente funciones sigmoidales, i.e.,

$$\sigma_i(x_i) = \frac{a_i}{1 + e^{-b_i x_i}} - c_i.$$

La estructura de este sistema dinámico se muestra en la figura (4-10).

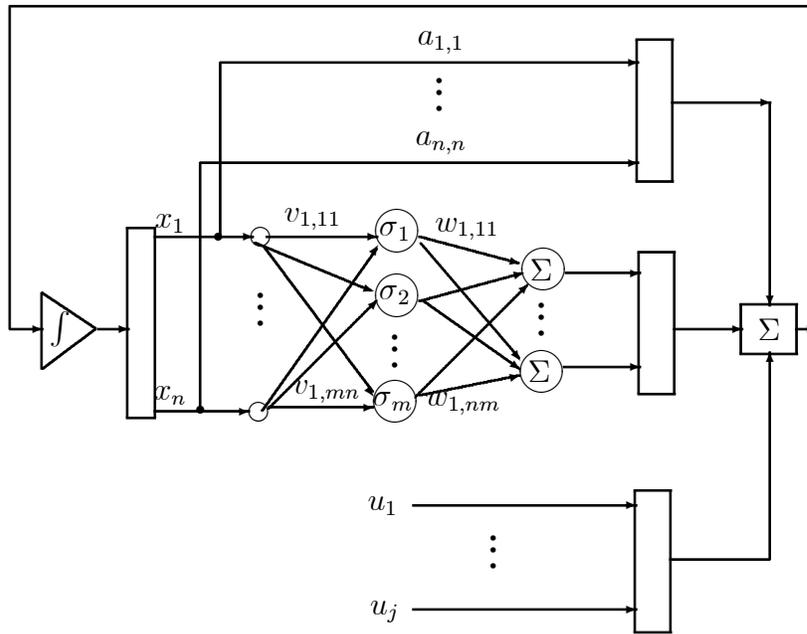


Figura 4-10: Estructura general de una red neuronal dinámica.

4.4.3 Preliminares

Considerese las siguientes suposiciones:

Suposición 4.6 (A1) *El control u_t es seleccionado como acotado, entonces:*

$$u_t \leq \bar{u}$$

En el caso general, cuando la RND (4.33) no puede identificar exactamente el sistema no-lineal dado (4.21), el sistema no-lineal (4.21) puede ser representado como:

$$\dot{\bar{x}}_t = A\bar{x}_t + W_1^* \sigma(V_1^* \bar{x}_t) + u_t + \tilde{f}_t. \quad (4.34)$$

donde \tilde{f}_t es el error de modelado, W_1^* y V_1^* son matrices constantes. Dado que las funciones sigmoideas σ y ϕ son uniformemente acotadas. Es razonable asumir que las dinámicas no-modeladas \tilde{f}_t , satisfacen la siguiente suposición:

Suposición 4.7 (A2) *Existe una constante positiva $\bar{\eta}$ tal que*

$$\left\| \tilde{f}_t \right\|_{\Lambda_f}^2 \leq \bar{\eta}, \quad \Lambda_f = \Lambda_f^T > 0$$

Definición 4.5 *Sea el error de identificación:*

$$\Delta_t := x_t - \hat{x}_t$$

Suposición 4.8 (A3) *Es claro que las funciones sigmoideas, comúnmente usadas en las redes neuronales satisfacen la condición de Lipschitz, basado en el Lema 1 (ver Apéndice) se concluye que*

$$\tilde{\sigma}_t^T \Lambda_1 \tilde{\sigma}_t \leq \Delta_t^T \Lambda_\sigma \Delta_t, \quad \tilde{\sigma}_t' = D_\sigma \tilde{V}_{1,t} \hat{x}_t + \nu_\sigma$$

donde:

$$\begin{aligned} \tilde{\sigma}_t &:= \sigma(V_1^* x_t) - \sigma(V_1^* \hat{x}_t), & \tilde{\sigma}_t' &:= \sigma(V_1^* \hat{x}_t) - \sigma(V_{1,t} \hat{x}_t), \\ D_\sigma &= \frac{\partial \sigma^T(Z)}{\partial Z} \Big|_{Z=V_{1,t} \hat{x}_t} \in R^{m \times m}, & \|\nu_\sigma\|_{\Lambda_1}^2 &\leq l_1 \left\| \tilde{V}_{1,t} \hat{x}_t \right\|_{\Lambda_1}^2, \quad l_1 > 0 \\ \tilde{V}_{1,t} &:= V_1^* - V_{1,t}, & \tilde{W}_{1,t} &:= W_1^* - W_{1,t}, \end{aligned}$$

Λ_1 y Λ_σ son matrices definidas positivas.

Se conoce [44] que si la matriz A es estable, el par $(A, R^{1/2})$ es controlable, el par $(Q^{1/2}, A)$ es observable, y la *condición especial local de frecuencia* o su matriz equivalente:

$$A^T R^{-1} A - Q \geq \frac{1}{4} [A^T R^{-1} - R^{-1} A] R [A^T R^{-1} - R^{-1} A]^T \geq 0 \quad (4.35)$$

se cumple, entonces que la matriz de Riccati

$$A^T P + PA + PRP + Q = 0 \quad (4.36)$$

tiene una solución positiva. Siguiendo este hecho, se introduce adicionalmente la siguiente suposición:

Suposición 4.9 (A4) *Para una matriz estable A dada, existe una matriz estrictamente positiva Q_1 tal que la ecuación de la matriz de Riccati (4.36) con*

$$\begin{aligned} R &= 2\overline{W}_1 + \Lambda_f^{-1}, \quad \overline{W}_1 := W_1^{*T} \Lambda_1^{-1} W_1^*, \\ Q &= Q_1 + \Lambda_\sigma, \end{aligned} \quad (4.37)$$

tiene una solución positiva. Estas condiciones se cumplen fácilmente si se selecciona A como una matriz diagonal estable.

4.4.4 Regla de Actualización de Pesos

La regla dinámica de actualización de los pesos dada por

$$\begin{aligned} \dot{W}_{1,t} &= -s_t \left[K_1 P \Delta_t \sigma^T - K_1 P \Delta_t \hat{x}_t^T \tilde{V}_{1,t}^T D_\sigma \right] \\ \dot{V}_{1,t} &= -s_t \left[K_2 P W_{1,t} D_\sigma \Delta_t \hat{x}_t^T - \frac{l_1}{2} K_2 \Lambda_1 \tilde{V}_{1,t} \hat{x}_t \hat{x}_t^T \right] \end{aligned} \quad (4.38)$$

donde $K_i \in R^{n \times n}$ ($i = 1, 2$) son matrices positivas definidas, P es la solución de la ecuación matricial de Riccati dada por (4.36). Las condiciones iniciales son $W_{1,0} = W_1^*$,

$V_{1,0} = V_1^* \cdot s_t$ es la función de zona muerta definida como:

$$s_t := \left[1 - \frac{\mu_1}{\|P^{1/2}\Delta\|} \right]_+, \quad [z]_+ = \begin{cases} z & z \geq 0 \\ 0 & z < 0 \end{cases} \quad (4.39)$$

$$\mu = \frac{\bar{\eta}}{\lambda_{\min} \left(P^{-\frac{1}{2}} Q_1 P^{-\frac{1}{2}} \right)} \quad (4.40)$$

4.4.5 *Análisis de Estabilidad*

Teorema 4.3 *Se considera el sistema no-lineal (4.1) y la red neuronal de una capa (4.33). Si se cumplen las suposiciones A1-A4 y los pesos de esta red se ajustan de acuerdo a (4.38). Los siguientes hechos se cumplen:*

$$\Delta_t, W_{1,t}, V_{1,t} \in L^\infty \quad (4.41)$$

el error de identificación Δ_t satisface el siguiente desempeño de seguimiento:

$$\limsup_{T \rightarrow \infty} \frac{1}{T} \int_0^T \Delta_t^T Q_1 \Delta_t s_t dt \leq \bar{\eta} \quad (4.42)$$

Prueba. De (4.33) y (4.34) la ecuación del error puede ser expresada como:

$$\dot{\Delta}_t = A\Delta_t + \widetilde{W}_{1,t}\sigma + W_1^* \tilde{\sigma}_t + W_1^* \tilde{\sigma}'_t + \tilde{f}_t. \quad (4.43)$$

Sea la función candidata de Lyapunov:

$$V_t := \left[\left\| P^{\frac{1}{2}} \Delta_t \right\| - \mu \right]_+^2 + \text{tr} \left[\widetilde{W}_{1,t}^T K_1^{-1} \widetilde{W}_{1,t} \right] + \text{tr} \left[\widetilde{V}_{1,t} K_3^{-1} \widetilde{V}_{1,t}^T \right]$$

donde $P = P^T > 0$, entonces con base en el Lema 2 (ver Apéndice), se calcula la derivada de V_t :

$$\begin{aligned}\dot{V}_t &\leq 2 \left[\left\| P^{\frac{1}{2}} \Delta_t \right\| - \mu \right]_+ \frac{(P^{\frac{1}{2}} \Delta_t)^T}{\left\| P^{\frac{1}{2}} \Delta_t \right\|} P^{\frac{1}{2}} \dot{\Delta}_t + 2tr \left[\begin{matrix} \dot{\widetilde{W}}_{1,t}^T \\ \widetilde{W}_{1,t} K_1^{-1} \widetilde{W}_{1,t} \end{matrix} \right] + 2tr \left[\begin{matrix} \dot{\widetilde{V}}_{1,t} \\ \widetilde{V}_{1,t} K_3^{-1} \widetilde{V}_{1,t}^T \end{matrix} \right] \\ &= 2 \left[1 - \mu \left\| P^{\frac{1}{2}} \Delta_t \right\|^{-1} \right]_+ \Delta P^{\frac{1}{2}} \dot{\Delta}_t + 2tr \left[\begin{matrix} \dot{\widetilde{W}}_{1,t}^T \\ \widetilde{W}_{1,t} K_1^{-1} \widetilde{W}_{1,t} \end{matrix} \right] + 2tr \left[\begin{matrix} \dot{\widetilde{V}}_{1,t} \\ \widetilde{V}_{1,t} K_3^{-1} \widetilde{V}_{1,t}^T \end{matrix} \right]\end{aligned}\quad (4.44)$$

Si se define a s_t como en (4.39), entonces (4.44) se tiene:

$$\dot{V}_t \leq s_t 2\Delta^T P \dot{\Delta} + 2tr \left[\begin{matrix} \dot{\widetilde{W}}_{1,t}^T \\ \widetilde{W}_{1,t} K_1^{-1} \widetilde{W}_{1,t} \end{matrix} \right] + 2tr \left[\begin{matrix} \dot{\widetilde{V}}_{1,t} \\ \widetilde{V}_{1,t} K_3^{-1} \widetilde{V}_{1,t}^T \end{matrix} \right]$$

De (4.43) se tiene:

$$2\Delta_t^T P \dot{\Delta}_t = 2\Delta_t^T P A \Delta_t + 2\Delta_t^T P W_1^* \tilde{\sigma}_t + 2\Delta_t^T P \widetilde{W}_{1,t} \sigma + 2\Delta_t^T P W_1^* \tilde{\sigma}'_t \quad (4.45)$$

Usando la desigualdad matricial:

$$X^T Y + (X^T Y)^T \leq X^T \Lambda^{-1} X + Y^T \Lambda Y \quad (4.46)$$

la cual es valida para cualquier $X, Y \in \mathfrak{R}^{n \times k}$ y para cualquier matriz definida positiva $0 < \Lambda = \Lambda^T \in \mathfrak{R}^{n \times n}$, y usando **A3** $\Delta_t^T P W_1^* \tilde{\sigma}_t$ en (4.45) puede ser reescrita como

$$2\Delta_t^T P W_1^* \tilde{\sigma}_t \leq \Delta_t^T P W_1^* \Lambda_1^{-1} W_1^{*T} P \Delta_t + \tilde{\sigma}_t^T \Lambda_1 \tilde{\sigma}_t \leq \Delta_t^T (P \overline{W}_1 P + \Lambda_\sigma) \Delta_t \quad (4.47)$$

Usando **A3**, el último término en (4.45) puede ser reescrita como:

$$2\Delta_t^T P W_1^* \tilde{\sigma}'_t = 2\Delta_t^T P W_{1,t} D_\sigma \widetilde{V}_{1,t} \widehat{x}_t + 2\Delta_t^T P \widetilde{W}_{1,t} D_\sigma \widetilde{V}_{1,t} \widehat{x}_t + 2\Delta_t^T P W_1^* \nu_\sigma \quad (4.48)$$

El término $2\Delta_t^T PW_1^* \nu_\sigma$ en (4.48) puede ser reescrito como:

$$2\Delta_t^T PW_1^* \nu_\sigma \leq \Delta_t^T PW_1^{*T} \Lambda_1^{-1} W_1^* P \Delta_t + \nu_\sigma^T \Lambda_1 \nu_\sigma \leq \Delta_t^T P \bar{W}_1 P \Delta_t + l_1 \left\| \tilde{V}_{1,t} \hat{x}_t \right\|_{\Lambda_1}^2 \quad (4.49)$$

De **A2** $2\Delta_t^T P \tilde{f}_t$ puede ser estimado como:

$$2\Delta_t^T P \tilde{f}_t \leq \Delta_t^T P \Lambda_f^{-1} P \Delta_t + \tilde{f}_t^T \Lambda_f \tilde{f}_t \leq \Delta_t^T P \Lambda_f^{-1} P \Delta_t + s_t \bar{\eta} \quad (4.50)$$

Usando (4.45), (4.47), y (4.50) se reescribe como:

$$\dot{V}_t \leq s_t \Delta_t^T L \Delta_t + L_{w1} + L_{v1} - s_t \Delta_t^T Q_1 \Delta_t + s_t \bar{\eta} \quad (4.51)$$

donde:

$$\begin{aligned} L &= A^T P + PA + PRP + Q, \\ L_{w1} &= 2tr \left[\begin{array}{c} \dot{\tilde{W}}_{1,t}^T \\ \tilde{W}_{1,t} \end{array} K_1^{-1} \tilde{W}_{1,t} \right] + 2s_t \Delta_t^T P \tilde{W}_{1,t} \sigma + 2s_t \Delta_t^T P \tilde{W}_{1,t} D_\sigma \tilde{V}_{1,t} \hat{x}_t, \\ L_{v1} &= 2tr \left[\begin{array}{c} \dot{\tilde{V}}_{1,t} \\ \tilde{V}_{1,t} \end{array} K_3^{-1} \tilde{V}_{1,t}^T \right] + 2s_t \Delta_t^T P W_{1,t} D_\sigma \tilde{V}_{1,t} \hat{x}_t + s_t l_1 \left\| \tilde{V}_{1,t} \hat{x}_t \right\|_{\Lambda_1}^2 \end{aligned}$$

Usando la regla de actualización (4.38), y dado que

$$\dot{\tilde{W}}_{1,t} = -\dot{W}_{1,t}, \quad \dot{\tilde{V}}_{1,t} = -\dot{V}_{1,t}$$

se obtiene

$$L_{w1} = 0, \quad L_{v1} = 0$$

Finalmente, de (4.51) se obtiene que:

$$\dot{V}_t \leq -s_t [\Delta_t^T Q_0 \Delta_t - \bar{\eta}] \quad (4.52)$$

El lado derecho de esta desigualdad puede ser estimada de la siguiente forma:

$$\dot{V}_t \leq -s_t \lambda_{\min} \left(P^{\frac{1}{2}} Q_1 P^{-\frac{1}{2}} \right) \left(\left\| P^{\frac{1}{2}} \Delta_t \right\|^2 - \mu \right) \leq 0 \quad (4.53)$$

donde μ es definido como en (4.40). Así, V_t es acotada, por lo que (4.41) es probada.

Desde que $0 \leq s_t \leq 1$, de (4.53) se tiene:

$$\dot{V}_t \leq -\Delta_t^T Q_1 \Delta_t s_t + \bar{\eta} s_t \leq -\Delta_t^T Q_0 \Delta_t s_t + \bar{\eta}$$

Integrando (4.52) de 0 hasta T se obtiene:

$$V_T - V_0 \leq -\int_0^T \Delta^T Q_0 \Delta s_t dt + \bar{\eta} T$$

así,

$$\int_0^T \Delta^T Q_1 \Delta s_t dt \leq V_0 - V_T + \bar{\eta} T \leq V_0 + \bar{\eta} T$$

Dado que $W_{1,0} = W_1^*$, $V_{1,0} = V_1^*$, V_0 están acotadas, (4.42) es probada. ■

Observación 4.8 *Se puede ver que la regla de actualización de pesos (4.38) de red neuronal dinámica (RND) (4.33) tiene una estructura parecida al algoritmo de propagación hacia atrás (backpropagation) de perceptrón multicapa (MLP) (ver [13]). Si consideramos $K_i P$ como la tasa de actualización, el primer término $K_1 P \Delta_t \sigma^T$ corresponde exactamente al esquema propuesto por dicho algoritmo (backpropagation). Los segundos términos son nuevos elementos y son utilizados para asegurar que el aprendizaje sea estable.*

Observación 4.9 *Dado que la regla de aprendizaje propuesta es parecida al algoritmo de propagación hacia atrás (backpropagation) con un término adicional, la estabilidad global asintótica del error es garantizada puesto que es derivada en base al enfoque de Lyapunov. Así el problema de mínimos locales (el cual es un punto importante en las redes neuronales estáticas) no se presenta en este caso.*

4.4.6 Simulaciones

Problema 4.4 Identificar los 4 estados del sistema no lineal TORA (3.7), usando el resultado obtenido de la estimación parcial de sus estados y una red neuronal dinámica multicapa (4.33).

Solución 4.3 Se usa la red neuronal dinámica multicapa (4.34), junto con la regla de actualización de pesos (4.38) y $W_{1,t} \in R^{4 \times 3}$, $V_{1,t} \in R^{3 \times 4}$, $\sigma(x) = \frac{2}{(1+e^{-2x})} - 0.5$. Las condiciones iniciales para las matrices de los pesos son:

$$W_{1,0}^T = V_{10} = \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & 2 & 1 & 2 \\ 2 & 1 & 2 & 1 \end{bmatrix}$$

con $\bar{\eta} = 0.2$, $K_{1,2} := 350I_4$, para realizar la identificación. La simulaciones se llevaron acabo usando el paquete matlab y el simulink, el diagrama general se muestra en la figura B-3. Los resultados obtenidos de identificación de los 4 estados se muestran en las figuras 4-11, 4-12, 4-13, 4-14.

Observación 4.10 Como se puede ver en las figuras anteriores, el resultado de identificación obtenido por la RND multicapa usando 2 estados estimados mediante un observador, es satisfactorio. Es importante mencionar que la RND multicapa usada para obtener estos resultados corresponde al caso cuando $W_2^* \phi(V_2^* x_t) := 1$, el caso general cuando $W_2^* \phi(V_2^* x_t) \neq 1$, se muestra en el capítulo 6.

Observación 4.11 El rango de ajuste de las ganancias $K_{1,2}$ estan entre [500, 1300] por lo que la sintonización fue fácil hasta obtener le valor final de 300.

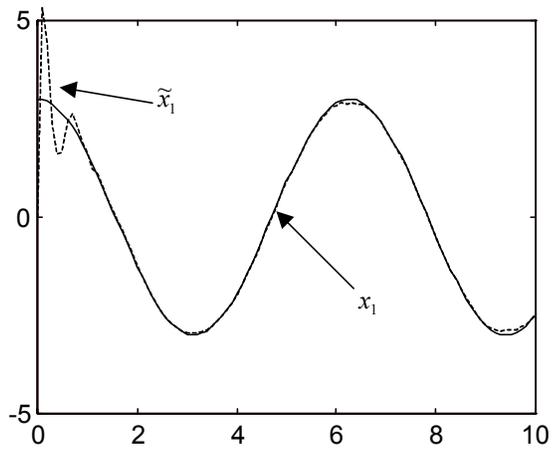


Figura 4-11: Identificación con estimación del estado q .

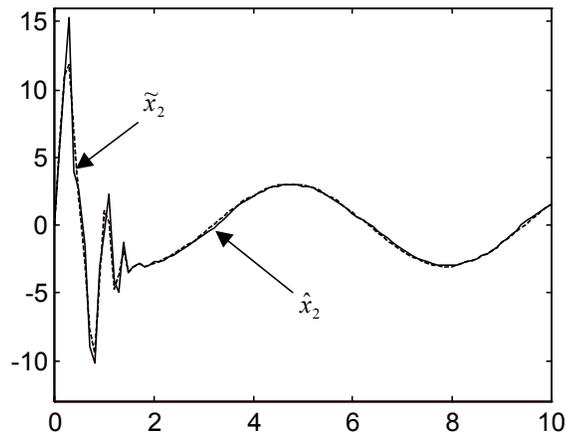


Figura 4-12: Identificación con estimación del estado \dot{q} .

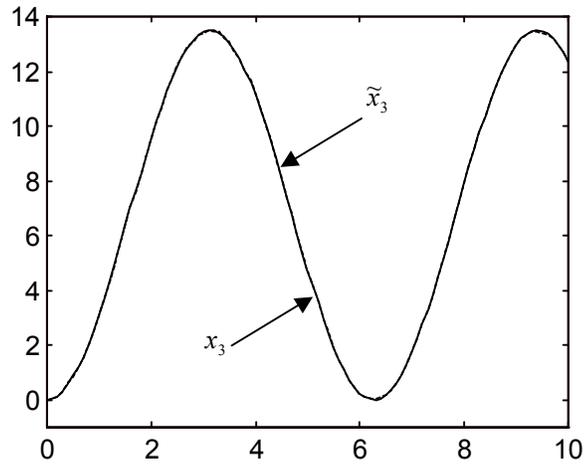


Figura 4-13: Identificación con estimación del estado θ .

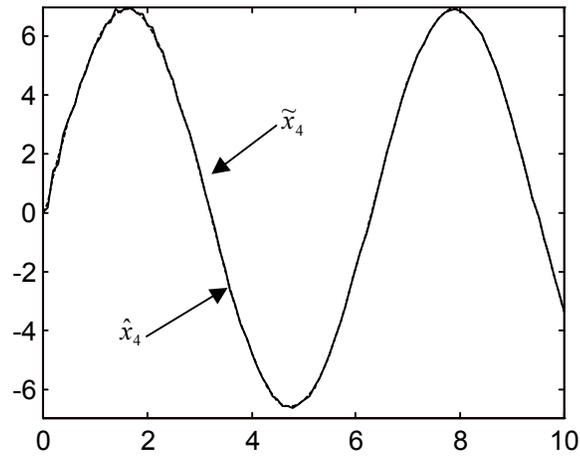


Figura 4-14: Identificación con estimación del estado $\dot{\theta}$.

Capítulo 5

Control Adaptable usando Redes Neuronales Dinámicas

5.1 *Introducción*

Las redes neuronales son una herramienta muy efectiva para controlar sistemas no-lineales complejos, cuando no se cuenta con información completa del modelo ó cuando se considera a la planta a ser controlada como una “caja negra”. Un panorama general del control neuronal puede ser revisado en [15].

Existen dos tipos de control neuronal: El control directo [19] y el control indirecto [37]. El control neuronal directo, es aquel donde la RNA se coloca en cascada con el sistema a ser controlado. El objetivo es que el sistema compuesto por la RNA y el sistema no-lineal den como resultado un mapeo identidad entre la entrada y la salida del sistema.

En el control neuronal indirecto, el controlador se basa en la identificación del sistema no-lineal, por lo que el proceso de identificación se convierte en la parte central de la construcción del neuro-controlador. En esta dirección existen 2 tipos de estructuras: El modelo serie-paralelo y el modelo paralelo [3]. El modelo serie-paralelo puede asegurar que todas las señales estan acotadas, si la planta es BIBO estable como se muestra en [3] para las redes neuronales multicapa de propagación hacia adelante y en [36] [19] para

las RND. Muchos trabajos publicados utilizan este modelo, aprovechando este resultado de estabilidad.

Por otra parte, el modelo paralelo es muy útil cuando se trabaja con sistemas con ruido, dado que este modelo elimina los problemas del bias causados, por el ruido generado por la salida del sistema real [47], además dado que la identificación de un modelo usualmente se realiza fuera de línea, el modelo paralelo es más adecuado. Sin embargo, este modelo carece de verificación teórica, por lo cual, se requiere cubrir esta carencia para poder disfrutar de sus ventajas.

No existen muchos análisis de estabilidad en el campo del control neuronal, por lo que existen pocos resultados publicados que se refieran al análisis de sistemas no-lineales controlados por RND. En [19], [20] se presentan resultados de una versión particular de las redes neuronales de alto orden. En [31], [32] una cota al error de identificación es dado usando un análisis tipo Lyapunov. Es importante mencionar que la metodología del análisis de estabilidad presentada a continuación es muy similar a la presentada en [46].

5.2 *Control Adaptable Directo*

5.2.1 *Planteamiento del Problema*

La clase de sistemas no-lineales:

$$\begin{aligned}
 \dot{x}_1 &= x_2 \\
 \dot{x}_2 &= x_3 \\
 &\vdots \\
 \dot{x}_n &= f(\mathbf{x}) + g(\mathbf{x})u \\
 y &= x_1
 \end{aligned} \tag{5.1}$$

con $\mathbf{x} = [x_1, x_2, \dots, x_n]^T$, $f, g : \Re^n \rightarrow \Re$ funciones suaves desconocidas con $f(\mathbf{0}) = 0$

Problema 5.1 Diseñar un neuro-controlador u que resuelva el problema de seguimiento para el sistema (5.1), usando una red neuronal dinámica

5.2.2 Estructura de la red neuronal dinámica

La RND tiene la siguiente estructura:

$$\begin{aligned}
 \dot{\hat{x}}_1 &= \hat{x}_2 \\
 \dot{\hat{x}}_2 &= \hat{x}_3 \\
 &\vdots \\
 \dot{\hat{x}}_n &= a\hat{x}_n + W_{1,t}\sigma(V_{1,t}\hat{\mathbf{x}}) + W_{2,t}\phi(V_{2,t}\hat{\mathbf{x}})u \\
 \hat{y} &= \hat{x}_1
 \end{aligned} \tag{5.2}$$

donde $\forall t \in [0, \infty)$, el vector $\hat{x}_t \in \mathfrak{R}^n$ son los estados de la RND, $a \in \mathfrak{R}$, con $a < 0$ y sera especificado después. Las matrices $W_i \in \mathfrak{R}^{1 \times m}$, $V_i \in \mathfrak{R}^{m \times n}$, $i = 1, 2$ son los pesos de la red que describen las conexiones de las capas ocultas y de salida, $u_t \in \mathfrak{R}^1$ es el control dado. $\sigma(\cdot) \in \mathfrak{R}^m$ es una función vectorial sigmoideal, $\phi(\cdot)$ es $\mathfrak{R}^{m \times m}$ una matriz diagonal, i.e.,

$$\phi(\cdot) = \text{diag}[\phi_1(V_{2,t}\hat{\mathbf{x}})_1 \cdots \phi_m(V_{2,t}\hat{\mathbf{x}})_m].$$

Los elementos de $\sigma_i(\cdot)$ (como los de $\phi_i(\cdot)$) son usualmente funciones sigmoideales, i.e.,

$$\sigma_i(x_i) = \frac{a_i}{1 + e^{-b_i x_i}} - c_i.$$

5.2.3 Diseño del Neuro-controlador

La linearización por retroalimentación es muy útil para resolver el problema de seguimiento:

Problema 5.2 Dada una salida deseada $y_d(t)$, encontrar una acción de control u tal que la planta siga una trayectoria deseada con una precisión aceptable (i.e. error de seguimiento acotado), mientras que todos los estados del sistema y el control permanezcan

acotados.

Para resolver este problema, se requieren establecer las siguientes suposiciones las cuales son comunmente usadas. Primero defínase el vector:

$$\mathbf{x}_d(t) = \begin{bmatrix} y_d & \dot{y}_d & \dots & y_d^{(n-1)} \end{bmatrix}^T$$

Suposición 5.1 (A1) *El vector de la trayectoria deseada $\mathbf{x}_d(t)$ es continuo y disponible para su medición, y $\|\mathbf{x}_d(t)\| \leq D$ con D una cota conocida.*

Definición 5.1 *Sea el error de seguimiento:*

$$\mathbf{e} = \mathbf{x} - \mathbf{x}_d$$

y el filtro del error

$$r = \begin{bmatrix} \Lambda^T & 1 \end{bmatrix} \mathbf{e} \in \mathfrak{R}^1$$

donde $\Lambda = [\lambda_1, \lambda_2, \dots, \lambda_{n-1}]^T$ y los coeficientes del vector son apropiadamente seleccionados, tal que $e \rightarrow 0$ exponencialmente como $r \rightarrow 0$ ($s^{n-1} + \lambda_{n-1}s^{n-2} + \dots + \lambda_1$ es Hurwitz). Así la derivada en tiempo del filtro del error se reescribe como:

$$\dot{r} = f(\mathbf{x}) + g(\mathbf{x})u + Y_d \tag{5.3}$$

donde $Y_d = -x_d^{(n)} + \begin{bmatrix} \Lambda^T & 0 \end{bmatrix} \mathbf{e}$.

Ahora, dado que las funciones f, g no son conocidas, se elige la acción de control $u_c \in R^1$ como:

$$u_c = \frac{1}{\hat{g}(\mathbf{x})} \left[-\hat{f}(\mathbf{x}) + v \right] \tag{5.4}$$

donde

$$v = -K_v r - Y_d. \tag{5.5}$$

Asi

$$\begin{aligned}
\dot{r} &= f + gu + Y_d + v - v \\
&= f + gu_c + Y_d + v - v \\
&= f + \left[(\tilde{g} + \hat{g}) \left(\frac{1}{\hat{g}} \left[-\hat{f} + v \right] \right) \right] + Y_d - Y_d - K_v r - v \\
&= -K_v r + \left(f - \hat{f} \right) + (g - \hat{g}) u_c
\end{aligned} \tag{5.6}$$

5.2.4 Preliminares

Suposición 5.2 (A2) *El control u_t es seleccionado como acotado, entonces:*

$$u_t \leq \bar{u}$$

La sub-estructura de (5.2) (sin el integrador), con entrada \mathbf{x} y salida h :

$$h(\mathbf{x}) = W\sigma(V\mathbf{x})$$

Sea $h(\mathbf{x})$ una función continua.

Suposición 5.3 (A3) *Dado el tamaño de la red neuronal, existe W^* y V^* tal que*

$$h(\mathbf{x}) = W^*\sigma(V^*\mathbf{x}) + \varepsilon_m(x) \tag{5.7}$$

con $\varepsilon_m(x)$ el error mínimo de reconstrucción de la red neuronal en u . Note que $\varepsilon_m(x)$ es mas pequeño cuando el número de neuronas m en la capa oculta se incrementa.

Suposición 5.4 (A4) *Dado $\sigma(V^*\mathbf{x})$, $\mathbf{x}_d(t)$, y un número lo suficientemente grande de neuronas ocultas m . Sea el error de reconstrucción de la red neuronal acotada de acuerdo con*

$$\|\varepsilon_m(\mathbf{x})\| = \sup_{\mathbf{x} \in U} |h - W^*\sigma(V^*\mathbf{x})| \leq \bar{\varepsilon} \quad \forall \mathbf{x} \in U \tag{5.8}$$

con $\bar{\varepsilon}$ una cota conocida.

Suposición 5.5 (A5) Dado que σ es una función sigmoideal, satisface la condición de Lipschitz, basado el Lema 1 (ver Appendix) se concluye que

$$\begin{aligned}\tilde{\sigma}_t^T \Lambda_1 \tilde{\sigma}_t &\leq \Delta_t^T \Lambda_\sigma \Delta_t, & \left(\tilde{\phi}_t u_t\right)^T \Lambda_2 \left(\tilde{\phi}_t u_t\right) &\leq \bar{u}^2 \Delta_t^T \Lambda_\phi \Delta_t \\ \tilde{\sigma}'_t &= D_\sigma \tilde{V}_{1,t} \hat{x}_t + \nu_\sigma, & \tilde{\phi}'_t u_t &:= D_\phi \tilde{V}_{2,t} \hat{x}_t + \nu_\phi\end{aligned}$$

donde

$$\begin{aligned}\left[\sigma(V_1^* \mathbf{x}) - \sigma(\hat{V}_1 \mathbf{x})\right] &= D_\sigma \tilde{V}_1 \mathbf{x} + \nu_\sigma \\ \left[\phi(V_2^* \mathbf{x}) u_c - \phi(\hat{V}_2 \mathbf{x}) u_c\right] &= D_\phi \tilde{V}_2 u_c \mathbf{x} + \nu_\phi \\ D_\sigma &= \frac{\partial \sigma^T(Z)}{\partial Z} \Big|_{Z=\hat{V}_1 \mathbf{x}} \in \mathfrak{R}^{m \times m}, & \|\nu_\sigma\|_{\Lambda_1}^2 &\leq l_1 \left\|\tilde{V}_1 \mathbf{x}\right\|_{\Lambda_1}^2, & l_1 > 0 \\ D_\phi &= \frac{\partial(\phi u_c)^T(Z)}{\partial Z} \Big|_{Z=\hat{V}_2 \mathbf{x}} \in R^{m \times m}, & \|\nu_\phi\|_{\Lambda_2}^2 &\leq l_2 \left\|\tilde{V}_2 \mathbf{x}\right\|_{\Lambda_2}^2, & l_2 > 0 \\ \tilde{V}_{1,t} &:= V_1^* - V_{1,t}, & \tilde{V}_{2,t} &:= V_2^* - V_{2,t} \\ \tilde{W}_{1,t} &:= W_1^* - W_{1,t}, & \tilde{W}_{2,t} &:= W_2^* - W_{2,t}\end{aligned} \tag{5.9}$$

Λ_1, Λ_2 son matrices definidas positivas

Se conoce [44] que si la matriz A es estable, el par $(A, R^{1/2})$ es controlable, el par $(Q^{1/2}, A)$ es observable, y la *condición especial local de frecuencia* o su matriz equivalente:

$$A^T R^{-1} A - Q \geq \frac{1}{4} [A^T R^{-1} - R^{-1} A] R [A^T R^{-1} - R^{-1} A]^T \geq 0 \tag{5.10}$$

se cumple, entonces la matriz de Riccati

$$A^T P + PA + PRP + Q = 0 \tag{5.11}$$

tiene una solución positiva. Siguiendo este hecho, se introduce adicionalmente la siguiente suposición.

Suposición 5.6 (A6) Para una matriz estable A dada, existe una matriz estrictamente

positiva Q_1 tal que la ecuación de la matriz de Riccati (5.11) con

$$\begin{aligned} R &= \overline{W}_1 + \overline{W}_2 - l_4^{-1} - l_5^{-1}, \quad \overline{W}_1 := W_1^{*T} \Lambda_1^{-1} W_1^*, \quad \overline{W}_2 := W_2^{*T} \Lambda_2^{-1} W_2^* \\ Q &= Q_1 + 2k_v \end{aligned}$$

tiene una solución positiva. Estas condiciones se cumplen fácilmente si se selecciona A como una matriz diagonal estable.

5.2.5 Regla de actualización de pesos

La regla dinámica de actualización de los pesos dada por

$$\begin{aligned} \dot{\widetilde{W}}_{1,t} &= -s_t \left[K_1 P \sigma^T r + K_1 P r \mathbf{x}^T \widetilde{V}_1^T D_\sigma \right] \\ \dot{\widetilde{W}}_{2,t} &= -s_t \left[K_3 P \phi^T u_c r + K_3 P u_c r \mathbf{x}^T \widetilde{V}_2^T D_\phi \right] \\ \dot{\widetilde{V}}_{1,t} &= -s_t \left[K_2 P D_\sigma \widehat{W}_1^T \mathbf{x}^T r + K_2 \widetilde{V}_1 \frac{l_2}{2} I x x^T \Lambda_2 \right] \\ \dot{\widetilde{V}}_{2,t} &= -s_t \left[K_4 P D_\phi \widehat{W}_2^T \mathbf{x}^T u_c r + K_4 \widetilde{V}_2 \frac{l_3}{2} |u_c|^2 I x x^T \Lambda_3 \right] \end{aligned} \quad (5.12)$$

donde $K_{1,2} \in \mathfrak{R}^1$, $K_{3,4} \in \mathfrak{R}^{m \times m}$, y I es la matriz identidad, P es la solución de la ecuación matricial de Riccati. Las condiciones iniciales son $W_{1,0} = W_1^*$, $W_{2,0} = W_2^*$, $V_{1,0} = V_1^*$, $V_{2,0} = V_2^*$. s_t es la función de zona muerta definida como:

$$s_t := \left[1 - \frac{\mu}{\|P^{1/2} \Delta\|} \right]_+, \quad [z]_+ = \begin{cases} z & z \geq 0 \\ 0 & z < 0 \end{cases} \quad (5.13)$$

$$\mu = \frac{\overline{\eta}}{\lambda_{\min} \left(P^{-\frac{1}{2}} Q_1 P^{-\frac{1}{2}} \right)} \quad (5.14)$$

donde $\overline{\eta} = \overline{\varepsilon}_1 + \overline{\varepsilon}_2 u$.

5.2.6 Análisis de Estabilidad

Teorema 5.1 *Se considera el sistema no-lineal (5.1) y la red neuronal dinámica multi-capa (5.2). Si se cumplen las suposiciones A1-A6 y los pesos de esta red se ajustan de acuerdo a (5.12). Los siguientes hechos se cumplen:*

$$r_t, W_{1,t}, W_{2,t}, V_{1,t}, V_{2,t} \in L^\infty \quad (5.15)$$

el error de seguimiento r_t satisface el siguiente desempeño:

$$\limsup_{T \rightarrow \infty} \frac{1}{T} \int_0^T r_t^T Q_1 r_t s_t dt \leq \bar{\eta} \quad (5.16)$$

Prueba. Dado $\mathbf{x}_d(t)$, K_v y las aproximaciones de la red neuronal cumplen con la precisión $\varepsilon_{1,2}$.

La expresión de error:

$$\dot{r}_t = -K_v r + (f - \hat{f}) + (g - \hat{g}) u_c \quad (5.17)$$

Sea la función candidata de Lyapunov

$$V_t := \left[\left\| P^{\frac{1}{2}} r_t \right\| - \mu \right]_+^2 + tr \left[\widetilde{W}_{1,t}^T K_1^{-1} \widetilde{W}_{1,t} \right] + tr \left[\widetilde{W}_{2,t}^T K_2^{-1} \widetilde{W}_{2,t} \right] \\ + tr \left[\widetilde{V}_{1,t} K_3^{-1} \widetilde{V}_{1,t}^T \right] + tr \left[\widetilde{V}_{2,t} K_4^{-1} \widetilde{V}_{2,t}^T \right]$$

donde $P = P^T > 0$, entonces con base en el Lema 2 (ver Apéndice), se calcula la derivada de V_t :

$$\dot{V}_t \leq 2 \left[\left\| P^{\frac{1}{2}} r_t \right\| - \mu \right]_+ \frac{(P^{\frac{1}{2}} \Delta_t)^T}{\|P^{\frac{1}{2}} \Delta_t\|} P^{\frac{1}{2}} \dot{r}_t + 2tr \left[\begin{matrix} \dot{\cdot}^T \\ \widetilde{W}_{1,t} K_1^{-1} \widetilde{W}_{1,t} \end{matrix} \right] \\ + 2tr \left[\begin{matrix} \dot{\cdot}^T \\ \widetilde{W}_{2,t} K_2^{-1} \widetilde{W}_{2,t} \end{matrix} \right] + 2tr \left[\begin{matrix} \dot{\cdot} \\ \widetilde{V}_{1,t} K_3^{-1} \widetilde{V}_{1,t}^T \end{matrix} \right] + 2tr \left[\begin{matrix} \dot{\cdot} \\ \widetilde{V}_{2,t} K_4^{-1} \widetilde{V}_{2,t}^T \end{matrix} \right]$$

$$\begin{aligned}
&= 2 \left[1 - \mu \left\| P^{\frac{1}{2}} r_t \right\|^{-1} \right]_+ r_t P \dot{r}_t + 2tr \left[\begin{smallmatrix} \dot{\cdot} \\ \widetilde{W}_{1,t} \end{smallmatrix}^T K_1^{-1} \widetilde{W}_{1,t} \right] \\
&+ 2tr \left[\begin{smallmatrix} \dot{\cdot} \\ \widetilde{W}_{2,t} \end{smallmatrix}^T K_2^{-1} \widetilde{W}_{2,t} \right] + 2tr \left[\begin{smallmatrix} \dot{\cdot} \\ \widetilde{V}_{1,t} \end{smallmatrix} K_3^{-1} \widetilde{V}_{1,t}^T \right] + 2tr \left[\begin{smallmatrix} \dot{\cdot} \\ \widetilde{V}_{2,t} \end{smallmatrix} K_4^{-1} \widetilde{V}_{2,t}^T \right]
\end{aligned} \tag{5.18}$$

Si se define s_t como en (5.13), entonces (5.18) se obtiene:

$$\begin{aligned}
\dot{V}_t &\leq s_t 2r_t P \left(\dot{r}_t \right) + 2tr \left[\begin{smallmatrix} \dot{\cdot} \\ \widetilde{W}_{1,t} \end{smallmatrix}^T K_1^{-1} \widetilde{W}_{1,t} \right] + 2tr \left[\begin{smallmatrix} \dot{\cdot} \\ \widetilde{W}_{2,t} \end{smallmatrix}^T K_2^{-1} \widetilde{W}_{2,t} \right] \\
&+ 2tr \left[\begin{smallmatrix} \dot{\cdot} \\ \widetilde{V}_{1,t} \end{smallmatrix} K_3^{-1} \widetilde{V}_{1,t}^T \right] + 2tr \left[\begin{smallmatrix} \dot{\cdot} \\ \widetilde{V}_{2,t} \end{smallmatrix} K_4^{-1} \widetilde{V}_{2,t}^T \right]
\end{aligned}$$

Sustituyendo (5.17) se obtiene:

$$\begin{aligned}
\dot{V}_t &\leq s_t 2r_t P \left(-K_v r + (f - \widehat{f}) + (g - \widehat{g}) u_c \right) \\
&+ 2tr \left[\begin{smallmatrix} \dot{\cdot} \\ \widetilde{W}_{1,t} \end{smallmatrix}^T K_1^{-1} \widetilde{W}_{1,t} \right] + 2tr \left[\begin{smallmatrix} \dot{\cdot} \\ \widetilde{W}_{2,t} \end{smallmatrix}^T K_2^{-1} \widetilde{W}_{2,t} \right] \\
&+ 2tr \left[\begin{smallmatrix} \dot{\cdot} \\ \widetilde{V}_{1,t} \end{smallmatrix} K_3^{-1} \widetilde{V}_{1,t}^T \right] + 2tr \left[\begin{smallmatrix} \dot{\cdot} \\ \widetilde{V}_{2,t} \end{smallmatrix} K_4^{-1} \widetilde{V}_{2,t}^T \right]
\end{aligned}$$

Usando (5.7) y A2 (5.9) se tiene:

$$\begin{aligned}
f - \widehat{f} &= W_1^* \sigma(V_1^* \mathbf{x}) - \widehat{W}_1 \sigma(\widehat{V}_1 \mathbf{x}) + \varepsilon_1 \\
&= W_1^* \left[\sigma(V_1^* \mathbf{x}) - \sigma(\widehat{V}_1 \mathbf{x}) \right] + (W_1^* - \widehat{W}_1) \sigma(\widehat{V}_1 \mathbf{x}) + \varepsilon_1 \\
&= W_1^* D_\sigma \widetilde{V}_1 \mathbf{x} + W_1^* \nu_\sigma + \widehat{W}_1 \sigma(\widehat{V}_1 \mathbf{x}) + \varepsilon_1 \\
&= \widehat{W}_1 D_\sigma \widetilde{V}_1 \mathbf{x} + \widetilde{W}_1 D_\sigma \widetilde{V}_1 \mathbf{x} + W_1^* \nu_\sigma + \widehat{W}_1 \sigma(\widehat{V}_1 \mathbf{x}) + \varepsilon_1
\end{aligned} \tag{5.19}$$

y

$$g - \widehat{g} = \widehat{W}_2 D_\phi \widetilde{V}_2 \mathbf{x} + \widetilde{W}_2 D_\phi \widetilde{V}_2 \mathbf{x} + W_2^* \nu_\phi + \widehat{W}_2 \phi(\widehat{V}_2 \mathbf{x}) + \varepsilon_2 \tag{5.20}$$

De (5.19) y (5.20) se obtiene:

$$s_t 2r_t P \left(\dot{r}_t \right) = 2r_t P \{ -K_v r_t + \widehat{W}_1 D_\sigma \widetilde{V}_1 \mathbf{x} + \widetilde{W}_1 D_\sigma \widetilde{V}_1 \mathbf{x} + W_1^* \nu_\sigma + \widetilde{W}_1 \sigma \left(\widehat{V}_1 \mathbf{x} \right) + \varepsilon_1 + \widehat{W}_2 D_\phi \widetilde{V}_2 \mathbf{x} u_c + \widetilde{W}_2 D_\phi \widetilde{V}_2 \mathbf{x} u_c + W_2^* \nu_\phi u_c + \widetilde{W}_2 \phi \left(\widehat{V}_2 \mathbf{x} \right) u_c + \varepsilon_2 u_c \} \quad (5.21)$$

Usando la desigualdad matricial:

$$X^T Y + (X^T Y)^T \leq X^T \Lambda^{-1} X + Y^T \Lambda Y \quad (5.22)$$

la cual es valida para cualquier $X, Y \in \mathfrak{R}^{n \times k}$ y para cualquier matriz definida positiva $0 < \Lambda = \Lambda^T \in \mathfrak{R}^{n \times n}$. El término $2r_t P \varepsilon_1$ puede ser rescrito como:

$$2r_t P \varepsilon_1 \leq r_t^T P l_4^{-1} P r_t + \varepsilon_1^2 l_4 \leq r_t^T P l_4^{-1} P r_t + \overline{\varepsilon}_1 \quad (5.23)$$

$2r_t P \varepsilon_2 u_c$ puede ser rescrito como:

$$2r_t P \varepsilon_2 u_c \leq r_t^T P l_5^{-1} P r_t + \varepsilon_2^2 u_c^2 l_5 \leq r_t^T P l_5^{-1} P r_t + \overline{\varepsilon}_2 \overline{u} \quad (5.24)$$

$2r_t P W_1^* \nu_\sigma$ puede ser rescrito como:

$$\begin{aligned} 2r_t P W_1^* \nu_\sigma &\leq r_t^T P W_1^{T*} \Lambda_2^{-1} W_1^* P r_t + \nu_\sigma^T \Lambda_2 \nu_\sigma \leq r_t^T P \overline{W}_1^* P r_t + l_2 \left\| \widetilde{V}_1 x \right\|_{\Lambda_2}^2 \\ &\leq l_2 \widetilde{V}_1 \Lambda_2 x \end{aligned} \quad (5.25)$$

$2r_t P W_2^* \nu_\phi u_c$ puede ser rescrito como:

$$\begin{aligned} 2r_t P W_2^* \nu_\phi u_c &\leq r_t^T P W_2^{T*} \Lambda_3^{-1} W_2^* P r_t + |u_c|^2 \nu_\phi^T \Lambda_3 \nu_\phi \leq r_t^T P \overline{W}_2^* P r_t + l_3 |u_c|^2 \left\| \widetilde{V}_2 x \right\|_{\Lambda_3}^2 \\ &\leq l_3 |u_c|^2 \widetilde{V}_2 \Lambda_3 x \end{aligned} \quad (5.26)$$

donde μ es definido como en (5.14). Asi, V_t es acotada, por lo que (5.15) es probada.

Desde que $0 \leq s_t \leq 1$, de (5.30) se tiene:

$$\dot{V}_t \leq -r_t^T Q_1 r_t s_t + \bar{\eta} s_t \leq -r_t^T Q_0 r_t s_t + \bar{\eta}$$

Integrando (5.29) de 0 hasta T se obtiene:

$$V_T - V_0 \leq - \int_0^T r_t^T Q_0 r_t s_t dt + \bar{\eta} T$$

asi,

$$\int_0^T r_t^T Q_1 r_t s_t dt \leq V_0 - V_T + \bar{\eta} T \leq V_0 + \bar{\eta} T$$

Dado que $W_{1,0} = W_1^*$, $W_{2,0} = W_2^*$, $V_{1,0} = V_1^*$, $V_{2,0} = V_2^*$, V_0 estan acotadas, (5.16) es probada. ■

5.2.7 Simulaciones

Problema 5.3 *Hallar la ley de control directo, que resuelva el problema de seguimiento con $x_d = [0, 0, 0, 0]$ para el sistema no lineal TORA (3.7), usando una red neuronal dinámica.*

Solución 5.1 *Se usa la red neuronal dinámica multicapa (5.2), junto con la regla de actualización de pesos (5.12) y $W_{1,2} \in \mathfrak{R}^{1 \times 6}$, $V_{1,2} \in \mathfrak{R}^{6 \times 4}$, $\sigma(x) = \frac{2}{(1+e^{-2x})} - 0.5$. Las condiciones iniciales para las matrices de los pesos son:*

$$W_{1,2} = \begin{bmatrix} 1 & 1 & 1 & 2 & 1 & 2 \end{bmatrix} \quad V_{1,2} = \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & 2 & 1 & 2 \\ 2 & 1 & 2 & 1 \\ 1 & 1 & 1 & 1 \\ 1 & 2 & 1 & 2 \\ 2 & 1 & 2 & 1 \end{bmatrix}$$

con $\bar{\eta} = 0.2$, $K_{1,2,3,4} := 0.005 I_6$, $l_2 = l_3 = 1$, $\Lambda = [-1, -1, -1]$, $\Lambda_2 = \Lambda_3 = 0.01I_4$, $K_v = 0.1$, para obtener la acción de control. Las simulaciones se llevaron a cabo usando el paquete matlab y el simulink, el diagrama general se muestra en la figura B-4. Los resultados obtenidos de aplicar la ley de control se muestran en las figuras 5-2, 5-3, 5-4, 5-5, además la señal de control se muestra en la figura 5-1.

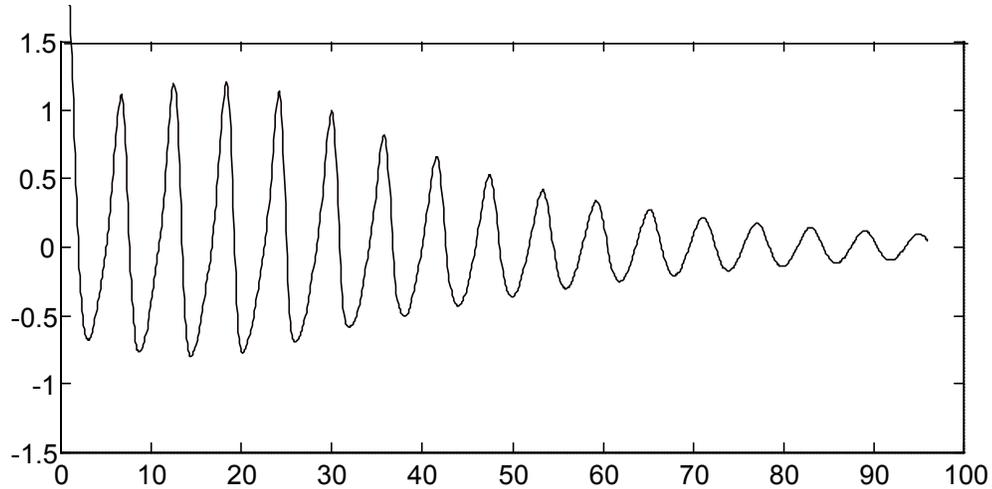


Figura 5-1: Comportamiento en tiempo de la ley de control (5.4).

Observación 5.1 Como se puede ver en las figuras anteriores, la acción de control obtenida a partir de la RND multicapa aplicando la actualización de los pesos en línea, es satisfactorio. Es importante mencionar que la RND multicapa usada para obtener estos resultados corresponde al caso cuando general cuando $W_2^* \phi(V_2^* x_t) \neq 1$.

Observación 5.2 El rango de ajuste de las ganancias $K_{1,2}$ están entre $[0.001, 0.1]$ por lo que la sintonización requirió de bastante tiempo hasta obtener el valor final de 0.005. Sin embargo este resultado puede compararse por el obtenido en [16], donde la acción de control utiliza directamente los estados del sistema y elimina directamente algunas no linealidades para obtener un resultado muy similar al aquí presentado, por que la dificultad para hallar el valor de las ganancias se justifica.

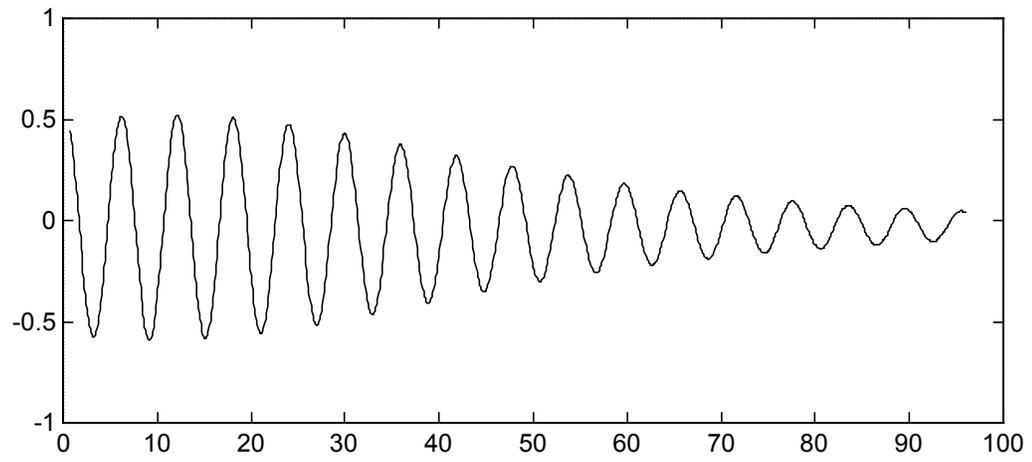


Figura 5-2: Control Directo para el estado q .

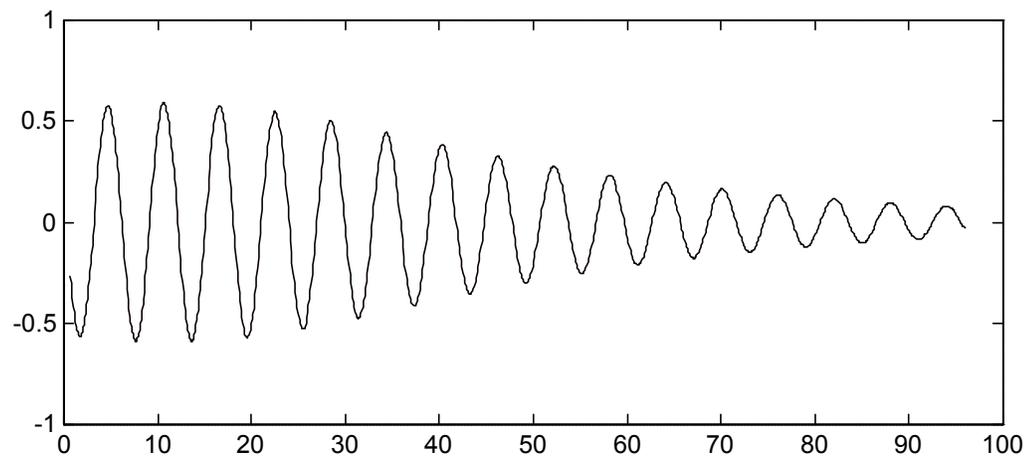


Figura 5-3: Control Directo para el estado \dot{q} .

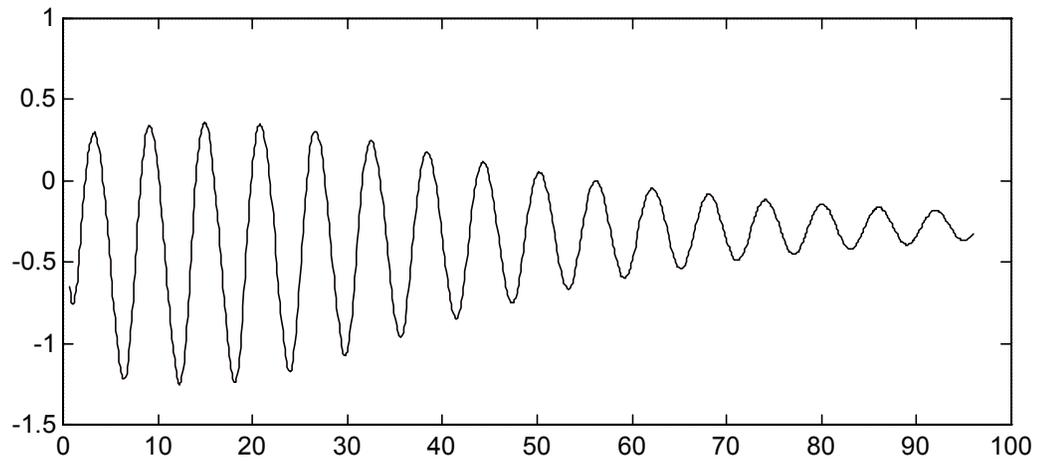


Figura 5-4: Control Directo para el estado θ .

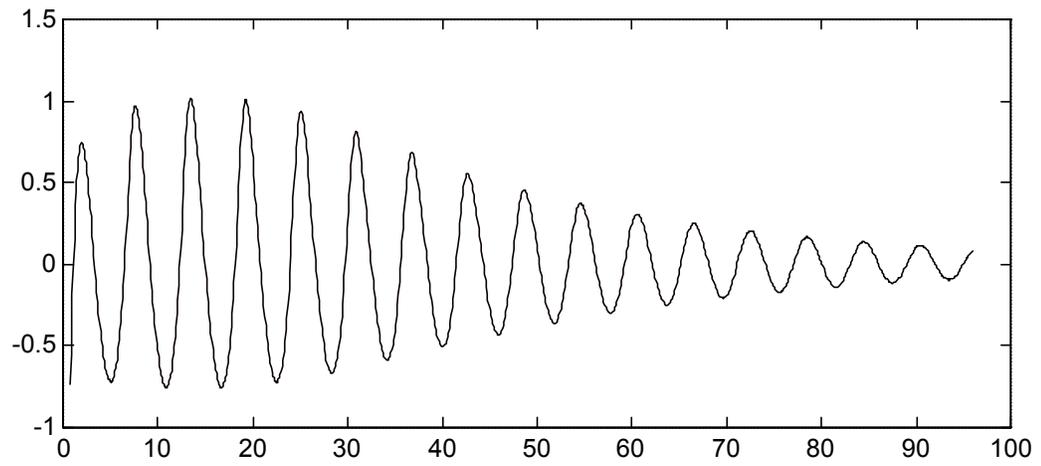


Figura 5-5: Control Directo para el estado $\dot{\theta}$.

5.3 Control Adaptable Indirecto

5.3.1 Planteamiento del Problema

La clase de sistemas no-lineales:

$$\begin{aligned}
 \dot{x}_1 &= x_2 \\
 \dot{x}_2 &= x_3 \\
 &\vdots \\
 \dot{x}_n &= f(\mathbf{x}) + g(\mathbf{x})u \\
 y &= x_1
 \end{aligned} \tag{5.31}$$

con $x = [x_1, x_2, \dots, x_n]^T$, y $f, g : \mathfrak{R}^n \rightarrow \mathfrak{R}$ funciones suaves desconocidas con $f(\mathbf{0}) = 0$

Problema 5.4 *Identificar el sistema (5.31) y diseñar el neuro-controlador u que resuelva el problema de seguimiento, usando una red neuronal dinámica*

5.3.2 Estructura de la red neuronal dinámica

La RND multicapa tiene la siguiente estructura:

$$\begin{aligned}
 \dot{\hat{x}}_1 &= \hat{x}_2 \\
 \dot{\hat{x}}_2 &= \hat{x}_3 \\
 &\vdots \\
 \dot{\hat{x}}_n &= a\hat{x}_n + W_{1,t}\sigma(V_{1,t}\hat{\mathbf{x}}) + W_{2,t}\phi(V_{2,t}\hat{\mathbf{x}})u \\
 \hat{y} &= \hat{x}_1
 \end{aligned} \tag{5.32}$$

donde $\forall t \in [0, \infty)$, el vector $\hat{x}_t \in \mathfrak{R}^n$ son los estados de la RND, $a \in \mathfrak{R}$, con $a < 0$ y será especificado despues. Las matrices $W_i \in \mathfrak{R}^{1 \times m}$, $V_i \in \mathfrak{R}^{m \times n}$, $i = f, g$ son los pesos de la red que describen las conexiones de las capas ocultas y de salida, $u_t \in \mathfrak{R}^1$ es el control dado. $\sigma(\cdot) \in \mathfrak{R}^m$ es una función vectorial sigmoideal, $\phi(\cdot)$ es $\mathfrak{R}^{m \times m}$ una matriz diagonal,

i.e.,

$$\phi(\cdot) = \text{diag} [\phi_1(V_{2,t}\widehat{\mathbf{x}})_1 \cdots \phi_m(V_{2,t}\widehat{\mathbf{x}})_m].$$

Los elementos de $\sigma_i(\cdot)$ (como los de $\phi_i(\cdot)$) son usualmente funciones sigmoideas, i.e.,

$$\sigma_i(x_i) = \frac{a_i}{1 + e^{-b_i x_i}} - c_i.$$

5.3.3 Preliminares

Considerese las siguientes suposiciones:

Suposición 5.7 (A6) *El control u_t es seleccionado como acotado, entonces:*

$$u_t \leq \bar{u}$$

En el caso general, cuando la RND (5.32) no puede identificar exactamente el sistema no-lineal dado (5.31), el sistema no-lineal (5.31) puede ser representado como:

$$\dot{x}_n = ax_n + W_1^* \sigma(V_1^* \mathbf{x}) + W_2^* \phi(V_2^* \mathbf{x}) u_t + \tilde{f}_t. \quad (5.33)$$

donde \tilde{f}_t es el error de modelado, W_1^* , W_2^* , V_1^* y V_2^* son matrices constantes. Dado que las funciones sigmoideas σ y ϕ son uniformemente acotadas. Es razonable asumir que las dinámicas no-modeladas \tilde{f}_t , satisface la siguiente suposición.

Suposición 5.8 (A7) *Existe una constante positiva $\bar{\eta}$ tal que*

$$\left\| \tilde{f}_t \right\|_{\Lambda_f}^2 \leq \bar{\eta}, \quad \Lambda_f = \Lambda_f^T > 0$$

Definición 5.2 *Sea el error de identificación:*

$$\Delta_t := \mathbf{x} - \widehat{\mathbf{x}}$$

Suposición 5.9 (A8) Dado que σ es una función sigmoideal, satisface la condición de Lipschitz, basado el Lema 1 (ver Appendix) se concluye que

$$\begin{aligned}\tilde{\sigma}_t^T \Lambda_1 \tilde{\sigma}_t &\leq \Delta_t^T \Lambda_\sigma \Delta_t, & (\tilde{\phi}_t u_t)^T \Lambda_2 (\tilde{\phi}_t u_t) &\leq \bar{u}^2 \Delta_t^T \Lambda_\phi \Delta_t \\ \tilde{\sigma}'_t &= D_\sigma \tilde{V}_{1,t} \hat{x}_t + \nu_\sigma, & \tilde{\phi}'_t u_t &:= D_\phi \tilde{V}_{2,t} \hat{x}_t + \nu_\phi\end{aligned}$$

donde

$$\begin{aligned}\tilde{\sigma}_t &:= \sigma(V_1^* x_t) - \sigma(V_1^* \hat{x}_t), & \tilde{\phi}_t &:= \phi(V_2^* x_t) - \phi(V_2^* \hat{x}_t) \\ \tilde{\sigma}'_t &:= \sigma(V_1^* \hat{x}_t) - \sigma(V_{1,t} \hat{x}_t), & \tilde{\phi}'_t u_t &:= \phi(V_2^* \hat{x}_t) u_t - \phi(V_{2,t} \hat{x}_t) u_t \\ D_\sigma &= \frac{\partial \sigma^T(Z)}{\partial Z} \Big|_{Z=V_{1,t} \hat{x}_t} \in \mathfrak{R}^{m \times m}, & \|\nu_\sigma\|_{\Lambda_1}^2 &\leq l_1 \left\| \tilde{V}_{1,t} \hat{x}_t \right\|_{\Lambda_1}^2, & l_1 &> 0 \\ D_\phi &= \frac{\partial (\phi_i u_i)^T(Z)}{\partial Z} \Big|_{Z=V_{2,t} \hat{x}_t} \in \mathfrak{R}^{m \times m}, & \|\nu_\phi\|_{\Lambda_2}^2 &\leq l_2 \left\| \tilde{V}_{2,t} \hat{x}_t \right\|_{\Lambda_2}^2, & l_2 &> 0, \\ \tilde{V}_{1,t} &:= V_1^* - V_{1,t}, & \tilde{V}_{2,t} &:= V_2^* - V_{2,t} \\ \tilde{W}_{1,t} &:= W_1^* - W_{1,t}, & \tilde{W}_{2,t} &:= W_2^* - W_{2,t}\end{aligned}$$

$\Lambda_1, \Lambda_2, \Lambda_\sigma$ y Λ_ϕ son matrices definidas positivas.

Se conoce [44] que si la matriz A es estable, el par $(A, R^{1/2})$ es controlable, el par $(Q^{1/2}, A)$ es observable, y la *condición especial local de frecuencia* o su matriz equivalente:

$$A^T R^{-1} A - Q \geq \frac{1}{4} [A^T R^{-1} - R^{-1} A] R [A^T R^{-1} - R^{-1} A]^T \geq 0 \quad (5.34)$$

se cumple, entonces la matriz de Riccati

$$A^T P + PA + PRP + Q = 0 \quad (5.35)$$

tiene una solución positiva. Siguiendo este hecho, se introduce adicionalmente la siguiente suposición.

Suposición 5.10 (A9) Para una matriz estable A dada, existe una matriz estricta-

mente positiva Q_1 tal que la ecuación de la matriz de Riccati (5.35) con

$$\begin{aligned} R &= 2\overline{W}_1 + 2\overline{W}_2 + \Lambda_f^{-1}, \quad \overline{W}_1 := W_1^{*T} \Lambda_1^{-1} W_1^*, \quad \overline{W}_2 := W_2^{*T} \Lambda_2^{-1} W_2^* \\ Q &= Q_1 + \Lambda_\sigma + \overline{u}^2 \Lambda_\phi, \end{aligned}$$

tiene una solución positiva. Estas condiciones se cumplen fácilmente si se selecciona A como una matriz diagonal estable.

5.3.4 Regla de Actualización de Pesos

La regla dinámica de actualización de los pesos dada por:

$$\begin{aligned} \dot{\widetilde{W}}_{1,t} &= -s_t \left[K_1 P \Delta_t \sigma^T - K_1 P \Delta_t \widehat{x}_t^T \widetilde{V}_{1,t}^T D_\sigma \right] \\ \dot{\widetilde{W}}_{2,t} &= -s_t \left[K_2 P \Delta_t (\phi u_t)^T - K_2 P \Delta_t \widehat{x}_t^T \widetilde{V}_{2,t}^T D_\phi \right] \\ \dot{\widetilde{V}}_{1,t} &= -s_t \left[K_3 P W_{1,t} D_\sigma \Delta_t \widehat{x}_t^T - \frac{l_1}{2} K_3 \Lambda_1 \widetilde{V}_{1,t} \widehat{x}_t \widehat{x}_t^T \right] \\ \dot{\widetilde{V}}_{2,t} &= -s_t \left[K_4 P W_{2,t} D_\phi \Delta_t \widehat{x}_t^T - \frac{l_2}{2} K_4 \Lambda_2 \widetilde{V}_{2,t} \widehat{x}_t \widehat{x}_t^T \right] \end{aligned} \quad (5.36)$$

donde $K_i \in \mathfrak{R}^{n \times n}$ ($i = 1 \dots 4$) son matrices definidas positivas, P es la solución de la ecuación matricial de Riccati dada por (5.35). Las condiciones iniciales son $W_{1,0} = W_1^*$, $W_{2,0} = W_2^*$, $V_{1,0} = V_1^*$, $V_{2,0} = V_2^*$. s_t es la función de zona muerta definida como:

$$s_t := \left[1 - \frac{\mu}{\|P^{1/2} \Delta\|} \right]_+, \quad [z]_+ = \begin{cases} z & z \geq 0 \\ 0 & z < 0 \end{cases} \quad (5.37)$$

$$\mu = \frac{\overline{\eta}}{\lambda_{\min} \left(P^{-\frac{1}{2}} Q_1 P^{-\frac{1}{2}} \right)} \quad (5.38)$$

5.3.5 Análisis de Estabilidad

Teorema 5.2 *Se considera el sistema no-lineal (5.31) y la red neuronal dinámica multicapa (5.32). Si se cumplen las suposiciones A5-A9 y los pesos de esta red se ajustan de*

acuerdo a (5.36). Los siguientes hechos se cumplen:

$$\Delta_t, W_{1,t}, W_{2,t}, V_{1,t}, V_{2,t} \in L^\infty \quad (5.39)$$

el error de identificación Δ_t satisface el siguiente desempeño de seguimiento:

$$\limsup_{T \rightarrow \infty} \frac{1}{T} \int_0^T \Delta_t^T Q_1 \Delta_t s_t dt \leq \bar{\eta} \quad (5.40)$$

Prueba. De (5.32) y (5.33) la ecuación del error puede ser expresada como:

$$\dot{\Delta}_t = A\Delta_t + \widetilde{W}_{1,t}\sigma + \widetilde{W}_{2,t}\phi u_t + W_1^* \tilde{\sigma}_t + W_2^* \tilde{\phi}_t u_t + W_1^* \tilde{\sigma}'_t + W_2^* \tilde{\phi}'_t u_t + \tilde{f}_t. \quad (5.41)$$

Sea la función candidata de Lyapunov:

$$V_t := \left[\left\| P^{\frac{1}{2}} \Delta_t \right\| - \mu \right]_+^2 + tr \left[\widetilde{W}_{1,t}^T K_1^{-1} \widetilde{W}_{1,t} \right] + tr \left[\widetilde{W}_{2,t}^T K_2^{-1} \widetilde{W}_{2,t} \right] \\ + tr \left[\widetilde{V}_{1,t} K_3^{-1} \widetilde{V}_{1,t}^T \right] + tr \left[\widetilde{V}_{2,t} K_4^{-1} \widetilde{V}_{2,t}^T \right]$$

donde $P = P^T > 0$, entonces con base en el Lema 2 (ver Apéndice), se calcula la derivada de V_t :

$$\dot{V}_t \leq 2 \left[\left\| P^{\frac{1}{2}} \Delta_t \right\| - \mu \right]_+ \frac{(P^{\frac{1}{2}} \Delta_t)^T}{\left\| P^{\frac{1}{2}} \Delta_t \right\|} P^{\frac{1}{2}} \dot{\Delta}_t + 2tr \left[\dot{\widetilde{W}}_{1,t}^T K_1^{-1} \widetilde{W}_{1,t} \right] \\ + 2tr \left[\dot{\widetilde{W}}_{2,t}^T K_2^{-1} \widetilde{W}_{2,t} \right] + 2tr \left[\dot{\widetilde{V}}_{1,t} K_3^{-1} \widetilde{V}_{1,t}^T \right] + 2tr \left[\dot{\widetilde{V}}_{2,t} K_4^{-1} \widetilde{V}_{2,t}^T \right] \quad (5.42) \\ = 2 \left[1 - \mu \left\| P^{\frac{1}{2}} \Delta_t \right\|^{-1} \right]_+ \Delta P^{\frac{1}{2}} \dot{\Delta}_t + 2tr \left[\dot{\widetilde{W}}_{1,t}^T K_1^{-1} \widetilde{W}_{1,t} \right] \\ + 2tr \left[\dot{\widetilde{W}}_{2,t}^T K_2^{-1} \widetilde{W}_{2,t} \right] + 2tr \left[\dot{\widetilde{V}}_{1,t} K_3^{-1} \widetilde{V}_{1,t}^T \right] + 2tr \left[\dot{\widetilde{V}}_{2,t} K_4^{-1} \widetilde{V}_{2,t}^T \right]$$

Si se define s_t como en (5.37), entonces (5.42) se obtiene:

$$\begin{aligned} \dot{V}_t \leq & s_t 2\Delta^T P \dot{\Delta} + 2tr \left[\begin{array}{c} \cdot \\ \cdot \\ \cdot \end{array} \begin{array}{c} T \\ \\ \end{array} \begin{array}{c} \widetilde{W}_{1,t} \\ K_1^{-1} \widetilde{W}_{1,t} \end{array} \right] + 2tr \left[\begin{array}{c} \cdot \\ \cdot \\ \cdot \end{array} \begin{array}{c} T \\ \\ \end{array} \begin{array}{c} \widetilde{W}_{2,t} \\ K_2^{-1} \widetilde{W}_{2,t} \end{array} \right] \\ & + 2tr \left[\begin{array}{c} \cdot \\ \cdot \\ \cdot \end{array} \begin{array}{c} T \\ \\ \end{array} \begin{array}{c} \widetilde{V}_{1,t} \\ K_3^{-1} \widetilde{V}_{1,t} \end{array} \right] + 2tr \left[\begin{array}{c} \cdot \\ \cdot \\ \cdot \end{array} \begin{array}{c} T \\ \\ \end{array} \begin{array}{c} \widetilde{V}_{2,t} \\ K_4^{-1} \widetilde{V}_{2,t} \end{array} \right] \end{aligned}$$

De (5.41) se tiene:

$$\begin{aligned} 2\Delta_t^T P \dot{\Delta}_t = & 2\Delta_t^T P A \Delta_t + 2\Delta_t^T P \left(W_1^* \tilde{\sigma}_t + W_2^* \tilde{\phi}_t u_t \right) \\ & + 2\Delta_t^T P \left(\widetilde{W}_{1,t} \sigma + \widetilde{W}_{2,t} \phi u_t \right) + 2\Delta_t^T P \left(W_1^* \tilde{\sigma}'_t + W_2^* \tilde{\phi}'_t u_t \right). \end{aligned} \quad (5.43)$$

Usando la desigualdad matricial:

$$X^T Y + (X^T Y)^T \leq X^T \Lambda^{-1} X + Y^T \Lambda Y \quad (5.44)$$

la cual es valida para cualquier $X, Y \in \mathfrak{R}^{n \times k}$ y para cualquier matriz definida positiva $0 < \Lambda = \Lambda^T \in \mathfrak{R}^{n \times n}$, y usando **A8** $\Delta_t^T P \left(W_1^* \tilde{\sigma}_t + W_2^* \tilde{\phi}_t u_t \right)$ en (5.43) puede ser reescrita como

$$\begin{aligned} 2\Delta_t^T P W_1^* \tilde{\sigma}_t & \leq \Delta_t^T P W_1^* \Lambda_1^{-1} W_1^{*T} P \Delta_t + \tilde{\sigma}_t^T \Lambda_1 \tilde{\sigma}_t \leq \Delta_t^T (P \overline{W}_1 P + \Lambda_\sigma) \Delta_t \\ 2\Delta_t^T P W_2^* \tilde{\phi}_t u_t & \leq \Delta_t^T (P \overline{W}_2 P + \overline{u}^2 \Lambda_\phi) \Delta_t. \end{aligned} \quad (5.45)$$

Usando **A8**, el último término en (5.43) puede ser reescrita como:

$$\begin{aligned} 2\Delta_t^T P W_1^* \tilde{\sigma}'_t & = 2\Delta_t^T P W_{1,t} D_\sigma \widetilde{V}_{1,t} \widehat{x}_t + 2\Delta_t^T P \widetilde{W}_{1,t} D_\sigma \widetilde{V}_{1,t} \widehat{x}_t + 2\Delta_t^T P W_1^* \nu_\sigma \\ 2\Delta_t^T P W_2^* \tilde{\phi}'_t u_t & = 2\Delta_t^T P W_{2,t} D_\phi \widetilde{V}_{2,t} \widehat{x}_t + 2\Delta_t^T P \widetilde{W}_{2,t} D_\phi \widetilde{V}_{2,t} \widehat{x}_t + 2\Delta_t^T P W_2^* \nu_\phi \end{aligned} \quad (5.46)$$

El término $2\Delta_t^T P W_1^* \nu_\sigma$ en (5.46) puede ser reescrito como:

$$2\Delta_t^T P W_1^* \nu_\sigma \leq \Delta_t^T P W_1^{*T} \Lambda_1^{-1} W_1^* P \Delta_t + \nu_\sigma^T \Lambda_1 \nu_\sigma \leq \Delta_t^T P \overline{W}_1 P \Delta_t + l_1 \left\| \widetilde{V}_{1,t} \widehat{x}_t \right\|_{\Lambda_1}^2 \quad (5.47)$$

asi como $2\Delta_t^T P W_2^* \nu_\phi$ en (5.46) puede ser reescrito como:

$$2\Delta_t^T P W_2^* \nu_\phi \leq \Delta_t^T P \bar{W}_2 P \Delta_t + l_2 \left\| \tilde{V}_{2,t} \hat{x}_t \right\|_{\Lambda_2}^2. \quad (5.48)$$

De **A7** $2\Delta_t^T P \tilde{f}_t$ puede ser estimado como:

$$2\Delta_t^T P \tilde{f}_t \leq \Delta_t^T P \Lambda_f^{-1} P \Delta_t + \tilde{f}_t^T \Lambda_f \tilde{f}_t \leq \Delta_t^T P \Lambda_f^{-1} P \Delta_t + s_t \bar{\eta} \quad (5.49)$$

Usando (5.43), (5.45), (5.47), (5.48) y (5.49) se reescribe como:

$$\dot{V}_t \leq s_t \Delta_t^T L \Delta_t + L_{w1} + L_{w2} + L_{v1} + L_{v2} - s_t \Delta_t^T Q_1 \Delta_t + s_t \bar{\eta} \quad (5.50)$$

donde

$$\begin{aligned} L &= A^T P + P A + P R P + Q, \\ L_{w1} &= 2tr \left[\begin{array}{c} \dot{\tilde{W}}_{1,t}^T \\ \tilde{W}_{1,t} \end{array} K_1^{-1} \tilde{W}_{1,t} \right] + 2s_t \Delta_t^T P \tilde{W}_{1,t} \sigma + 2s_t \Delta_t^T P \tilde{W}_{1,t} D_\sigma \tilde{V}_{1,t} \hat{x}_t, \\ L_{w2} &= 2tr \left[\begin{array}{c} \dot{\tilde{W}}_{2,t}^T \\ \tilde{W}_{2,t} \end{array} K_2^{-1} \tilde{W}_{2,t} \right] + 2s_t \Delta_t^T P \tilde{W}_{2,t} \phi u_t + 2s_t \Delta_t^T P \tilde{W}_{2,t} D_\phi \tilde{V}_{2,t} \hat{x}_t \\ L_{v1} &= 2tr \left[\begin{array}{c} \dot{\tilde{V}}_{1,t} \\ \tilde{V}_{1,t} \end{array} K_3^{-1} \tilde{V}_{1,t}^T \right] + 2s_t \Delta_t^T P W_{1,t} D_\sigma \tilde{V}_{1,t} \hat{x}_t + s_t l_1 \left\| \tilde{V}_{1,t} \hat{x}_t \right\|_{\Lambda_1}^2 \\ L_{v2} &= 2tr \left[\begin{array}{c} \dot{\tilde{V}}_{2,t} \\ \tilde{V}_{2,t} \end{array} K_4^{-1} \tilde{V}_{2,t}^T \right] + 2s_t \Delta_t^T P W_{2,t} D_\phi \tilde{V}_{2,t} \hat{x}_t + s_t l_2 \left\| \tilde{V}_{2,t} \hat{x}_t \right\|_{\Lambda_2}^2 \end{aligned}$$

Usando la regla de actualización (5.36), y dado que

$$\begin{aligned} \dot{\tilde{W}}_{1,t} &= -\dot{W}_{1,t}, \quad \dot{\tilde{W}}_{2,t} = -\dot{W}_{2,t} \\ \dot{\tilde{V}}_{1,t} &= -\dot{V}_{1,t}, \quad \dot{\tilde{V}}_{2,t} = -\dot{V}_{2,t} \end{aligned}$$

se obtiene

$$L_{w1} = 0, \quad L_{w2} = 0, \quad L_{v1} = 0, \quad L_{v2} = 0$$

Finalmente, de (5.50) se obtiene que:

$$\dot{V}_t \leq -s_t [\Delta_t^T Q_0 \Delta_t - \bar{\eta}] \quad (5.51)$$

El lado derecho de esta desigualdad puede ser estimada de la siguiente forma:

$$\dot{V}_t \leq -s_t \lambda_{\min} \left(P^{\frac{1}{2}} Q_1 P^{-\frac{1}{2}} \right) \left(\left\| P^{\frac{1}{2}} \Delta_t \right\|^2 - \mu \right) \leq 0 \quad (5.52)$$

donde μ es definido como en (5.38). Así, V_t es acotada, por lo que (5.39) es probada.

Desde que $0 \leq s_t \leq 1$, de (5.52) se tiene:

$$\dot{V}_t \leq -\Delta_t^T Q_1 \Delta_t s_t + \bar{\eta} s_t \leq -\Delta_t^T Q_0 \Delta_t s_t + \bar{\eta}$$

Integrando (5.51) de 0 hasta T se obtiene:

$$V_T - V_0 \leq - \int_0^T \Delta^T Q_0 \Delta s_t dt + \bar{\eta} T$$

así,

$$\int_0^T \Delta^T Q_1 \Delta s_t dt \leq V_0 - V_T + \bar{\eta} T \leq V_0 + \bar{\eta} T$$

Dado que $W_{1,0} = W_1^*$, $W_{2,0} = W_2^*$, $V_{1,0} = V_1^*$, $V_{2,0} = V_2^*$, V_0 están acotadas, (5.40) es probada. ■

5.3.6 Diseño del Neuro Controlador

La linearización por retroalimentación es muy útil para resolver el problema de seguimiento:

Problema 5.5 *Dada una salida deseada $y_d(t)$, encontrar una acción de control u tal que la planta siga una trayectoria deseada con una precisión aceptable (i.e. error de seguimiento acotado), mientras que todos los estados del sistema y el control permanezcan acotados.*

Para resolver este problema, se requieren establecer las siguientes suposiciones las cuales son comunmente usadas. Primero defínase el vector:

$$\mathbf{x}_d(t) = \begin{bmatrix} y_d & \dot{y}_d & \dots & y_d^{(n-1)} \end{bmatrix}^T$$

Suposición 5.11 (A10) *El vector de la trayectoria deseada $\mathbf{x}_d(t)$ es continuo y disponible para su medición, y $\|\mathbf{x}_d(t)\| \leq D$ con D una cota conocida.*

Definición 5.3 *Sea la siguiente semi-norma:*

$$\|z\|_Q^2 = \limsup_{\tau \rightarrow \infty} \frac{1}{\tau} \int_0^\tau z^T(t) Q z(t) dt, \quad (5.53)$$

donde $Q = Q^T > 0$

Asi el seguimiento de esta trayectoria por los estados de sistema, puede formularse como:

$$J_{\min} = \min_{u(t)} J, \quad J = \|\mathbf{x} - \mathbf{x}_d\|_{Q_c}^2 \quad (5.54)$$

Asi, para cualquier $\eta > 0$, se tiene

$$J \leq (1 + \eta) \|\mathbf{x} - \hat{\mathbf{x}}\|_{Q_c}^2 + (1 + \eta^{-1}) \|\hat{\mathbf{x}} - \mathbf{x}_d\|_{Q_c}^2 \quad (5.55)$$

el mínimo del término $\|\mathbf{x} - \hat{\mathbf{x}}\|_{Q_c}^2$ ya ha sido resuelto en la sección anterior. Seleccionando $\tilde{R}_c = (1 + \eta^{-1}) R_c$, ahora podemos reformular nuestro objetivo de control como: minimizar el termino $\|\hat{\mathbf{x}} - \mathbf{x}_d\|_{Q_c}^2$. Entonces

Definición 5.4 *Sea el error de seguimiento:*

$$\mathbf{e} = \hat{\mathbf{x}} - \mathbf{x}_d \quad (5.56)$$

y el filtro del error

$$r = \begin{bmatrix} \Lambda^T & 1 \end{bmatrix} \mathbf{e} \in \mathfrak{R}^1$$

donde $\Lambda = [\lambda_1, \lambda_2, \dots, \lambda_{n-1}]^T$ y los coeficientes del vector son apropiadamente seleccionados, tal que $e \rightarrow 0$ exponencialmente como $r \rightarrow 0$ ($s^{n-1} + \lambda_{n-1}s^{n-2} + \dots + \lambda_1$ es Hurwitz).

Así la derivada en tiempo del filtro del error se reescribe como:

$$\dot{r} = \begin{bmatrix} \lambda_1 & \dots & \lambda_{n-1} & 1 \end{bmatrix} \begin{bmatrix} \hat{x}_1 - \dot{\mathbf{x}}_d \\ \vdots \\ \hat{x}_{n-1} - \mathbf{x}_d^{n-1} \\ \hat{x}_n - \mathbf{x}_d^n \end{bmatrix} \quad (5.57)$$

$$\dot{r} = \lambda_1 (\hat{x}_1 - \dot{\mathbf{x}}_d) + \dots + \lambda_{n-1} (\hat{x}_{n-1} - \mathbf{x}_d^{n-1}) + (\hat{x}_n - \mathbf{x}_d^n)$$

$$\dot{r} = [\Lambda^T \ 0] e + (\hat{x}_n - \mathbf{x}_d^n)$$

$$\dot{r} = \hat{f} + \hat{g}u + Y_d$$

donde $Y_d = ax_n - x_d^{(n)} + [\Lambda^T \ 0] e$

Además

$$\hat{f} = W_{1,t}^* \sigma(V_{1,t}^* \hat{\mathbf{x}}) \quad (5.58)$$

$$\hat{g} = W_{2,t}^* \phi(V_{2,t}^* \hat{\mathbf{x}})$$

Ahora, dado que las funciones f, g no son conocidas, se elige la acción de control $u_c \in R^1$ como:

$$u = \frac{1}{\hat{g}} [-\hat{f} + v] \quad (5.59)$$

donde

$$v = -K_v r - Y_d. \quad (5.60)$$

Así,

$$\begin{aligned} \dot{r} &= \hat{f} + \hat{g}u + Y_d + v - v \\ &= \hat{f} + \hat{g} \left(\frac{1}{\hat{g}} [-\hat{f} + v] \right) + Y_d - Y_d - K_v r - v \\ &= \hat{f} - \hat{f} + v - v - K_v r \\ &= -K_v r \end{aligned} \quad (5.61)$$

Observación 5.3 Si se selecciona $K_v > 0$ el error de seguimiento es estable y además $r \rightarrow 0$

5.3.7 Simulaciones

Problema 5.6 Hallar la ley de control indirecto, que resuelva el problema de seguimiento con $x_d = [0, 0, 0, 0]$ para el sistema no lineal TORA (3.7), usando una red neuronal dinámica.

Solución 5.2 La identificación del sistema TORA (3.7) obtenida usando la RND multicapa (5.32) se muestran en las figs. 5-6, 5-7, 5-8, 5-9.

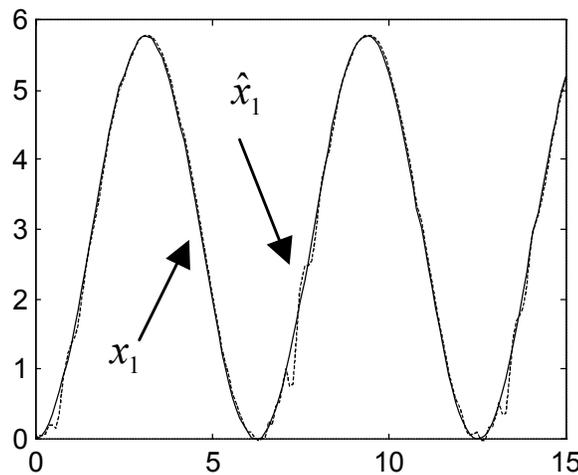


Figura 5-6: Identificación usando una RND multicapa de estado q .

Como se puede ver en estas gráficas el resultado no es satisfactorio. Por lo que:

Solución 5.3 Se usa la red neuronal dinámica de una capa (ver Observación C1) (4.33), junto con la regla de actualización de pesos (4.6) y $W_{1,2} \in \mathfrak{R}^{1 \times 4}$, $\sigma(x) = \frac{2}{(1+e^{-2x})} - 0.5$. Las condiciones iniciales para las matrices de los pesos son:

$$W_1 = \begin{bmatrix} 1 & 1 & 1 & 2 \end{bmatrix} \quad W_2 = \begin{bmatrix} 1 & 2 & 1 & 2 \end{bmatrix}$$

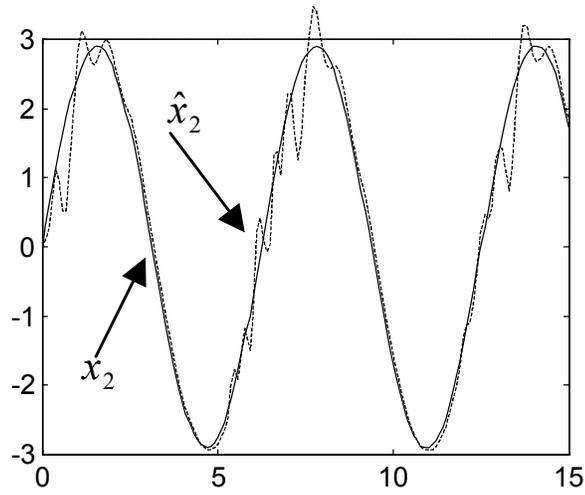


Figura 5-7: Identificación usando una RND multicapa de estado \dot{q} .

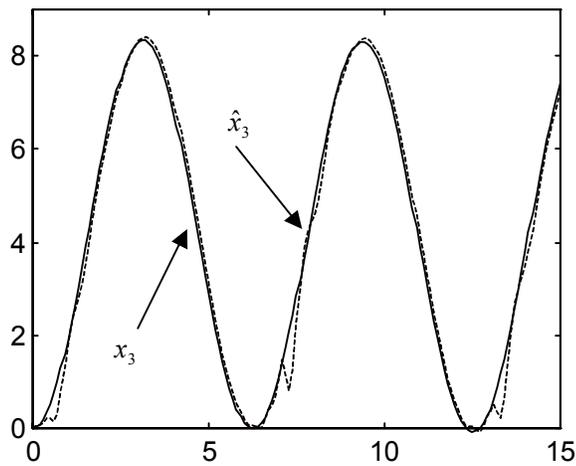


Figura 5-8: Identificación usando una RND multicapa de estado θ .

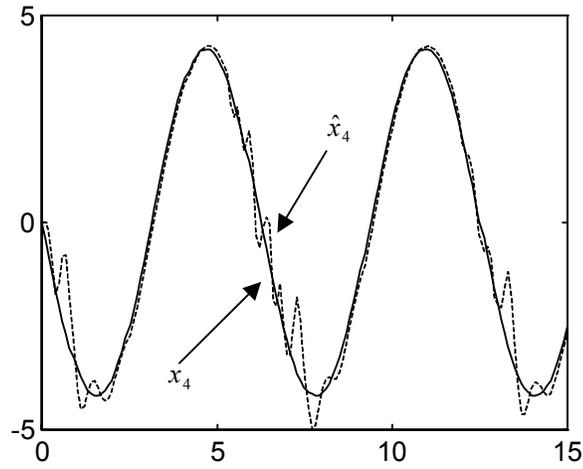


Figura 5-9: Identificación usando una RND multicapa de estado $\dot{\theta}$.

con $\bar{\eta} = 0.2$, $K_{1,2} := 560$, $\Lambda = [-1, -1, -1]$, $K_v = 0.1$, para identificar al sistema y luego obtener la acción de control. La simulaciones se llevaron acabo usando el paquete matlab y el simulink, el diagrama general se muestra en la figura B-5 Los resultados obtenidos de aplicar la acción de control se muestran en las figuras 5-11, 5-12, 5-13, 5-14, ademas la señal de control se muestra en la figura 5-10.

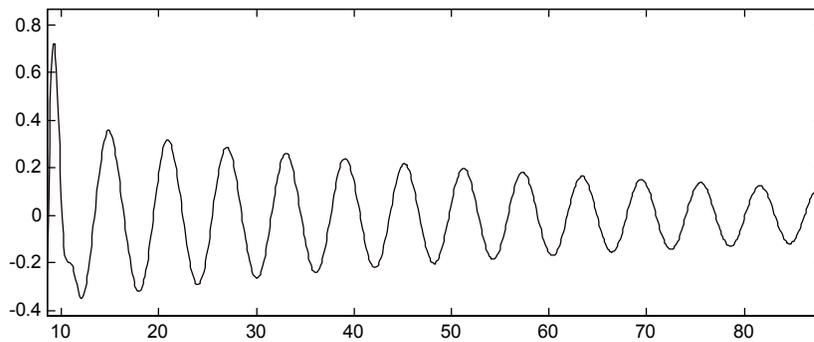


Figura 5-10: Comportamiento en tiempo de la ley de control (5.59).

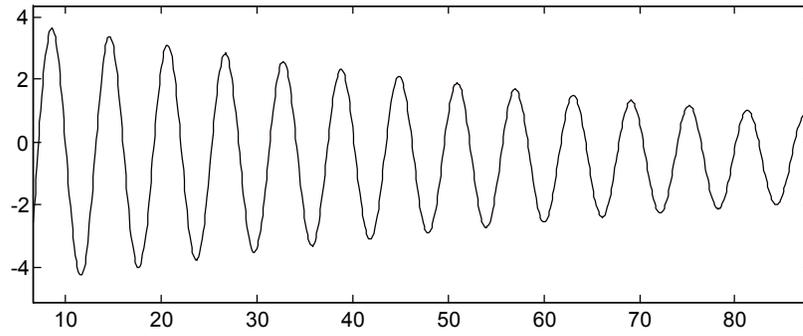


Figura 5-11: Control Indirecto para el estado q .

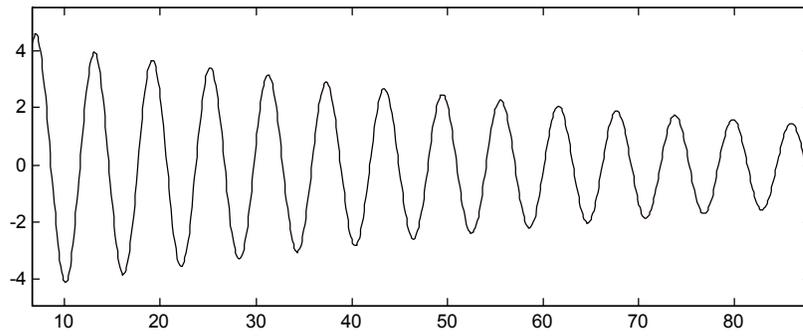


Figura 5-12: Control Indirecto para el estado \dot{q} .

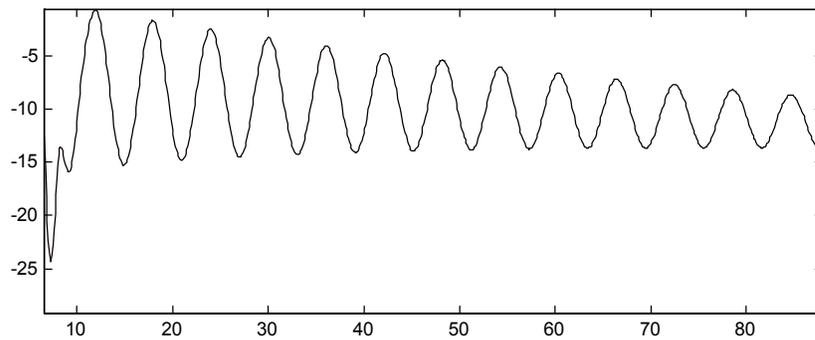


Figura 5-13: Control Indirecto para el estado θ .

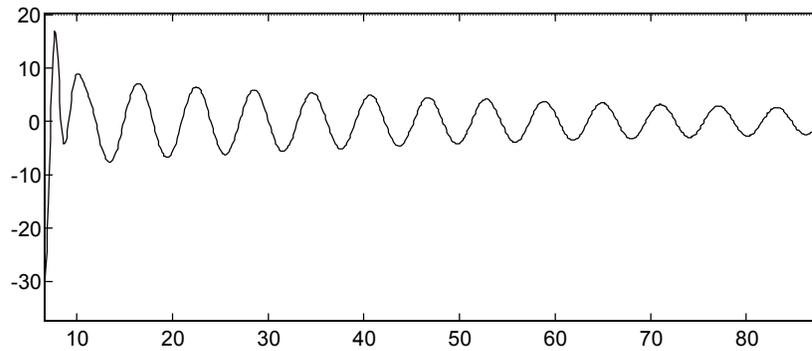


Figura 5-14: Control Indirecto para el estado $\dot{\theta}$.

Observación 5.4 Como se puede ver en las figuras anteriores, la acción de control obtenida a partir de la de una capa aplicando la actualización de los pesos, no es muy satisfactorio. Es importante mencionar que dado que, primero se realizar la identificación del sistema el error producido por esta, afecta en parte el desempeño de la acción de control.

Observación 5.5 El rango de ajuste de las ganancias $K_{1,2}$ estan entre $[450, 700]$ por lo que la sintonización no requirio de mucho tiempo hasta obtener el valor final de 560.

Capítulo 6

Conclusiones

En este reporte de tesis, se presentaron diferentes estructuras de redes neuronales dinámicas y su aplicación en la identificación y seguimiento de trayectorias para sistemas no-lineales.

En el **Capítulo 2**, se muestran las diferentes estructuras de redes neuronales, siendo la de mayor importancia para este trabajo las redes neuronales dinámicas para las cuales se presenta un nuevo teorema que muestra la aproximación entre el sistema no-lineal y la red neuronal dinámica, converge a un número menor a ε . Este teorema es el equivalente al presentado en 1988 por G. Cybenko (ver Apéndice teorema **AP1**, más información en [10]) para redes neuronales estáticas. Además, se mencionan las estructuras de control más conocidas.

En el **Capítulo 3**, se muestra el sistema no-lineal (TORA) y diferentes modelos del mismo, el cual es utilizado como ejemplo para aplicar los algoritmos obtenidos en los siguientes capítulos, este fue seleccionado dado que fue propuesto en [45], como un sistema estándar a partir del cual poder establecer comparaciones entre diferentes metodologías de control.

En el **Capítulo 4**, se enfoca en la identificación de sistemas no-lineales usando redes neuronales dinámicas. Las condiciones de estabilidad del error de identificación se determinan mediante el uso de un análisis tipo Lyapúnov en el cual mediante el uso

de una función definida positiva se puede mostrar la estabilidad, ver [18]; las reglas de aprendizaje (4.6) y (4.38) propuestas, la primera para una RND de una capa y la segunda para una RND multicapa, aseguran la convergencia del error de identificación a una región acotada. De esta forma, se presentan un análisis completo para diferentes arquitecturas de RND, con lo cual se cuenta con una herramienta completa cuando se desee utilizar las RND en el campo de la identificación, se tiene la facilidad de elegir que tipo de arquitectura se desea usar.

En el **Capítulo 5**, Primero se presenta un control adaptable directo para el cuál se realiza un análisis completo tipo Lyapunov mediante el cuál se puede llevar acabo el estudio de estabilidad ver [18], para el diseño de un control adaptatable directo, como resultado se obtiene la regla de aprendizaje (5.12) la cual es nueva y es usada por el neuro controlador que resuelve el problema de seguimiento, esta nueva regla es el resultado más importante de la tesis.

En la segunda parte, se realiza un análisis muy similar al del la primera parte, aunque dado que se busca implementar un control adaptable indirecto, la mayor parte del desarrollo se enfoca a la identificación del sistema no-lineal usando la regla de aprendizaje (5.36), con base a la cual se diseña el neuro controlador que resuelve el problema de seguimiento. Es importante mencionar que aunque se propone una red neuronal dinamica para realizar la identificación, en el caso particular del sistema no-lineal (TORA), se obtuvo un mejor resultado (comparar figs. 4-1 - 4-4 con 5-6 - 5-9) usando una RND de una capa, por lo que se eligio utilizar esta última RND para implementar la identificación y ley de control en las simulaciones.

Bibliografía

- [1] M.Agarwal, “A systematic classification of neural network based control”, *.IEEE Control Systems Magazine*, vol. 17, pp 75-93, 1997
- [2] F.Albertini and E.D.Sontag, “For neural networks function determines form”, *Neural Networks*, vol. 6, pp 975-990, 1993.
- [3] K.S.Narendra, and K.Parthasarathy, “Identification and control for dynamic systems using neural networks”, *IEEE Trans. on Neural Networks*, vol. 1, pp 4-27, 1990
- [4] D.S.Broomhead, and D.Lowe, “Multivariable functional approximation and adaptive networks”, *Complex Systems*, vol.2, pp 321-355, 1988.
- [5] R.T. Bupp, D.S. Bernstein and V.T. Coppola, “Experimental implementation of integrator backstepping and passive nonlinear controllers on the RTAC testbed,” *Int. J. Robust Nonlinear Control*, vol.8, pp. 435-457, 1998.
- [6] R.T. Bupp, D.S. Bernstein and V.T. Coppola, “A Benchmark problem for nonlinear control design,” *Int. J. Robust Nonlinear Control*, vol.8, pp. 307-310, 1998.
- [7] S. Chen, S.A.Billings,C.F.Cowan, and P.M.Grant, “Practical identification of NARMAX models using radial basis functions”, *Intl. Journal of Control*, vol. 52, pp 1327-1350, 1990.
- [8] E.A.Coddington and N.Levinson. *Theory of Ordinary Differential Equations*. Malabar, Fla: Krieger Publishing Company, New York, 1984.

- [9] M.J.Corless and G.Leitmann, Countinuous Stste Feedback Guaranteeing Uniform Ultimate Boundedness fo Uncertain Dynamic Systems, *IEEE Trans. Automat. Contr.* Vol.26, 1139-1144, 1981
- [10] G.Cybenko, "Approximation by Superposition of a Sigmoidal Function", *Mathematics of Control, Signals, and Systems*, vol. 2, pp 303-314, 1989.
- [11] G.C.Economu, M.Morari, and B.O.Palson, "Internal model control. 5.Extension to nonlinear systems", *Ind. Eng. Chem. Process De.Dev.*, vol. 25, pp 403-411, 1986.
- [12] M.M.Gupta and D.N.Rao, Editors, *Neuro-Control Systems, Theory and Applications*, IEEE Press, USA, 1994.
- [13] S.Haykin , *Neural Networks- A comprehensive Foundation*, Macmillan College Publ. Co., New York, 1994.
- [14] J.Hernandez and J-P.Barbot, Sliding Observer-based Feedback Control for Flexible Joints Manipulator, *Automatica*, Vol.32, No.9, 1243-1254, 1996
- [15] K.J.Hunt, D.Sbarbaro, R.Zbikowski and P.J.Gawthrop, "Neural Networks for Control Systems- A Survey", *Automatica*, vol. 28, pp. 1083-1112, 1992.
- [16] Z.Jiang, D.J. Hill and Y. Guo, "Stabilization and tracking via output feedback for the nonlinear benchmark system," *Automatica*, vol.34, no.7,pp. 907-915,1998.
- [17] Y.H. Kim and F.L. Lewis, *High-Level Feedback Control with Neural Networks*, World Scientific Publ. Co., 1998.
- [18] H.K.Khalil, *Nonlinear Systems* (Second Edition), Prentice Hall NJ 07458, 1996
- [19] E.B.Kosmatopoulos, M.M.Polycarpou, M.A.Christodoulou and P.A.Ioannpu, "High-Order Neural Network Structures for Identification of Dynamical Systems", *IEEE Trans. on Neural Networks*, Vol.6, No.2, 442-431, 1995.

- [20] E.B.Kosmatopoulos, M.A.Christodoulou and P.A.Ioannpu, "Dynamical Neural Networks that Ensure Exponential Identification Error Convergence", *Neural Networks*, Vol.10, No.2, 299-314, 1997
- [21] F.L.Lewis, K.Liu, and A.Yesilderek, "Neural net robot controller with guaranteed tracking performance", *IEEE Trans. on Neural Networks*, vol. 6, pp 703-715, 1995.
- [22] W.S.McCullock and W. Pitts, "A logical calculus of the ideas immanent in nervous activity", *Bulletin of Mathematical Biophysics*, vol. 5, 115-133.
- [23] R.Marino "High Gain Stabilization and Partial Feedback Linearization," *Proc. IEEE Conf. Decision and Control*, Athens, Greece, , pp. 209-213, December, 1986.
- [24] J.E.Moody and C.J.Darken, "Fast learning in networks of locally tuned processing units", *Neural Computations*, vol. 1, pp 281-294, 1989.
- [25] M.Morari and E.Zafiriou, *Robust Process Control*, Prentice Hall New Jersey, USA, 1989.
- [26] S.Nicosia and A.Tornambe, High-Gain Observers in the State and Parameter Estimation of Robots Having Elastic Joints, *System & Control Letter*, Vol.13, 331-337, 1989
- [27] L.Lung, *System Identification-Theory for the user*, Prentice Hall, New Jersey, USA, 1987.
- [28] T.Poggio and F.Girosi, "Networks for approximation and learning", *Proceedings of the IEEE*, vol. 78, pp 1481-1497, 1989.
- [29] B.T.Polyak, *Introduction to Optimization*, Optimization Software, Publication Division, New York, 1987.
- [30] M.J.D.Powell, "Radial basis functions for multivariable interpolation: a review", *IMA Conference on Algorithms for the Approximation of Functions and Data*, pp 143-167, Shrivvenham, U.K., 1985.

- [31] Poznyak, A.S and sanchez, E.N, “Nonlinear system approximation by neural networks error stability analysis”, *Int. J. Intell. Autom. Soft Comput*, vol.1, 247-258, 1995.
- [32] A.S. Poznyak, W.Yu , E.N. Sanchez and J.P. Perez, Stability Analysis of Dynamic Neural Control, *Expert System with Applications*, Vol.14, No.1, 227-236, 1998
- [33] S. Renals, “Radial basis function network for speech pattern classification”, *Electronics Letters*, vol. 25, pp 437-439, 1989.
- [34] F.Rosenblatt, “The perceptron: a probalistic model for information storage and organization in the brain”, *Psychological Review*, vol. 65, pp 386-408, 1958.
- [35] F.Rosenblatt, *Principles of Neurodynamics*, Spartan Books, Washington D.C., 1962.
- [36] G.A.Rovithakis and M.A.Christodoulou, “ Adaptive Control of Unknown Plants Using Dynamical Neural Networks“, *IEEE Trans. on Syst., Man and Cybern.*, Vol. 24, 400-412, 1994.
- [37] G.A.Rovithakis and M.A.Christodoulou, “Direct Adaptive Regulation of Unknown Nonlinear Dynamical System via Dynamical Neural Networks“, *IEEE Trans. on Syst., Man and Cybern.*, Vol. 25, 1578-1594, 1995.
- [38] D.G.Sbarbaro, “Connectionist feedforward networks for control of nonlinear systems”, *Ph. D. Thesis*, Faculty of Engineering, Glasgow University, 1992.
- [39] J.J.Slotine and R.M.Sanner, “Neural networks for adaptive control and recursive identification: a theoretical framework”, in *Essays on Control: Perspectives in the theory and its applications*, H.L.Tretelman and J.C.Willems, Editors, Birkhauser, Germany, 1993..
- [40] H.J.Sussmann, “Uniqueness of the weights for minimal feedforwrnets with a given input-output map”, *Neural Networks*, vol. 5, pp589-593, 1992.

- [41] J.A Suykens, J.P.L Vandewalle and B.L.R De Moor, *Artificial Neural Networks for Modelling and Control of Non-linear Systems*, Kluwer Academic Publishers, 1997.
- [42] S.Jagannathan and F.L. Lewis, "Identification of nonlinear dynamical systems using multilayered neural networks," *Automatica*, vol.32, no.12, pp. 1707-1712, 1996.
- [43] M. Jankovic, D. Fontaine and P.V.Kokotović, "TORA Example: Cascade-and passivity-Based Control Designs," *IEEE Transactions on control systems technology*, vol.4, no.3, pp.292-297, 1996
- [44] J.C.Willems, "Least squares optimal control and algebraic Riccati equations", *IEEE Trans. on Automatic Control*, Vol. 16, No 6, pp 621-634, 1971.
- [45] C.J Wan, D.S. Bernstein and V.T.Coppola, "Global stabilization of the oscillating eccentric rotor," *Proc. IEEE Conf. Decision and Control*, Orlando, FL,1994, 4024-4029. Also, *Nonlinear Dyn*, 10,49-62, 1996.
- [46] K.Wang, and A.N.Mitchel, "Robustness and perturbation analysis of a class of nonlinear systems with applications to neural networks", *IEEE Trans. on Circ. and Syst.*, Part 1, Vol. 41, No 1, pp 24-32, January 1994.
- [47] B.Widrow and S.D.Steans, *Adaptive Signal Processing*, Prentice-Hall, Englewood Cliffs, NJ, 1985
- [48] W. Yu, Alexander S.Poznyak, Indirect Adaptive Control via Parallel Dynamic Neural Networks, *IEE Proceedings - Control Theory and Applications*, Vol.37, No.1, 1999.
- [49] W. Yu and Alexander S.Poznyak, Robust Asymptotic Neuro Observer with Time Delay Term, submitted to CDC'99

Apéndice A

Matemático

Teorema A.1 (AP1) (Cybenko [10]) sea $\varphi(\cdot)$ una función creciente, monótona, acotada y estacionaria. Sea I_n denotado como un hipercubo n -dimensional. Sea $C(I_n)$ el espacio de funciones continuas en I_n . Entonces para cualquier $f \in C(I_n)$ y $\epsilon > 0$, Existe un entero m y unas constantes reales α_i, ρ_i y w_{ij} con $i = 1, \dots, m$ $j = 1, \dots, n$, tal que definiendo $F(u_1, u_2, \dots, u_n)$ como:

$$F(u_1, u_2, \dots, u_n) = \sum_{i=1}^m \alpha_i \varphi \left(\sum_{j=1}^n w_{ij} u_j - \rho_i \right)$$

es una realización aproximada de $f(\cdot)$; esto es:

$$|F(u_1, u_2, \dots, u_n) - f(u_1, u_2, \dots, u_n)| < \epsilon, \quad \forall (u_1, u_2, \dots, u_n) \in I_n.$$

Este teorema aplica directamente en un perceptrón multicapa, con las siguientes características:

1. Los nodos de entrada son: u_1, u_2, \dots, u_n .
2. La capa oculta de m neuronas que están completamente conectadas a la entrada.
3. La función de activación, para las neuronas ocultas, es una función monótona cre-

ciente, acotada y constante.

4. La salida de la red es una combinacion lineal de las neuronas de salida ocultas.

Lema A.1 *Para cualquier funcion vectorial diferenciable $g(x) \in \mathfrak{R}^n$, ($x \in \mathfrak{R}^n$) que ademas satisfaga la condicion de Lipschitz globalmente, i.e. Existe una constante positiva L_g tal que*

$$\|g(x_1) - g(x_2)\| \leq L_g \|x_1 - x_2\|, \forall x_1, x_2 \in \mathfrak{R}^n \quad (\text{A.1})$$

para cualquier $x_1, x_2 \in \mathfrak{R}^n$, la siguiente propiedad se cumple para cualquier $x, \Delta x \in \mathfrak{R}^n$:

$$g(x + \Delta x) = g(x) + \nabla^T g(x) \Delta x + v_g \quad (\text{A.2})$$

donde el vector v_g cumple con: $\|v_g\| \leq 2L_g \|\Delta x\|$.

Lema A.2 *Si se define una función positiva $V(x)$, $x \in \mathfrak{R}^n$ como*

$$V(x) := [\|x - x^*\| - \mu]_+^2$$

donde $[\cdot]_+^2 := ([\cdot]_+)^2$, $[\cdot]_+$ es definido como:

$$[z]_+ = \begin{cases} z & z \geq 0 \\ 0 & z < 0 \end{cases}$$

entonces, la función $V(x)$ es diferenciable y su gradiente es:

$$\nabla V(x) = 2 [\|x - x^*\| - \mu]_+ \frac{x - x^*}{\|x - x^*\|}$$

con la constante de Lipschitz igual a 1.

Apéndice B

Diagramas de Simulink

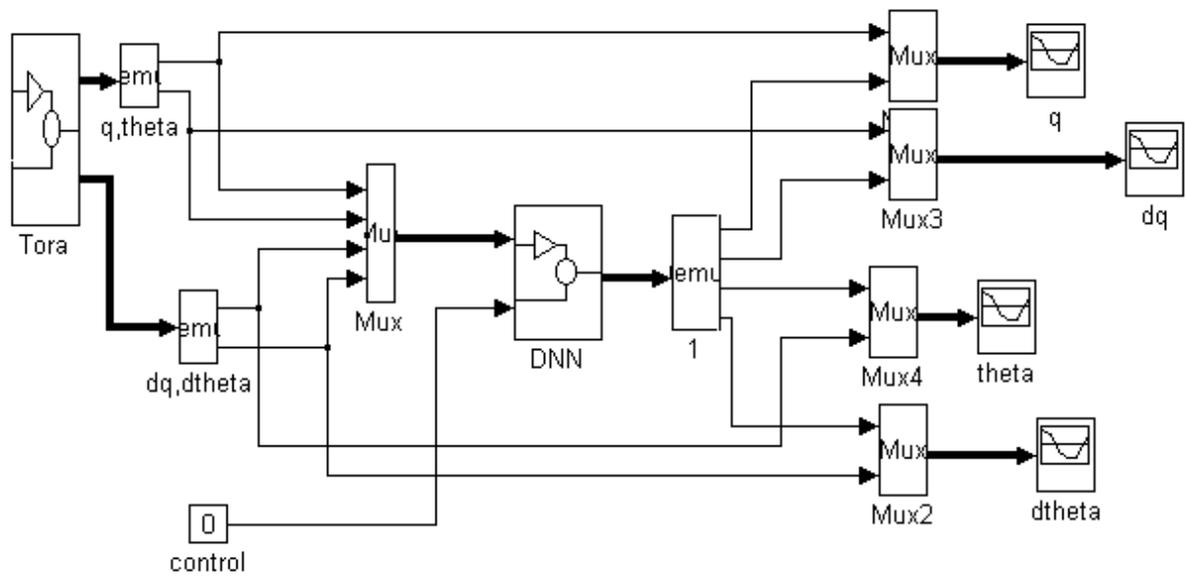


Figura B-1: Diagrama Principal de la RND de una capa.

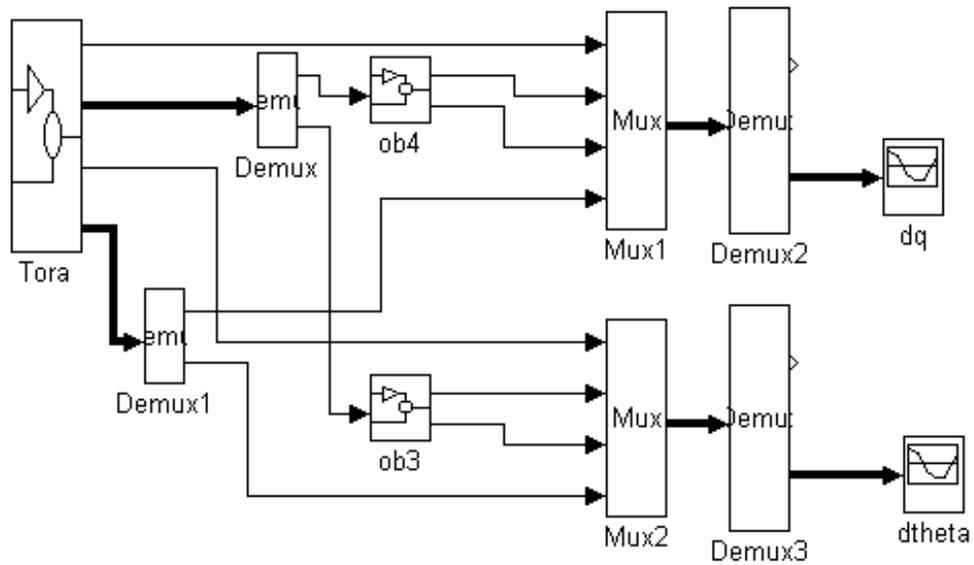


Figura B-2: Diagrama Principal de los observadores usados para estimar los estados.

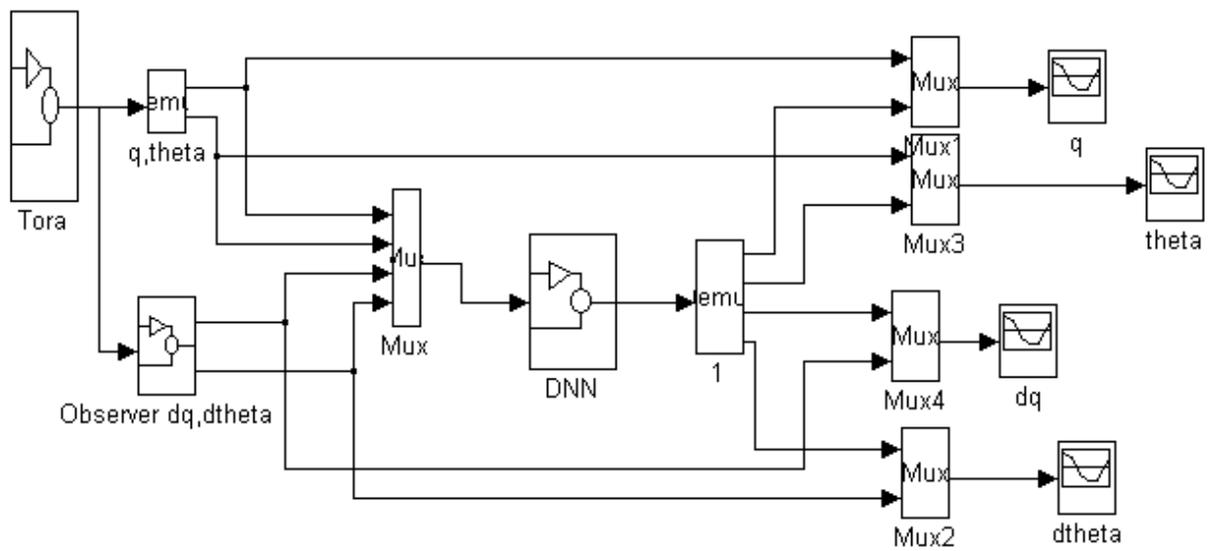


Figura B-3: Diagrama Principal de Identificación usando la estimación de 2 estados.

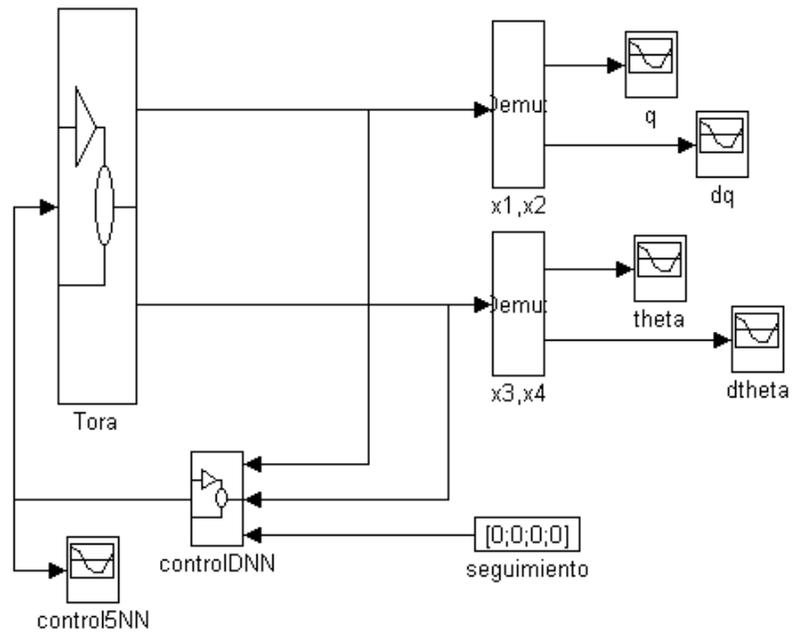


Figura B-4: Diagrama Principal del Control Directo.

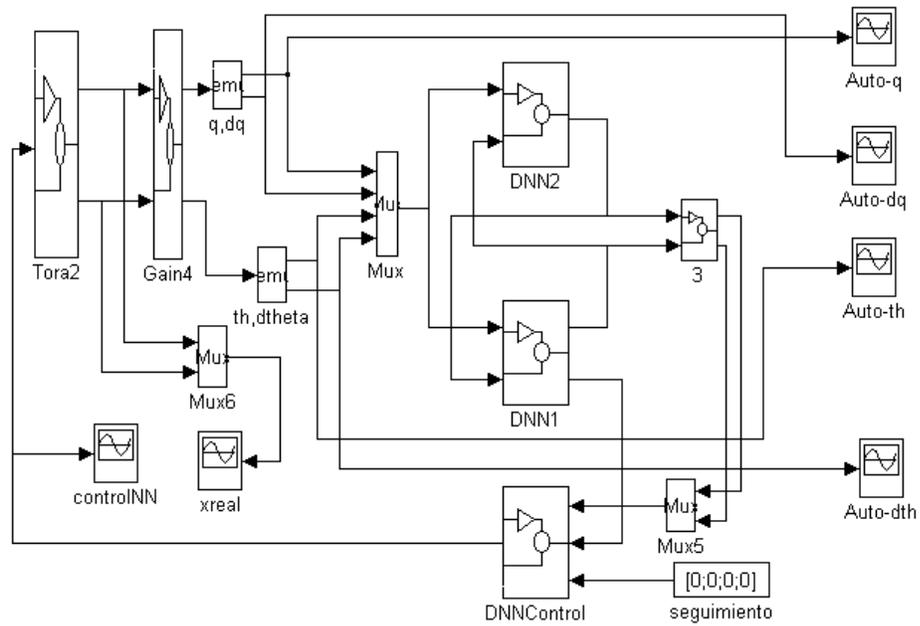


Figura B-5: Diagrama Principal del Control Indirecto.