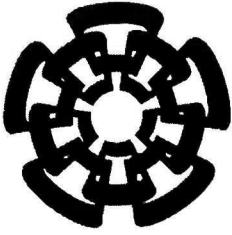


xx(178589.1)



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

Control Robusto Discreto de Robots Manipuladores con Redes Neuronales



CENTRO DE INVESTIGACIÓN Y
DE ESTUDIOS AVANZADOS DEL
INSTITUTO POLITÉCNICO
NACIONAL

COORDINACIÓN GENERAL DE
SERVICIOS BIBLIOGRÁFICOS

Tesis que presenta:

Fernando Ornelas Tellez

para obtener el grado de:

Maestro en Ciencias

en la especialidad de:

Ingeniería Eléctrica

Directores de Tesis

Dr. Eduardo José Bayro Corrochano
Dr. Alexander Georgievich Loukianov

CINVESTAV
IPN
ADQUISICION
DE LIBROS

Guadalajara, Jalisco, Julio de 2008.

CLASIF.: TRIG. 28 074 2008
ADQUIS.: 581-532
FECHA: 23-11-2009
PROCED.: Don. 2009
◆

ID: 158259-1001

Control Robusto Discreto de Robots Manipuladores con Redes Neuronales

**Tesis de Maestría en Ciencias
Ingeniería Eléctrica**

Por:

Fernando Ornelas Tellez

Ingeniero Electrónico

Instituto Tecnológico de Morelia 2001-2005

Becario de CONACYT, expediente no. 203118

Directores de Tesis

Dr. Eduardo José Bayro Corrochano

Dr. Alexander Georgievich Loukianov

CINVESTAV del IPN Unidad Guadalajara, Julio de 2008.

Agradecimientos

A mis padres, Gaspar Ornelas y María de los Reyes Tellez, que por su cariño, comprensión y ejemplo de vida, me han dado motivación, valor y fuerza para la realización de mis estudios.

A mi esposa Yenhi y mi hijo Alex, por su amor y paciencia durante todo el tiempo que físicamente estuve ausente de su lado, pero que mi mente, corazón y amor siempre estuvieron con ellos.

A mis hermanas Mary, Ale, Tere y Jime, ya que su amor y confianza me han inspirado para realizar este trabajo de la mejor manera posible. Reconozco de forma muy especial a mi tío Cosme por su apoyo moral en todos mis proyectos, tanto personales como profesionales.

Al Cinvestav Unidad Guadalajara por la oportunidad y facilidades brindadas para la realización de mis estudios de maestría plasmados en este trabajo, así como al Consejo Nacional de Ciencia y Tecnología (CONACYT) por el apoyo económico otorgado.

A mis directores de tesis, con mucho respeto y admiración, al Dr. Eduardo José Bayro Corrochano y al Dr. Alexander Georgievich Loukianov por confiarme la responsabilidad de este trabajo y por la paciencia brindada en el desarrollo de la misma, ya que sin su esfuerzo no hubiese sido posible la culminación de esta tesis.

Al Dr. Edgar Sanchez Camperos, quien asesoró también de manera muy cercana el desarrollo de este trabajo y que basándose en su conocimiento y experiencia, se logró progresar exitosamente.

Al grupo de GeoVisLab: Ruben Machucho, Julio Zamora y Jorge Rivera, por las facilidades y apoyos prestados para la realización de este trabajo. A mis amigos y compañeros de clase: Antonio F. “el Gato” José María, Abelardo H., David G., al Mike, Sergio A., “la Chela”, Canche, Victor, Diana, Fernando G., Alvaro, Pedro Esquivel y Josue Rayas, por su amistad y apoyo durante estos dos años de maestría.

Resumen

Analizar sistemas no lineales de gran escala requiere de mucho esfuerzo ya que, los parámetros que forman parte del sistema (modelo) son difíciles de obtener en muchos casos; por eso, obtener un sistema de control, en el cual se tienen incertidumbres paramétricas, no es práctico. Más aún, existen sistemas en los cuales el comportamiento dinámico es modificado de acuerdo a perturbaciones externas, variaciones paramétricas y dinámicas no modeladas. Las complicaciones anteriores hacen que el diseño del control sea más complicado. Un caso típico de estos sistemas, son los robots.

Se hace un análisis de forma general sobre los modelos de robots; estos modelos son tratados como una clase de sistemas los cuales tienen la forma no lineal controlable por bloques y además, las interacciones entre los diferentes eslabones, son vistos como términos de perturbación. Un análisis similar es hecho a la red neuronal que se propone. De esta forma, se tiene por objetivo el diseñar un controlador basado en redes neuronales recurrentes de alto orden para controlar un robot planar de dos grados de libertad. El mismo procedimiento aplica para robots de cualquier número de grados de libertad. Se utilizan redes neuronales recurrentes para identificar el sistema debido a que éstas son capaces de modelar sistemas no lineales.

En este trabajo, el sistema (robot) es visto como un conjunto de subsistemas interconectados en el cual, cada subsistema corresponde a un componente del robot (eslabón). Se usa sólo información local (posición y velocidad de cada eslabón) para diseñar un controlador por eslabón que sea robusto ante perturbaciones internas y externas; a esto se le llama control descentralizado. La identificación y control descentralizado tienen un diseño tal que se garantiza la estabilidad del sistema completo.

La parte principal de las redes neuronales está en la forma en la que éstas aprenden. En este sentido, se implementa el filtro de Kalman para realizar la identificación en línea.

El diseño del controlador propuesto está basado en el modelo neuronal en tiempo discreto y utiliza la técnica de control por bloques en tiempo discreto en combinación con modos deslizantes. Los algoritmos de identificación y control son probados, tanto en simulación como en tiempo real, en un robot planar de dos grados de libertad.

El proceso de identificación y control son realizados al mismo tiempo.

Abstract

Analysis of large-scale nonlinear system requires of a lot of effort, since, the real model parameters some times are so difficult to obtain. So, considering the before mentioned, to get a controller based on the plant model which have uncertainties is not practical. Moreover, there exist systems whose dynamical behavior changes according to unknown external disturbances or due to parameter variations and unmodelled dynamics, which make the controller design more complicated, as typical case we have the robots.

We analyze, in a general way, the robot's model as a class of systems that are constituted of the nonlinear block controllable form with disturbance term, the same analysis is made for the neural network. In robots, we deal with systems which have interacting components, so, we consider them in this thesis how disturbance terms. We have for target to design a controller based on recurrent high order neural networks for a two degree of freedom planar robot. To identify, we use recurrent neural networks since these identifier is capable of modeling uncertain nonlinear systems.

In this dissertation, the system (robot) can be seen as a set of interconnected subsystems corresponding to each one components of the robot (links). Using only local information (position and velocity in this case) of each link we can design a robust controller; this is called decentralized control. The identification and control must be done guaranteeing stability for the whole system.

An important part in neural networks is a training algorithm; in this sense, we use an extended Kalman filter to get the online identification.

The proposed controller is based on the designed discrete-time recurrent high order neural networks model and discrete-time block control combined with sliding modes. The proposed neural identifier and control applicability is illustrated by trajectory tracking as much in simulation as in real time for a two degree of freedom planar robot.

The performance of the online identification and control are realized at the same time.

Prefacio

Con el paso del tiempo se han venido perfeccionando técnicas avanzadas para la solución de numerosos problemas reales donde los métodos tradicionales han sido insuficientes o inapropiados en la resolución del problema, ya sea porque estos son tediosos analíticamente o porque simplemente no se cuenta con los medios e información necesaria para su análisis y desarrollo.

De lo anterior, es donde ciertas técnicas brindan sus bondades de análisis y diseño, permitiendo tener un conocimiento del comportamiento de algún proceso para posteriormente manipularlo, de ser posible, en algún beneficio particular. Una de las técnicas que han tenido avances significativos basándose en métodos heurísticos es el uso de Redes Neuronales, las cuales han tenido diversos campos de aplicación, que entre ellos destaca, el modelado y control de robots manipuladores.

Conteniendo seis capítulos, esta tesis está organizada de la siguiente manera:

- **Capítulo 1.** Presenta el planteamiento del problema, así como el motivo de la propuesta de solución. También se abordan antecedentes relacionados con el planteamiento del problema y algunas de las soluciones existentes.
- **Capítulo 2.** Se presentan las descripciones matemáticas generales sobre modelado y análisis de los robots manipuladores; un análisis matemático de la estructura de la red neuronal propuesta y su entrenamiento basado en el filtro de Kalman.

Capítulo 3. Se propone un algoritmo de control basado en la técnica de control por bloques en tiempo discreto en combinación con modos deslizantes.

Capítulo 4. Presenta la aplicación de los algoritmos de identificación control a un robot de dos grados de libertad.

Capítulo 5. Presenta la descripción física del robot de dos grados de libertad e implementación en tiempo real de los algoritmos de control propuestos.

Capítulo 6. Presentación de conclusiones y trabajo futuro.

Índice general

1. Introducción	1
1.1. Motivación .	2
1.2. Antecedentes	2
1.3. Objetivos	4
2. Modelado de robots e identificación	5
2.1. Modelado de robots manipuladores	5
2.1.1. Distribución de masa	6
2.1.2. Formulación de Lagrange para modelado de manipuladores	7
2.1.3. Una clase de sistemas descentralizados	13
2.2. Identificación con redes neuronales	14
2.2.1. Modelo de las redes neuronales recurrentes discretas	15
2.2.2. Redes neuronales descentralizadas	16
2.2.3. Ley de aprendizaje en línea	17
2.3. Entrenamiento con el filtro de Kalman	19
2.3.1. El filtro de Kalman	19
2.3.2. El Filtro de Kalman Extendido	21
2.3.3. Entrenamiento de la red neuronal	21
3. Control por bloques en tiempo discreto	25
3.1. Control por bloques con modos deslizantes	25
3.1.1. Transformación de linealización	26
3.1.2. Ley de control	28

3.1.3. Análisis de estabilidad	28
4. Aplicación a un robot de dos DOF	31
4.1. Modelos neuronales descentralizados	31
4.1.1. Modelo neuronal 1	33
4.1.2. Modelo neuronal 2	33
4.1.3. Modelo neuronal 3	34
4.2. Evaluación experimental	36
4.2.1. Resultados de simulación	37
5. Implementación en tiempo real	47
5.1. Cómputo, interfaz y adquisición de datos	47
5.2. Fuente de poder y etapa de potencia	49
5.3. Actuadores, codificadores y planta (Robot)	50
5.4. Instrucciones de uso	51
5.5. Evaluación experimental	53
5.5.1. Resultados en tiempo real	54
5.5.2. Algoritmo para la planeación de trayectorias	62
6. Conclusiones y trabajo futuro	67
6.1. Conclusiones .	67
6.2. Trabajo futuro	68
A. Sintonización de los parámetros para el EKF	69
B. Generación de trayectorias	71
C. Cinemática	75
Bibliografía	79

Índice de figuras

2.1. <i>Distribución de masa de un cuerpo rígido.</i>	6
2.2. <i>Brazo manipulador de dos grados de libertad.</i>	8
2.3. <i>Estructura de una red neuronal discreta de alto orden.</i>	19
2.4. <i>Sistema dinámico lineal en tiempo discreto.</i>	20
4.1. <i>Esquema de identificación descentralizada.</i>	32
4.2. <i>Esquema del modelo neuronal 1.</i>	33
4.3. <i>Esquema del modelo neuronal 2.</i>	34
4.4. <i>Esquema del modelo neuronal 3.</i>	35
4.5. <i>Esquema de identificación y control.</i>	37
4.6. <i>Seguimiento de trayectoria de la posición (MN1).</i>	39
4.7. <i>Error de posición y velocidad (MN1).</i>	39
4.8. <i>Variedades deslizantes (MN1).</i>	40
4.9. <i>Entradas de control (MN1).</i>	40
4.10. <i>Seguimiento de trayectoria de la posición (MN2).</i>	41
4.11. <i>Error de posición y velocidad (MN2).</i>	42
4.12. <i>Variedades deslizantes (MN2).</i>	42
4.13. <i>Entradas de control (MN2).</i>	43
4.14. <i>Seguimiento de trayectoria de la posición (MN3).</i>	44
4.15. <i>Error de posición y velocidad (MN3).</i>	45
4.16. <i>Variedades deslizantes (MN3).</i>	45
4.17. <i>Entradas de control (MN3).</i>	46
5.1. <i>Sistema físico completo del robot manipulador.</i>	48

5.2. <i>Ubicación de los componentes del robot manipulador</i>	48
5.3. <i>Tarjeta de adquisición de datos.</i>	49
5.4. <i>Diagrama a bloques de la tarjeta de adquisición de datos.</i>	50
5.5. <i>Etapas de potencia para los motores del brazo manipulador.</i>	51
5.6. <i>Motor 1 que controla el eslabón 1.</i>	52
5.7. <i>Motor 2 que controla el eslabón 2.</i>	52
5.8. <i>Flujo de información.</i>	53
5.9. <i>Robot planar de dos DOF.</i>	54
5.10. <i>Seguimiento de trayectoria de la posición en tiempo real (MN1).</i>	55
5.11. <i>Error de posición y velocidad en tiempo real (MN1).</i>	55
5.12. <i>Variedades deslizantes en tiempo real (MN1).</i>	56
5.13. <i>Entradas de control en tiempo real (MN1).</i>	57
5.14. <i>Seguimiento de trayectoria de la posición en tiempo real (MN2).</i>	57
5.15. <i>Error de posición y velocidad en tiempo real (MN2).</i>	58
5.16. <i>Variedades deslizantes en tiempo real (MN2).</i>	58
5.17. <i>Entradas de control en tiempo real (MN2).</i>	59
5.18. <i>Seguimiento de trayectoria de la posición en tiempo real (MN3).</i>	60
5.19. <i>Error de posición y velocidad en tiempo real (MN3).</i>	60
5.20. <i>Variedades deslizantes en tiempo real (MN3).</i>	61
5.21. <i>Entradas de control en tiempo real (MN3).</i>	61
5.22. <i>Seguimiento de trayectoria de la posición en tiempo real (MN3).</i>	63
5.23. <i>Diagrama a bloques para la generación de trayectoria.</i>	64
5.24. <i>Posición de la referencia y del efector final del robot.</i>	65
5.25. <i>Evolución de las coordenadas.</i>	66
5.26. <i>Referencias angulares para realizar la trayectoria de un triángulo.</i>	66
B.1. <i>Formas de trayectorias.</i>	72

Capítulo 1

Introducción

El hombre como ser racional siempre ha buscado la realización de sus actividades de una forma que implique el menor esfuerzo, el menor riesgo posible, menor aburrimiento, entre otras. El medio para conseguirlo era que hubiese *alguien* o *algo* que lo realizara. En este sentido, esta tesis se enfocará en el *algo* que realice dichas actividades, en particular los robots, entre los cuales se encuentran los manipuladores reprogramables de uso general con sensores externos que pueden efectuar diferentes tareas de montaje, transporte, etc. Con esto, un robot debe poseer inteligencia, la cual se debe normalmente a los algoritmos de computador asociados con su sistema de control y sensorial [5].

Para la parte de inteligencia (control), se propone un esquema basado en un modelo artificial haciendo uso de redes neuronales recurrentes. Por red neuronal recurrente nos referimos al hecho de retroalimentar la salida de alguna neurona ya sea con signo misma o con neuronas aledañas en la red [14] y que tiene la capacidad de modelar sistemas no lineales, que en ocasiones tienen incertidumbres paramétricas [15]. Esquemas de redes neuronales se pueden encontrar en [8], [17] y [18].

El exitoso funcionamiento de la red neuronal está basado principalmente en el proceso de aprendizaje; de éste se consigue tener una identificación del proceso o planta de la mejor forma posible. Ahora la identificación puede ser tratada como un problema de filtrado de Kalman [3], además de que éste se puede implementar fácilmente utilizando pocas herramientas conceptuales. Otra característica importante del filtro de Kalman es que puede ser implementado en sistemas con múltiples entradas-múltiples salidas (MIMO, por sus siglas en inglés de *multi-input*, *multi-output*).

Primeramente se analiza de forma general la obtención del modelado de robots manipuladores de N grados de libertad, los cuales, como se verá, tendrán una forma no lineal controlable por bloques; por conveniencia se diseñará una red neuronal con una estructura similar al sistema que identifica.

Dada la importancia de tener algoritmos de control eficientes, en esta tesis los controlado-

res propuestos están basados en el modelo artificial neuronal que describe el comportamiento dinámico del robot. Este controlador usará la técnica de control por bloques en tiempo discreto en combinación con modos deslizantes

Básicamente existen dos caminos para diseñar un controlador en tiempo discreto: diseñar un controlador en tiempo continuo y posteriormente discretizarlo por alguna técnica de discretización, mientras que el segundo es hacer un análisis y diseño del controlador en tiempo discreto. En ambos se debe considerar el periodo de muestreo. Se aclara que de implementar, en un computador digital, una ley de control que está diseñada en tiempo continuo, puede no cumplir con el objetivo de control o incluso volver la planta inestable.

1.1. Motivación

Las técnicas de control apoyadas en los modelos dinámicos no lineales de los robots son abundantes; el problema principal con este tipo de controladores es que no son robustos ante perturbaciones internas y externas en la planta. De lo anterior surge que, el análisis y diseño de los controladores no es regularmente lo más complicado; en realidad, el trabajo arduo está en la obtención de los parámetros del robot (algunos de ellos incluso variables por naturaleza), ya que no se cuenta con los instrumentos necesarios para realizar dichas mediciones de parámetros. Además, el modelo matemático resulta tedioso, complicado e inclusive inoperable cuando se incrementan los grados de libertad, pues las ecuaciones que describen su dinámica son demasiado extensas; es por ello que, surge la necesidad de un planteamiento matemático más robusto y eficaz al operar; un modelo (identificador) que describa el comportamiento preciso del robot y que pueda ser utilizado para determinar su control. Las redes neuronales son una alternativa idónea en la realización de esa tarea como se mostrará en este trabajo.

En esta tesis, el sistema (robot) es visto como un conjunto de subsistemas interconectados en el cual, cada subsistema corresponde a un componente del robot (eslabón). Se usa sólo información local (posición y velocidad de cada eslabón) para diseñar un controlador por eslabón que sea robusto ante perturbaciones internas y externas; a esto se le llama control descentralizado, en éste se facilita el diseño de la identificación y control.

1.2. Antecedentes

La aparición de la robótica surge para facilitar al hombre ciertas actividades que en un inicio eran simples, repetitivas y que no requerían de gran precisión; actividades como transferir materiales, pintura industrial, entre otras, son económicamente viables para automatización [4]. Sin embargo, las actividades que se pueden realizar con la robótica han ido

en aumento y en las cuales se precisa de mayor movilidad, destreza y autonomía; por ello, en aplicaciones no industriales como medicina, espaciales, construcción, etc., las leyes de control que rigen el comportamiento de los robots deben mejorar.

Entre los controles aplicados a la robótica podemos mencionar:

- **Estabilización por retroalimentación de estado.** En este control se pretende estabilizar el robot en un punto de equilibrio (*regulación*), más aún, estabilizar sobre la trayectoria de una referencia [20].
- **Línealización por retroalimentación.** Es transformar un sistema no lineal a un sistema lineal controlable, en el cual se puedan aplicar las técnicas utilizadas para el control de sistemas lineales. Dependiendo del sistema, este cambio de coordenadas no siempre es posible [13].
- **Retroalimentación de la salida.** La hipótesis de que todos los estados del sistema están disponibles para la retroalimentación no siempre es posible. La estabilización global de un sistema no lineal por medio de la retroalimentación de la salida, es en general, un problema difícil. Sin embargo, existen sistemas para los cuales la estabilidad global es alcanzada por medio de un control basado en observadores [13], los cuales para esta tarea utilizan las salidas disponibles.

Control Adaptable. Es utilizado en sistemas lineales y no lineales, invariantes o no; que además, tienen parámetros desconocidos, constantes o que cambian lentamente tal que el control pueda adaptarse. Estos parámetros pueden ser ajustados en línea y utiliza la entrada y la salida medida del sistema [7].

- **PID.** Este control pretende hacer al sistema asintóticamente estable aun en presencia de perturbaciones. Las ventajas de este control es que tiene un buen desempeño y es simple de operar; también son efectivos cuando no se tiene conocimiento exacto de parámetros como inercia o masa del robot (en si, un conocimiento exacto de los parámetros de la planta). El problema del PID es la sintonización adecuada para los valores de las tres constantes de acción: proporcional, integral y derivativa. Un ejemplo de este control en robots manipuladores está en el PUMA 560. Una de las desventajas principales es que sus ganancias usadas para la retroalimentación son constantes y preespecificadas. Este control no tiene la capacidad de modificar las ganancias de retroalimentación bajo condiciones de carga variable, lo cual es un caso típico en un robot industrial real, resultando así en un control deficiente ante cargas variables en el robot.

Control Inteligente. Se tienen dos propuestas principales para este control: *lógica difusa* y *redes neuronales*. La propuesta en esta tesis es el uso de las redes neuronales.

Como antecedente de esta tesis, se tiene que en [16] se hace un análisis de redes neuronales descentralizadas en tiempo continuo. El análisis que se desarrolla en este trabajo de tesis es el uso de redes neuronales en tiempo discreto y con un enfoque descentralizado.

1.3. Objetivos

Encontrar o mejorar la solución a las necesidades del individuo y sociedad siempre serán los objetivos de la comunidad científica y tecnológica. En esta tesis se contribuirá con:

- Dar, de forma general, las bases para la obtención del modelo de un robot manipulador de N grados de libertad. A modo de ejemplo se obtiene el modelo de un robot manipulador de dos grados de libertad en tiempo continuo, que posteriormente es *discretizado*; de éste último se pueden obtener estrategias y consideraciones para el análisis y diseño de la red neuronal en lo que respecta a la estructura de la red y para pruebas de simulación. En este sentido, se diseñará la red neuronal tomando como base la estructura del modelo discretizado de un robot de dos grados de libertad (DOF), por sus siglas en inglés *Degree Of Freedom*. El algoritmo de identificación - control se puede extender a N grados de libertad.
- Establecer criterios para la sintonización del filtro de Kalman, el cual es empleado como algoritmo de entrenamiento para la red neuronal.
- Establecer un esquema de red neuronal descentralizada, es decir, un esquema en el cual haya un identificador - controlador por cada eslabón, y que además, sea independiente del funcionamiento de los otros eslabones en el robot (manipulador de dos DOF en nuestro caso). El controlador usará solamente información local de su respectivo eslabón o grado de libertad (posición y velocidad).
- Que este modelo neuronal permita identificar y controlar al robot (sobre una trayectoria deseada) por medio de la técnica de control por bloques y modos deslizantes. Se diseñarán varias estructuras de redes neuronales cuya diferencia vendrá dada por el número de interconexiones en la red neuronal, así como las variables involucradas en las interconexiones.
- Establecer un algoritmo para la planeación de trayectorias apoyado en el Jacobiano, la cinemática directa e inversa de un robot manipulador de dos DOF.

Implementación en tiempo real de los algoritmos propuestos anteriormente.

Capítulo 2

Modelado de robots e identificación

En este capítulo se hace una descripción breve sobre la obtención del modelo de robots manipuladores. Esta obtención se caracteriza por generar un conjunto de ecuaciones diferenciales y algebraicas, normalmente no lineales, que se obtienen a partir de un estudio analítico del sistema; para ello, se hace uso de leyes de comportamiento físico (leyes de conservación, equilibrio de fuerzas, etc.). También se aborda la identificación de sistemas, la cual se caracteriza por considerar el sistema como una caja negra o gris, y en donde las medidas experimentales son usadas para deducir las relaciones entrada-salida o encontrar modelos matemáticos que simulen el comportamiento del sistema físico; este último es el que se aborda y resuelve en esta tesis.

2.1. Modelado de robots manipuladores

El estudio de los robots conlleva el analizar, entre otras cosas, consideraciones como cinemática directa, cinemática inversa y fuerzas que actúan sobre el robot para causar su movimiento. En esta sección se incluye una descripción resumida de las ecuaciones de movimiento para un manipulador así como su deducción, para mayor detalle del análisis de movimiento de robots ver [4] y [5]. Se aplicarán los conceptos necesarios para obtener el modelo de un robot manipulador rígido de N grados de libertad [4], en particular se desarrolla el modelo para $N = 2$ (robot de dos grados de libertad), el cual servirá de apoyo para probar los algoritmos de control propuestos.

Los manipuladores están compuestos por elementos físicos en cadena (eslabones) los cuales tienen masa, forma, dimensiones, etc.; en el modelado, estos aparecerán afectando el comportamiento dinámico del robot, por lo que se hará una breve descripción de ellos.

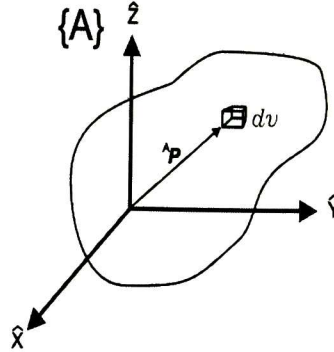


Figura 2.1: Distribución de masa de un cuerpo rígido.

2.1.1. Distribución de masa

Cuando se analiza el movimiento rotacional alrededor de un eje, la noción de momento de inercia es familiar. Un cuerpo rígido en tercera dimensión tiene un número infinito de ejes sobre el cual éste puede girar; si nos enfocamos en cualesquiera de ellos se requerirá la caracterización de la distribución de masa con respecto a ese eje. Aquí se introduce el **tensor de inercia**, el cual da información de la distribución de masa de un cuerpo rígido relativo a un marco de referencia. La Figura 2.1 muestra un cuerpo rígido con su respectivo marco de referencia.

El tensor de inercia relativo al marco \mathbf{A} está expresado en la matriz

$${}^A I = \begin{pmatrix} I_{xx} & -I_{xy} & -I_{xz} \\ -I_{xy} & I_{yy} & -I_{yz} \\ -I_{xz} & -I_{yz} & I_{zz} \end{pmatrix} \quad (2.1)$$

donde los elementos escalares vienen dados por:

$$\begin{aligned}
 I_{xx} &= \int_V (y^2 + z^2) \rho dv \\
 I_{yy} &= \int_V (x^2 + z^2) \rho dv \\
 I_{zz} &= \int_V (x^2 + y^2) \rho dv \\
 I_{xy} &= \int_V xy \rho dv \\
 I_{xz} &= \int_V xz \rho dv \\
 I_{yz} &= \int_V yz \rho dv.
 \end{aligned}$$

Aquí, el cuerpo rígido está compuesto de elementos diferenciales de volumen dv y ρ describiendo la densidad del material. Cada elemento de volumen contiene un vector asociado $^A I$. Los elementos I_{xx} , I_{yy} y I_{zz} son llamados *momentos de inercia de masa* [4], mientras que los términos con índices cruzados *productos de inercia*. Un punto importante es que si se tiene la libertad de elegir la orientación del marco de referencia, es posible hacer los productos cruzados cero [4]. Este análisis de distribución de masa es importante al deducir la ecuación de energía cinética asociada a los cuerpos del robot (eslabones) como se verá más adelante.

2.1.2. Formulación de Lagrange para modelado de manipuladores

Entre las formulaciones convencionales para el modelado de la dinámica de los robots manipuladores están: la **aproximación de Newton-Euler** y la **formulación dinámica de Lagrange**. La primera analiza la dinámica sobre un “balance de fuerzas”, mientras que la segunda esta “basada en energía”. El resultado obtenido por ambos métodos es el mismo. A modo de ejemplo, la formulación con la cual se desarrollo el modelo del manipulador de dos DOF es el de Lagrange y en base a este modelo se desarrollaron las técnicas de control propuestas más adelante.

Antes de iniciar el desarrollo de la dinámica del robot, es necesario conocer algunos parámetros físicos y geométricos del robot; además, se requiere conocer las relaciones de transformación (rotación y traslación) entre los eslabones del robot. Las transformaciones pueden ser encontradas una vez determinados los parámetros de DENAVIT-HARTENBERG (DH) [4], [5]. Por ejemplo, para el manipulador rígido de dos DOF mostrado en la Figura 2.2, los parámetros DH son dados en la Tabla 2.1. Para este robot (dos DOF) se tiene que $i = 1, 2, 3$ y donde $m_i(kg)$, $l_i(m)$, $l_{ci}(m)$ y g son: masa del eslabón i , longitud del eslabón i ,

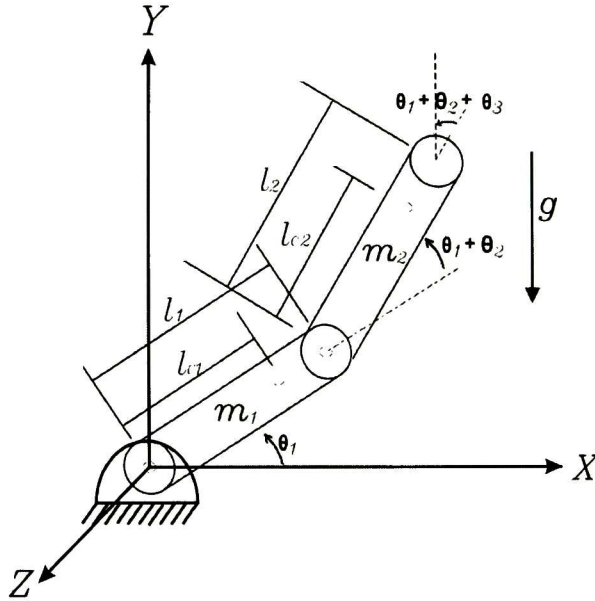


Figura 2.2: Brazo manipulador de dos grados de libertad.

Tabla 2.1: Parámetros DH del robot de dos DOF

i	α_{i-1}	a_{i-1}	d_i	θ_i
1	0	0	0	θ_1
2	0	l_1	0	θ_2
3	0	l_2	0	θ_3

longitud al centro de masa del eslabón i y la gravedad ($g = 9,81m/s^2$), respectivamente.

De los parámetros mencionados con anterioridad se pueden deducir las relaciones de transformación entre el marco de referencia del eslabón $i - 1$ con el i . La transformación ${}^i{}_{i-1}T$, es decir, las orientación (${}^i{}_{i-1}R$) y posición (${}^i{}_{i-1}P$) entre los eslabones en coordenadas homogéneas pueden ser expresadas en la forma

$${}^i{}_{i-1}T = \left(\begin{array}{c|c} {}^i{}_{i-1}R & {}^i{}_{i-1}P \\ \hline 0 & 1 \end{array} \right) \quad (2.2)$$

o en forma general, considerando los parámetros DH, se expresa como

$${}^i{}_{i-1}T = \left(\begin{array}{ccc|c} c\theta_i & -s\theta_i & 0 & a_i \\ s\theta_i c\alpha_{i-1} & c\theta_i c\alpha_{i-1} & -s\alpha_{i-1} & -s\alpha_{i-1}d_i \\ s\theta_i s\alpha_{i-1} & c\theta_i s\alpha_{i-1} & c\alpha_{i-1} & c\alpha_{i-1}d_i \\ \hline 0 & 0 & 0 & 1 \end{array} \right) \quad (2.3)$$

De acuerdo a la Tabla 2.1 y a (2.3) se tienen las relaciones de transformación entre los eslabones de la siguiente forma:

$${}^0{}_1T = \begin{pmatrix} c_1 & -s_1 & 0 & 0 \\ s_1 & c_1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}; \quad {}^1{}_2T = \begin{pmatrix} c_2 & -s_2 & 0 & l_1 \\ s_2 & c_2 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}; \quad {}^2{}_3T = \begin{pmatrix} 1 & 0 & 0 & l_2 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \quad (2.4)$$

donde $s_1 = \text{sen}(\theta_1)$, $c_1 = \text{cos}(\theta_1)$ y así sucesivamente.

Ahora, volviendo a la descripción dinámica del manipulador, se tiene que la energía cinética del i -ésimo eslabón, k_i , se obtiene a partir de

$$k_i = \frac{1}{2} m_i v_{C_i}^T v_{C_i} + \frac{1}{2} {}^i w_i^T C_i I_i {}^i w_i \quad (2.5)$$

donde el primer término corresponde a la energía cinética de la velocidad lineal del centro de masa i con

$${}^{i+1}v_{i+1} = {}^{i+1}R({}^i v_i + {}^i w_i \times {}^i P_{i+1}) \quad (2.6)$$

mientras el segundo correspondiente a la energía cinética de la velocidad angular del eslabón i con

$${}^{i+1}w_{i+1} = {}^{i+1}R {}^i w_i + \begin{pmatrix} 0 \\ 0 \\ \dot{\theta}_{i+1} \end{pmatrix} \quad (2.7)$$

para más detalles ver [4] y [5]. La ecuación de energía cinética total es

$$k = \sum_{i=1}^N k_i. \quad (2.8)$$

La energía potencial del i -ésimo eslabón viene dado por

$$u_i = -m_i {}^0g^T P_{C_i} + u_{ref_i} \quad (2.9)$$

donde N es el número de eslabones, 0g es el vector de gravedad, ${}^0P_{C_i}$ es el vector que localiza el centro de masa del i -ésimo eslabón y u_{ref_i} es elegido tal que el valor mínimo de u_i llegue a ser cero.

Al igual que con la energía cinética, la energía potencial total es calculada como

$$u = \sum_{i=1}^N u_i. \quad (2.10)$$

Así, la ecuación total (suma de energías) para la descripción de movimiento del manipulador es

$$\frac{d}{dt} \frac{\partial k}{\partial \dot{\Theta}} - \frac{\partial k}{\partial \Theta} + \frac{\partial u}{\partial \Theta} = \tau. \quad (2.11)$$

De (2.11) se puede obtener la forma general para robots manipuladores como

$$\tau = M(\Theta)\ddot{\Theta} + V(\Theta, \dot{\Theta}) + G(\Theta) \quad (2.12)$$

que planteada en forma dinámica de aceleración resulta en

$$\ddot{\Theta} = M^{-1}(\Theta) \left[\tau - V(\Theta, \dot{\Theta}) - G(\Theta) - F(\Theta, \dot{\Theta}) \right] \quad (2.13)$$

donde $M^{-1}(\Theta)$ representa la contribución de las fuerzas inerciales a la ecuación dinámica, $V(\Theta, \dot{\Theta})$ representa las fuerzas centrífugas y de Coriolis, $G(\Theta)$ representa las fuerzas gravitacionales, el término adicional $F(\Theta, \dot{\Theta})$ simula la fricción de las uniones de los eslabones y τ es el vector de fuerzas o pares (*torques*) aplicados.

Por otro lado, las dinámicas de los modelos matemáticos de los robots manipuladores resultan acotadas [21]; este aspecto es importante para realizar la identificación con un esquema o modelo neuronal. De acuerdo al análisis de los modelos dinámicos de los robots manipuladores, las dinámicas del robot están acotadas de la siguiente manera:

$$\begin{aligned} \|M(\Theta)\ddot{\Theta}\| &\leq c_m \|\ddot{\Theta}\| \\ \|V(\Theta, \dot{\Theta})\| &\leq c_v \|\dot{\Theta}\| \\ \|F(\dot{\Theta})\| &\leq c_f \|\dot{\Theta}\| \\ \|G(\Theta)\| &\leq c_g \end{aligned} \quad (2.14)$$

donde c_m , c_v , c_f y c_g son valores constantes positivos; para su determinación habría que evaluar las funciones correspondientes a cada término de la dinámica del robot en función de los ángulos y velocidades que intervienen. Los términos de acotamiento (2.14), hacen que las dinámicas de acoplamiento involucradas en dichos términos (dinámicas de los diferentes eslabones) puedan ser vistas como disturbios acotados y de esa forma lograr la descentralización, como se verá en la Sección (2.1.3).

Los datos iniciales ($i = 0$) para la obtención del modelo (usando la formulación de Lagrange) de un robot de dos DOF son los siguientes:

$${}^0v_0 = \begin{pmatrix} 0 \\ 0 \\ 0 \end{pmatrix}; \quad {}^0w_0 = \begin{pmatrix} 0 \\ 0 \\ 0 \end{pmatrix}; \quad {}^0P_1 = \begin{pmatrix} 0 \\ 0 \\ 0 \end{pmatrix}; \quad {}^0g = \begin{pmatrix} 0 \\ -g \\ 0 \end{pmatrix} \quad (2.15)$$

Desarrollando las ecuaciones (2.5) y (2.9) para el manipulador de dos grados de libertad, donde $i = 1, 2$, se tiene que

$$k_1 = \frac{1}{2} (m_1 l_{c1}^2 + I_{zz1}) \dot{\theta}_1^2 \quad (2.16)$$

$$k_2 = \frac{1}{2} m_2 \left[l_1^2 \dot{\theta}_1^2 + 2c_2 l_1 l_{c2} (\dot{\theta}_1^2 + \dot{\theta}_1 \dot{\theta}_1) \right] + \frac{1}{2} (m_2 l_{c2}^2 + I_{zz2}) (\dot{\theta}_1 + \dot{\theta}_2)^2 \quad (2.17)$$

$$u_1 = m_1 g l_{c1} s_1 + m_1 g l_{c1} \quad (2.18)$$

$$u_2 = m_2 g (l_1 s_1 + l_{c2} s_{12}) + m_2 g (l_1 + l_{c2}). \quad (2.19)$$

Resolviendo (2.11) conforme al valor de sus parámetros, es decir, energía cinética y potencial, se determina la estructura (2.12) con los elementos siguientes:

$$M(\Theta) \ddot{\Theta} = \begin{pmatrix} m_1 l_{c1}^2 + m_2 l_1^2 m_2 l_{c2}^2 + 2m_2 l_1 l_{c2} c_2 + I_{zz1} + I_{zz2} & m_2 l_{c2}^2 + m_2 l_1 l_{c2} c_2 + I_{zz2} \\ m_2 l_{c2}^2 + m_2 l_1 l_{c2} c_2 + I_{zz2} & m_2 l_{c2}^2 + I_{zz2} \end{pmatrix} \begin{pmatrix} \ddot{\theta}_1 \\ \ddot{\theta}_2 \end{pmatrix} \quad (2.20)$$

$$V(\Theta, \dot{\Theta}) = \begin{pmatrix} -m_2 l_1 l_{c2} s_2 (\dot{\theta}_1 + \dot{\theta}_2) \dot{\theta}_2 - m_2 l_1 l_{c2} \dot{\theta}_1 \dot{\theta}_2 s_2 \\ m_2 l_1 l_{c2} s_2 \dot{\theta}_1^2 \end{pmatrix} \quad (2.21)$$

$$G(\Theta) = g \begin{pmatrix} (m_1 l_{c1} + m_2 l_1) c_1 + m_2 l_{c2} c_{12} \\ m_2 l_{c2} c_{12} \end{pmatrix} \quad (2.22)$$

$$\tau = \begin{pmatrix} \tau_1 \\ \tau_2 \end{pmatrix} \quad (2.23)$$

donde $\Theta = [\theta_1 \ \theta_2]^T$ Simbólicamente pueden ser escritas como

$$M(\Theta) = \begin{pmatrix} D_{11} & D_{12} \\ D_{12} & D_{22} \end{pmatrix}; \quad V(\Theta, \dot{\Theta}) = \begin{pmatrix} V_1 \\ V_2 \end{pmatrix}; \quad G(\Theta) = \begin{pmatrix} G_1 \\ G_2 \end{pmatrix}; \quad F(\Theta, \dot{\Theta}) = \begin{pmatrix} F_1 \\ F_2 \end{pmatrix}$$

Modelo del robot de dos DOF

Para análisis y simulación se hará referencia a la estructura (2.13), en donde es necesario realizar un cambio de variable en la forma

$$q_1 = \theta_1; \quad q_2 = \theta_2; \quad q_3 = \dot{\theta}_1; \quad q_4 = \dot{\theta}_2. \quad (2.24)$$

Así, el modelo del manipulador rígido de dos grados de libertad viene descrito por

$$\begin{pmatrix} \dot{q}_1 \\ \dot{q}_2 \\ \dot{q}_3 \\ \dot{q}_4 \end{pmatrix} = \begin{pmatrix} q_3 \\ q_4 \\ \frac{D_{22}}{D_{11}D_{22}-D_{12}^2}(\tau_1 - V_1 - G_1 - F_1) - \frac{D_{12}}{D_{11}D_{22}-D_{12}^2}(\tau_2 - V_2 - G_2 - F_2) \\ -\frac{D_{12}}{D_{11}D_{22}-D_{12}^2}(\tau_1 - V_1 - G_1 - F_1) + \frac{D_{11}}{D_{11}D_{22}-D_{12}^2}(\tau_2 - V_2 - G_2 - F_2) \end{pmatrix} \quad (2.25)$$

donde:

$$D_{11}(\Theta) = m_1 l_{c1}^2 + m_2 l_1^2 m_2 l_{c2}^2 + 2m_2 l_1 l_{c2} c_2 + I_{zz1} + I_{zz2}$$

$$D_{12}(\Theta) = m_2 l_1 l_{c2} c_2 + I_{zz2}$$

$$D_{22}(\Theta) = m_2 l_{c2}^2 + I_{zz2}$$

$$V_1(\Theta, \dot{\Theta}) = -m_2 l_1 l_{c2} s_2 (q_3 + q_4) q_4 - m_2 l_1 l_{c2} q_3 q_4 s_2$$

$$V_2(\Theta, \dot{\Theta}) = m_2 l_1 l_{c2} s_2 q_3^2$$

$$G_1(\Theta) = g((m_1 l_{c1} + m_2 l_1) c_1 + m_2 l_{c2} c_{12})$$

$$G_2(\Theta) = g m_2 l_{c2} c_{12}$$

$$F_1(\Theta, \dot{\Theta}) = \mu_1 q_3$$

$$F_2(\Theta, \dot{\Theta}) = \mu_2 q_4.$$

Recordando que, $s_1 = \text{sen}(q_1)$, $s_{12} = \text{sen}(q_1 + q_2)$ y así respectivamente; τ_1 y τ_2 son los pares (torques) aplicados.

Modelo del robot de dos DOF en tiempo discreto

Después de una discretización aproximada [11], por series de Taylor del modelo (2.25), se obtiene el modelo discreto como

$$\begin{aligned} \chi_{1,k+1}^1 &= \chi_{1,k}^1 + \chi_{1,k}^2 T \\ \chi_{2,k+1}^1 &= \chi_{2,k}^1 + \chi_{2,k}^2 T \\ \chi_{1,k+1}^2 &= \chi_{1,k}^2 + \left(\frac{-D_{22}(V_1 + G_1 + F_1) + D_{12}(V_2 + G_2 + F_2)}{D_{11}D_{22} - D_{12}^2} \right. \\ &\quad \left. + \frac{D_{22} u_{1,k} - D_{12} u_{2,k}}{D_{11}D_{22} - D_{12}^2} \right) T \\ \chi_{2,k+1}^2 &= \chi_{2,k}^2 + \left(\frac{D_{12}(V_1 + G_1 + F_1) - D_{11}(V_2 + G_2 + F_2)}{D_{11}D_{22} - D_{12}^2} \right. \\ &\quad \left. + \frac{-D_{12} u_{1,k} + D_{11} u_{2,k}}{D_{11}D_{22} - D_{12}^2} \right) T \end{aligned} \quad (2.26)$$

donde χ_1^1 , χ_2^1 , χ_1^2 , χ_2^2 es el equivalente en tiempo discreto de q_1 , q_2 , q_3 , q_4 en tiempo continuo respectivamente; T es el periodo de muestreo, $u_{1,k}$ y $u_{2,k}$ son los pares (torques) aplicados.

Es evidente que (2.26) tiene la forma no lineal controlable por bloques (NBC por sus siglas en inglés *Nonlinear Block-Controllable*).

El modelo descrito en (2.26) es el que describe el comportamiento dinámico aproximado (en tiempo discreto) de un robot de dos DOF en posición vertical; de éste puede deducirse fácilmente el modelo de un robot de dos DOF planar¹ con sólo hacer $G_1(\Theta) = 0$ y $G_2(\Theta) = 0$, ya que en el robot planar no tiene efecto la gravedad.

2.1.3. Una clase de sistemas descentralizados

El modelo (2.26) puede analizarse para ser tratado como un sistema descentralizado [10], es decir, un sistema completo el cual pueda subdividirse en subsistemas interconectados por medio de algún término como se mostrará en el desarrollo siguiente. Esto es útil para facilitar el análisis de sistemas no lineales grandes y complejos; por ejemplo en robots con varios grados de libertad. Para esto, considere una clase de sistema no lineal en tiempo discreto en la forma

$$\chi_{i,k+1} = f_i(\chi_{i,k}) + B_i(\chi_{i,k}) u_{i,k} + \Gamma_{i\ell,k} \quad (2.27)$$

el cual puede ser presentado (posiblemente después de una transformación no lineal) en la forma NBC [12] y que consiste de r bloques

$$\begin{aligned} \chi_{i,k+1}^1 &= f_i^1(\chi_{i,k}^1) + B_i^1(\chi_{i,k}^1) \chi_{i,k}^2 + \Gamma_{i\ell}^1 \\ \chi_{i,k+1}^2 &= f_i^2(\chi_{i,k}^1, \chi_{i,k}^2) + B_i^2(\chi_{i,k}^1, \chi_{i,k}^2) \chi_{i,k}^3 + \Gamma_{i\ell}^2 \\ &\vdots \\ \chi_{i,k+1}^{r-1} &= f_i^{r-1}(\chi_{i,k}^1, \chi_{i,k}^2, \dots, \chi_{i,k}^{r-1}) \\ &\quad + B_i^{r-1}(\chi_{i,k}^1, \chi_{i,k}^2, \dots, \chi_{i,k}^{r-1}) \chi_{i,k}^r + \Gamma_{i\ell}^{r-1} \\ \chi_{i,k+1}^r &= f_i^r(\chi_{i,k}) + B_i^r(\chi_{i,k}) u_{i,k} + \Gamma_{i\ell}^r \end{aligned} \quad (2.28)$$

donde se define que $\chi_i \in R^{n_i}$. $\chi_i = [\chi_i^{1T} \chi_i^{2T} \dots \chi_i^{rT}]^T$ y además con $\chi_i^j \in \mathfrak{R}^{n_{ij} \times 1}$, $\chi_i^j = [\chi_{i1}^j \chi_{i2}^j \dots \chi_{il}^j]^T$ $i = 1, \dots, N$; $j = 1, \dots, r$; $l = 1, \dots, n_{ij}$; N es el número de subsistemas, $u_i \in R^{m_i}$, el rango de $B_i^j = n_{ij}$, $\sum_{j=1}^r n_{ij} = n_i$, $\forall \chi_i^j \in D_{\chi_i^j} \subset R^{n_{ij}}$. Se supone que f_i^j , B_i^j y $\Gamma_{i\ell}^j$ son funciones suaves y acotadas, $f_i^j(0) = 0$ y $B_i^j(0) = 0$. Los enteros $n_{i1} \leq n_{i2} \leq \dots \leq n_{ij} \leq m_i$ definen las diferentes estructuras de los sistemas. Los términos de interconexión *matched* (términos que se encuentran dentro del subespacio de control) y *unmatched* (términos que

¹ Sobre este modelo se realizó el análisis e implementación en tiempo real del controlador, pues el robot físico disponible es un robot rígido planar de dos DOF.

no se encuentran dentro del subespacio de control) están dados por

$$\begin{aligned}
 \Gamma_{il}^1 &= \sum_{\ell=1, \ell \neq i}^N \gamma_{il}^1(\chi_{\ell,k}^1) \\
 \Gamma_{il}^2 &= \sum_{\ell=1, \ell \neq i}^N \gamma_{il}^2(\chi_{\ell,k}^1, \chi_{\ell,k}^2) \\
 &\vdots \\
 \Gamma_{il}^{r-1} &= \sum_{\ell=1, \ell \neq i}^N \gamma_{il}^{r-1}(\chi_{\ell,k}^1, \chi_{\ell,k}^2, \dots, \chi_{\ell,k}^{r-1}) \\
 \Gamma_{il}^r &= \sum_{\ell=1, \ell \neq i}^N \gamma_{il}^r(\chi_{\ell})
 \end{aligned} \tag{2.29}$$

donde χ_{ℓ} representa el vector de estados del ℓ -ésimo subsistema con $1 \leq \ell < N$ y $\ell \neq i$. Los términos (2.29) son acotados por las funciones no lineales γ_{il}^j .

2.2. Identificación con redes neuronales

Para llevar a cabo la identificación de sistemas se pueden usar varios tipos de modelos, por ejemplo, continuos, discretos, lineales o no, entre otros. Esta tesis trabaja sobre modelos discretos y no lineales. Para ello se propone un modelo con redes neuronales, el cual será la base para el diseño de la ley de control. Cabe mencionar que como la ley de control está en tiempo discreto, ésta se puede implementar directamente en un computador digital.

Existen varios tipos de redes neuronales, entre las que están las estáticas y dinámicas. Esta tesis se enfoca en las redes neuronales dinámicas o redes neuronales recurrentes (RNN, por las siglas en inglés de *Recurrent Neural Networks*), en las cuales las señales de las neuronas son integradas o retrasadas y retroalimentadas a la red [19] y que tienen la capacidad de modelar (identificar) sistemas no lineales inciertos (incertidumbre en algunos o incluso todos los parámetros) [15]. Las RNN pueden expresarse en espacio de estados, siendo esto conveniente para análisis y diseño. Existen diferentes modelos de redes que de acuerdo a sus variables de alimentación se pueden obtener: paralelo, serie-paralelo, etc., para más detalles ver [17]. El modelo que se utilizará es el serie-paralelo, pues éste involucra las entradas externas de la red, es decir, los estados de la planta o proceso al que se pretende modelar (identificar) con la red neuronal.

En esta tesis, para realizar la identificación, se suponen disponibles todos los estados del sistema (planta).

2.2.1. Modelo de las redes neuronales recurrentes discretas

Una de las redes más comunes en este tipo es la de Hopfield y que para tareas de control se pueden extender a las llamadas redes neuronales recurrentes de alto orden (RHONN por sus siglas en inglés de *Recurrent High Order Neural Networks*). La red neuronal será usada para identificar el sistema (2.27) y así la RHONN se puede describir mediante la ecuación en diferencias de la siguiente forma [15]:

$$x_{j,k+1} = w_{j,k}^T \rho_j(x_{j,k}, u_{j,k}) \quad (2.30)$$

donde $x_j(k)$ es el estado de la j -ésima neurona en la iteración k con $j = 1, \dots, n_i$; w_j es el vector de pesos adaptables en línea y definiendo $\rho_i(x(k), u(k))$ como

$$\rho_j(x_k, u_k) = \begin{bmatrix} \rho_{j1,k} \\ \rho_{j2,k} \\ \vdots \\ \rho_{jL_p,k} \end{bmatrix} = \begin{bmatrix} \prod_{\ell \in I_1} Z_{j\ell}^{d_{j\ell}(1)} \\ \prod_{\ell \in I_2} Z_{j\ell}^{d_{j\ell}(2)} \\ \vdots \\ \prod_{\ell \in I_{L_p}} Z_{j\ell}^{d_{j\ell}(L_p)} \end{bmatrix} \quad (2.31)$$

con valores enteros no negativos de $d_{j\ell}$ y donde L_p es el número p de conexiones de alto orden de la red neuronal. El conjunto $\{I_1, I_2, \dots, I_{L_p}\}$ es una colección de subconjuntos no ordenados de $\{1, 2, \dots, n_i + m_i\}$ elementos y m_i es el número de entradas externas. El vector Z_j está definido de la siguiente forma:

$$Z_j = \begin{bmatrix} Z_{j1} \\ \vdots \\ Z_{jn_i} \\ Z_{jn_i+1} \\ \vdots \\ Z_{jn_i+m_i} \end{bmatrix} = \begin{bmatrix} S(x_{1,k}) \\ \vdots \\ S(x_{j,k}) \\ u_{1,k} \\ \vdots \\ u_{m_i,k} \end{bmatrix} \quad (2.32)$$

con $u_i = [u_{i1}, u_{i2}, \dots, u_{im_i}]^T$ como el vector de entrada a la red neuronal.

La función sigmoidal $S(\cdot)$ está definida como

$$S(x) = \frac{\alpha_j}{1 + e^{-\beta_j x}} - \gamma_j$$

donde la sigmoide $S(\cdot) \in [-1, 1]$; α , β y γ son constantes positivas.

2.2.2. Redes neuronales descentralizadas

Ya que el modelo de la planta (2.28) cumple la forma NBC, es lógico diseñar un identificador neuronal en esta forma. Para resolver el problema de identificación y control descentralizado, se proponen las siguientes modificaciones de la RHONN discreta (2.30) para el sistema descrito por (2.28):

Solamente las conexiones de primer orden pueden ser introducidas. Además, se debe de diseñar la red neuronal tal que se tenga una estructura en la forma no lineal controlable por bloques.

- Algunos de los componentes del vector Z_j pasan a través de alguna función no necesariamente sigmoideal.
- Algunos de los pesos de la red deben ser fijos para asegurar la controlabilidad del identificador.

Basándose en dichas modificaciones y usando la estructura del sistema (2.28), se propone el siguiente modelo de red neuronal descentralizado:

$$x_{i,k+1}^j = W_{i,k}^j \rho_i(x_{i,k}^j) + W_i^{\prime j} \psi_i^j(x_{i,k}^{j+1}, u_{i,k}) \quad (2.33)$$

el cual puede ser presentado en la forma NBC [12] y que consiste también de r bloques

$$\begin{aligned} x_{i,k+1}^1 &= \sum_{l=1, p=1}^{n_{i1}, L_{1p}} w_{ilp,k}^1 \prod_{\ell \in I_{1p}} (Z_{ilp}^1)^{d_{1\ell}(p)} + W_i^{\prime 1} x_{i,k}^2 \\ x_{i,k+1}^2 &= \sum_{l=1, p=1}^{n_{i2}, L_{2p}} w_{ilp,k}^2 \prod_{\ell \in I_{2p}} (Z_{ilp}^2)^{d_{2\ell}(p)} + W_i^{\prime 2} x_{i,k}^3 \\ &\vdots \\ x_{i,k+1}^{r-1} &= \sum_{l=1, p=1}^{n_{i(r-1)}, L_{(r-1)p}} w_{ilp,k}^{r-1} \prod_{\ell \in I_{(r-1)p}} (Z_{ilp}^{r-1})^{d_{(r-1)\ell}(p)} \\ &\quad + W_i^{\prime (r-1)} x_{i,k}^r \\ x_{i,k+1}^r &= \sum_{l=1, p=1}^{n_{ir}, L_{rp}} w_{ilp,k}^r \prod_{\ell \in I_{rp}} (Z_{ilp}^r)^{d_{r\ell}(p)} + W_i^{\prime r} u_{i,k} \end{aligned} \quad (2.34)$$

donde $x_i = [x_i^{1T} x_i^{2T} \dots x_i^{rT}]$ es el i -ésimo bloque de estado neuronal con las mismas propiedades que (2.28), $W_{i,k}^j$ son las matrices de pesos adaptables en línea, $W_i^{\prime j}$ son matrices con

parámetros fijos y $\text{rango}(W_i^j) = n_{ij}$, con $j = 1, \dots, r$; ψ denota una función lineal de x_i^j o u_i correspondiente a la estructura de la planta (2.28) o entradas a la red, respectivamente. El vector ρ_i en (2.33) es como (2.31), pero el vector Z_{ilp}^j es redefinido como

$$Z_{ilp}^j = \begin{bmatrix} Z_{i1}^j \\ \vdots \\ Z_{il}^j \end{bmatrix} = \begin{bmatrix} S(x_{i1,k}^j) \\ \vdots \\ S(x_{il,k}^j) \end{bmatrix}$$

Las matrices de pesos ajustables en línea $W_{i,k}^j$ son definidas como

$$W_{i,k}^j = \begin{bmatrix} w_{i11}^j & \cdots & w_{i1p}^j \\ \vdots & & \vdots \\ w_{il1}^j & \cdots & w_{ilp}^j \end{bmatrix}$$

Observación 1. Es importante notar que (2.34) no considera los términos de interconexión, pues estos son considerados implícitos en los términos que se están ajustando.

2.2.3. Ley de aprendizaje en línea

El esquema de identificación descentralizada y el esquema de control están basados solamente sobre información disponible para cada subsistema.

Basándose en resultados presentados en [9] y [15], se hace la suposición de que existe una RHONN la cual describe completamente a (2.27); por consiguiente, el modelo de la planta puede ser descrito por

$$\chi_{i,k+1} = W_{i,k}^* \rho_i(\chi_{i,k}, u_{i,k}) + v_{i,k} \quad (2.35)$$

donde $W_{i,k}^* \in R^{L_{ip}}$ es la matriz desconocida de pesos óptimos [8], y el error de modelado $v_{i,k}$ está dado por

$$v_{i,k} = f_i(\chi_{i,k}) + B_i(\chi_{i,k}) u_{i,k} + \Gamma_{il,k} - W_{i,k} \rho(\chi_{i,k}, u_{i,k}).$$

Observación 2. El término de error de modelado $v_{i,k}$ puede ser hecho arbitrariamente pequeño seleccionando apropiadamente el número L_p de conexiones de alto orden [15].

Para la identificación y control de (2.35) se pueden usar varios modelos [9] para (2.33), aquí se mencionan sólo dos por ser los más usuales:

Modelo paralelo

$$x_{i,k+1}^j = W_{i,k}^j \rho_i(x_{i,k}^j) + W_i^{\prime j} \psi_i^j(x_{i,k}^{j+1}, u_{i,k}) \quad (2.36)$$

Modelo serie-paralelo

$$x_{i,k+1}^j = W_{i,k}^j \rho_i(\chi_{i,k}^j) + W_i^{\prime j} \psi_i^j(\chi_{i,k}^{j+1}, u_{i,k}). \quad (2.37)$$

La siguiente hipótesis es hecha para garantizar la estabilidad de convergencia del identificador y control:

Hipótesis 1. Los sistemas (2.28) y (2.35) son estables entrada-a-estado.

El seguimiento de una trayectoria deseada, definida en términos del estado de la planta χ_i^j , formulada en (2.28), puede ser establecida en la siguiente desigualdad:

$$\|\chi_{i\delta}^j - \chi_i^j\| \leq \|x_i^j - \chi_i^j\| + \|\chi_{i\delta}^j - x_i^j\| \quad (2.38)$$

donde $\|\cdot\|$ establece la norma Euclidiana, $\chi_{i\delta}^j$ es la señal de trayectoria deseada, la cual se supone que es suave y acotada. Ahora se establecen los siguientes requerimientos para la solución del control neuronal y poder llevar a cabo el seguimiento:

Requerimiento 1.

$$\lim_{t \rightarrow \infty} \|x_i^j - \chi_i^j\| \leq \zeta_i^j \quad (2.39)$$

siendo ζ_i^j una constante positiva pequeña.

Requerimiento 2.

$$\lim_{t \rightarrow \infty} \|\chi_{i\delta}^j - x_i^j\| = 0. \quad (2.40)$$

Un identificador neuronal basado en (2.33) asegura (2.39); el teorema que garantiza este error de identificación es dado al final de este capítulo.

El error de control, (2.40), es garantizado por un controlador discreto usando la técnica de control por bloques con modos deslizantes como se verá en el capítulo siguiente.

Las redes neuronales discretas presentarán las mismas características que las continuas y son ideales para su uso en modelado, identificación y control de sistemas discretos complejos debido a su facilidad de implementación y a la relativa sencillez de su estructura [1].

La Figura 2.3 muestra la estructura de una red neuronal discreta.

Entre las características más importantes de las redes neuronales podemos mencionar que:

- Permiten un modelado eficiente de sistemas dinámicos complejos;
- Permiten realizar identificación y control robusto sobre la planta;
- Son sencillas de diseñar y;
- El entrenamiento puede ser realizado en línea.

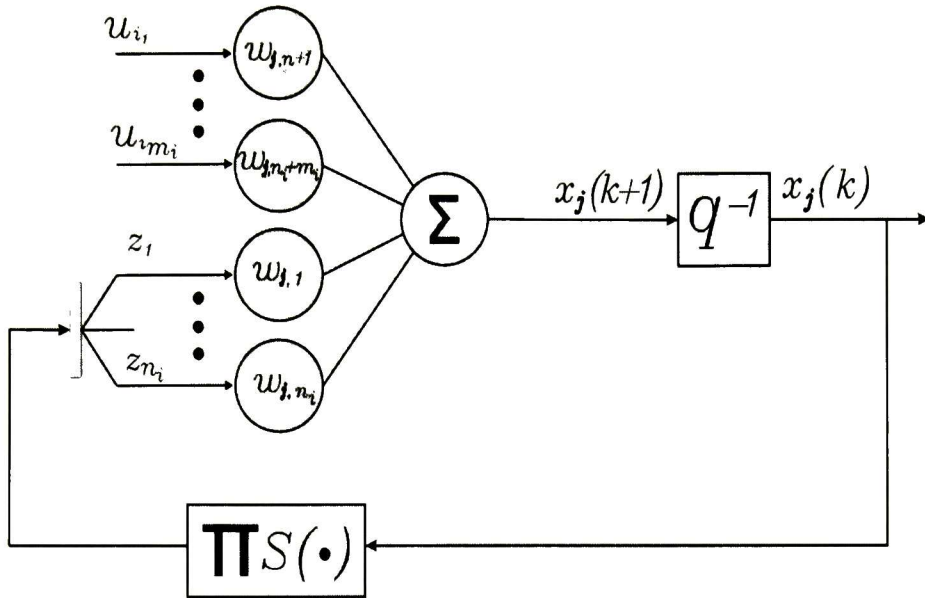


Figura 2.3: Estructura de una red neuronal discreta de alto orden.

2.3. Entrenamiento con el filtro de Kalman

El filtro de Kalman es esencialmente un conjunto de ecuaciones matemáticas que tiene por objetivo implementar un predictor-corrector tipo estimador, que es *óptimo* en el sentido de que minimiza la covarianza del error de estimación, siempre y cuando se cumplan algunas condiciones [9], [25]. La solución es recursiva tal que cada actualización de éste se realiza a partir del estado estimado anterior y un dato actual en función del error entre la observación y la estimación. Sólo se requerirá almacenar el estimado del peso anterior [17], dando al computador mayor eficiencia para una implementación del entrenamiento en tiempo real.

2.3.1. El filtro de Kalman

En base en el modelo lineal discreto de la Figura 2.4, el estado o vector de estado denotado por $w(k)$, es definido como el conjunto mínimo de datos suficientes para describir el comportamiento dinámico del sistema. El índice k es el instante de muestreo; se supone que $w(k)$ es *desconocido*, entonces, para estimarlo se hace uso de un conjunto de datos medidos denotados por $y(k)$. La descripción matemática es representada como

$$w(k+1) = F_{k+1,k} w(k) + v_1(k) \quad (2.41)$$

donde $F_{k+1,k} \in \mathfrak{R}^{L \times L}$ es la matriz de transición de estados de la iteración k a la $k + 1$, $w(k) \in \mathfrak{R}^L$ es el vector de estados en la iteración k ; $v_1(k) \in \mathfrak{R}^L$ representa el vector del ruido del proceso, considerado *Gaussiano* y con media cero.

La matriz de covarianza del ruido se define como

$$E \{v_1(l)v_1^T(k)\} = \begin{cases} Q(k) & \text{para } l = k \\ 0 & \text{para } l \neq k \end{cases} \quad (2.42)$$

mientras que para la ecuación de salida se tiene que

$$y(k) = H(k)w(k) + v_2(k) \quad (2.43)$$

donde $H \in \mathfrak{R}^{m \times L}$ es la matriz de medición, $y \in \mathfrak{R}^m$ es el vector de salida del proceso, el cual debe de estar disponible en la medición k ; $v_2 \in \mathfrak{R}^m$ representa el vector del ruido de medición el cual se considera *Gaussiano* y con media cero.

La matriz de covarianza del ruido de medición se define como

$$E \{v_2(l)v_2^T(k)\} = \begin{cases} R(k) & \text{para } l = k \\ 0 & \text{para } l \neq k. \end{cases} \quad (2.44)$$

Otra consideración a tomarse en cuenta es que, el ruido del proceso y el ruido de medición deben de estar descorrelacionados.

El vector de estados $w(k)$, dado en (2.41), es estimado por $\hat{w}(k)$ en el siguiente procedimiento:

$$\hat{w}^-(k) = F_{k+1,k} \hat{w}^-(k-1) \quad (2.45)$$

$$P^-(k) = F_{k+1,k}P^-(k-1)F^T(k, k-1) + Q(k-1) \quad (2.46)$$

$$K(k) = P^-(k)H^T(k) [H(k)P^-(k)H^T(k) + R(k)]^{-1} \quad (2.47)$$

$$\hat{w}(k) = \hat{w}^-(k) + K(k) (y(k) - H(k)\hat{w}^-(k)) \quad (2.48)$$

$$P(k) = (I - K(k)H(k)) P^-(k) \quad (2.49)$$

Inicialización para $k = 0$

$$\hat{w}(0) = E\{\chi(0)\}$$

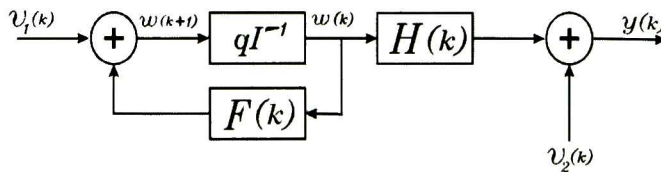


Figura 2.4: Sistema dinámico lineal en tiempo discreto.

$$P(0) = E\left\{(\chi(0) - E\{\chi(0)\})(\chi(0) - E\{\chi(0)\})\right\}$$

de esta forma \hat{w} es el mejor estimado del estado w [9].

2.3.2. El Filtro de Kalman Extendido

En la sección anterior se abordó el filtro de Kalman como un estimador de estado de un proceso en tiempo discreto, el cual es gobernado por una ecuación lineal en diferencias, pero ¿qué pasa cuando el proceso a ser estimado o cuando las mediciones del proceso son no lineales? En la práctica, los procesos en general se comportan de forma no lineal. Así, un filtro de Kalman que linealiza el sistema sobre la media actual y covarianza actual es nombrado *filtro de Kalman extendido* (EKF por sus siglas en inglés, *Extended Kalman Filter*) [25].

Considere un sistema discreto no lineal de la forma

$$w(k+1) = f(k, w(k), u(k)) + v_1(k) \quad (2.50)$$

$$y(k) = h(w(k)) + v_2(k) \quad (2.51)$$

donde $v_1(k)$ y $v_2(k)$ son ruidos independientes Gaussianos, con media cero y con matrices de covarianza del ruido del proceso $Q(k)$ y con matriz de covarianza del ruido de medición $R(k)$. La función f y h (omitiendo sus argumentos por facilidad de notación) son las funciones no lineales de transición de entrada-estado y de medición respectivamente.

Entonces, el filtro de Kalman extendido linealiza el modelo (2.50) y (2.51) por series de Taylor en cada instante de tiempo alrededor del estado estimado actual de la siguiente forma

$$F_{k+1,k} = \left. \frac{\partial f(k, w, u)}{\partial w} \right|_w = \hat{w}(k) \quad (2.52)$$

$$H(k) = \left. \frac{\partial h(k, w)}{\partial w} \right|_{w=\hat{w}^-(k)} \quad (2.53)$$

tal que F y H se aproximen a f y h , respectivamente [3] y [17]. Con este análisis se puede lograr la estimación de estados como fue descrito en el procedimiento de la sección anterior (procedimiento del filtro de Kalman normal).

2.3.3. Entrenamiento de la red neuronal

El entrenamiento tiene por objetivo encontrar los valores de pesos óptimos tal que se minimice el error de estimación del proceso a identificar con el esquema de RNN propuesto. Ahora se dará la derivación del EKF como algoritmo de entrenamiento de la RNN. Se supone

que existe una RNN que se comporta como (2.27), el cual es descrito por el siguiente sistema discreto no lineal:

$$\begin{aligned} W_{i,k+1}^{j*} &= W_{i,k}^{j*} + v_{1(i,k)}^j \\ y_{i,k}^j &= h_i(x_{i,k}^j, W_{i,k}^{j*}, u_{i,k}) + v_{2(i,k)}^j. \end{aligned} \quad (2.54)$$

La primera de ellas describe la ecuación del proceso, la cual especifica que el estado de la RNN es descrito por un proceso estacionario afectado por un ruido de proceso $v_{1(i,k)}^j$, donde el estado del sistema está dado por los valores de los pesos de la red $W_{i,k+1}^{j*}$. La segunda ecuación es conocida como la ecuación de observación o de medición, la cual representa la respuesta deseada $y_{i,k}^j$ de la RNN como una función no lineal de la entrada $u_{i,k}$ y los pesos $W_{i,k+1}^{j*}$; $y_{i,k}^j$ también está corrompida por un ruido en la medición $v_{2(i,k)}^j$. Ambos ruidos, $v_{1(i,k)}^j$ y $v_{2(i,k)}^j$, son considerados *ruidos blancos*, con media cero y descorrelacionados, con matrices de covarianza $Q_{i,k}^j$ y $R_{i,k}^j$ respectivamente.

Entonces, por medio del filtro de Kalman el sistema (2.54) será estimado de forma óptima por el siguiente sistema discreto no lineal

$$\begin{aligned} W_{i,k+1}^j &= W_{i,k}^j + \eta_i^j K_{i,k}^j e_{i,k}^j \\ \hat{y}_{i,k}^j &= h_i(x_{i,k}^j, W_{i,k}^j, u_{i,k}) \end{aligned} \quad (2.55)$$

donde η_i^j es una razón de aprendizaje adicional, K_i^j es la ganancia de Kalman y e_i^j es el error de salida.

El problema de entrenar la RNN con el filtro de Kalman puede ser descrito como encontrar la mínima media cuadrada del error de estimación del estado $W_{i,k}^{j*}$ usando todos los datos observados [9]. La solución al problema de entrenamiento es dado por

$$\begin{aligned} M_{i,k}^j &= [R_{i,k}^j + H_{i,k}^{jT} P_{i,k}^j H_{i,k}^j]^{-1} \\ K_{i,k}^j &= P_{i,k}^j H_{i,k}^j M_{i,k}^j \\ W_{i,k+1}^j &= W_{i,k}^j + \eta_i^j K_{i,k}^j e_{i,k}^j \\ P_{i,k+1}^j &= P_{i,k}^j - K_{i,k}^j H_{i,k}^{jT} P_{i,k}^j + Q_{i,k}^j \end{aligned} \quad (2.56)$$

donde la matriz $W_{i,k}^j$ representa el estimado del j -ésimo peso (estado de la red) del i -ésimo subsistema en el paso k . Este estimado está función de la ganancia de Kalman K_i^j y el error $e_{i,k}^j = y_{i,k}^j - \hat{y}_{i,k}^j$, donde $y_{i,k}^j$ es la salida de la planta y $\hat{y}_{i,k}^j$ es la salida de la RHONN. La ganancia de Kalman está función de la matriz del error P_i^j una matriz de derivadas de la salida de la red neuronal con respecto a todos los pesos que se ajustan como sigue

$$H_{i,k}^j = \left[\frac{\partial \hat{y}_{i,k}^j}{\partial W_{i,k}^j} \right]^T \quad (2.57)$$

y una matriz global de escalamiento M_i^j . Como parámetro adicional se introduce una razón de aprendizaje η_i^j tal que $0 \leq \eta_i^j \leq 1$. Usualmente P_i^j y Q_i^j son inicializadas como matrices diagonales con elementos $P_i^j(0)$ y $Q_i^j(0)$ respectivamente. En esta tesis los valores de P_i^j y Q_i^j son seleccionados como parámetros fijos.

En el apéndice A se pueden encontrar algunas consideraciones y recomendaciones para la sintonización de los parámetros del filtro de Kalman. Los valores tomados por $H_{ij,k}$, $K_{i,k}$ y $P_{i,k}$ deben de ser acotados, ésto es

$$\begin{aligned} \|H_{ij,k}\| &\leq \bar{H}_{ij} \\ \|K_{i,k}\| &\leq \bar{K}_i \\ \|P_{i,k}\| &\leq \bar{P}_i. \end{aligned} \tag{2.58}$$

Teorema 2.1: La red neuronal (2.55), entrenada con el algoritmo del EKF (2.56) para identificar la planta no lineal (2.27), asegura que el error de identificación es semiglobalmente uniformemente últimamente acotada; más aún, los pesos de la red neuronal permanecen acotados.

Para detalles de la prueba ver [2].

Capítulo 3

Control por bloques en tiempo discreto

En este capítulo el modelo de la red neuronal recurrente, utilizada para identificar el modelo de la planta, es usado para el diseño de la ley de control, en la cual se combina control por bloques en tiempo discreto y modos deslizantes. También se incluye el análisis de estabilidad con el control diseñado para el sistema completo en lazo cerrado.

3.1. Control por bloques con modos deslizantes

Dada una medición completa de los estados de la planta, el objetivo es diseñar una ley de control de seguimiento para el sistema (2.28); para esto, se usa un controlador neuronal en tiempo discreto y modos deslizantes [23]. Usando el modelo paralelo (2.36), el sistema (2.34) puede ser representado en la forma no lineal controlable por bloques [12] de la forma

$$\begin{aligned}x_{i,k+1}^1 &= W_{i,k}^1 \rho_i(x_{i,k}^1) + W_i'^1 x_{i,k}^2 \\x_{i,k+1}^2 &= W_{i,k}^2 \rho_i(x_{i,k}^1, x_{i,k}^2) + W_i'^2 x_{i,k}^3 \\&\vdots \\x_{i,k+1}^r &= W_{i,k}^r \rho_i(x_{i,k}^1, \dots, x_{i,k}^r) + W_i'^r u_{i,k}\end{aligned}\tag{3.1}$$

$$y_{i,k} = h(x_{i,k}^j)\tag{3.2}$$

donde $x_i = [x_i^{1T} x_i^{2T} \dots x_i^{rT}]$ es el i -ésimo bloque de estado neuronal con las mismas propiedades que (2.28), definido sobre una vecindad X del origen de \mathfrak{R}_i^n , y $u_{i,k} \in \mathfrak{R}_i^m$, $y_{i,k} \in \mathfrak{R}_i^p$ y donde $i = 1, \dots, N$; $j = 1 \dots, r$. $W_{i,k}^j$ son las matrices de pesos adaptables en línea, $W_i'^j$ son matrices con parámetros fijos. Por otro lado, el error de seguimiento de la salida es definido como la diferencia entre la salida del sistema $y_{i,k}$ y una señal de referencia $g(\omega_{i,k})$, es decir,

$e_{i,k} = y_{i,k} - g(\omega_{i,k})$. La señal de referencia $g(\omega_{i,k})$, es generada por un sistema externo descrito por

$$\omega_{k+1} = s(\omega_k). \quad (3.3)$$

También se establece la siguiente hipótesis:

A.1 La matriz $W_i'^j$ tiene rango completo por filas

$$\text{rank}(W_i'^j) = n_{ij}$$

con

$$\sum_{j=1}^r n_{ij} = n_i$$

3.1.1. Transformación de linealización

El sistema (3.1) será considerado con estructura $n_{i1} = n_{i2} = \dots = n_{ir} = m_i$, donde el procedimiento de diseño del control por bloques consiste de una construcción paso a paso de un nuevo sistema con estados $z_{i,k}^j = x_{i,k}^j - x_{i\delta,k}^j$, $j = 1, \dots, r$, donde $x_{i\delta,k}^j$ es el valor deseado para $x_{i,k}^j$, el cual será definido para tal construcción.

Se inicia definiendo una nueva variable $z_{i,k}^j$, como la diferencia entre la salida $y_{i,k}$ y la señal de referencia a ser seguida, es decir, $\chi_{i\delta,k}^1 = g(\omega_k)$, esta señal de referencia es generada por un sistema externo (3.3), así, el error de seguimiento es

$$z_{i,k}^1 = x_{i,k}^1 - \chi_{i\delta,k}^1 \quad (3.4)$$

donde $\chi_{i\delta,k}^j$ es la señal de trayectoria deseada.

Una vez definida la primera variable (3.4), se toma un paso adelante

$$z_{i,k+1}^1 = W_{i,k}^1 \rho_i(x_{i,k}^1) + W_i'^1 x_{i,k}^2 - \chi_{i\delta,k+1}^1. \quad (3.5)$$

La ecuación (3.5) es vista como un bloque con estado $z_{i,k}^1$, y $x_{i,k}^2$ como entrada de control, donde la dinámica deseada debe ser impuesta. Este requerimiento puede ser resuelto usando la hipótesis A.1 y anticipando la dinámica deseada para este bloque como sigue:

$$z_{i,k+1}^1 = W_{i,k}^1 \rho_i(x_{i,k}^1) + W_i'^1 x_{i,k}^2 - \chi_{i\delta,k+1}^1 = K_i^1 z_{i,k}^1 \quad (3.6)$$

donde $K_i^1 = \text{diag}\{k_{i1}^1, \dots, k_{in_{i1}}^1\}$ con $|k_{iq}^1| < 1$, $q = 1, \dots, n_{i1}$, para asegurar la estabilidad de (3.6). De (3.6), $x_{i,k}^2$ es calculada como

$$x_{i,k}^2 = \left(W_i'^1\right)^{-1} \left(-W_{i,k}^1 \rho_i(x_{i,k}^1) + \chi_{i\delta,k+1}^1 + K_i^1 z_{i,k}^1\right). \quad (3.7)$$

Note que el valor simbólico del estado $x_{i\delta,k}^2$ en (3.7) no es el valor real de tal estado, sino que representa el comportamiento para $x_{i,k}^2$ que llevará el primer bloque a la dinámica deseada. Para evitar confusiones se redefine este valor de $x_{i,k}^2$ como $x_{i\delta,k}^2$, así, la (3.7) es redefinida como

$$x_{i\delta,k}^2 = \left(W_i'^1\right)^{-1} \left(-W_{i,k}^1 \rho_i(x_{i,k}^1) + \chi_{i\delta,k+1}^1 + K_i^1 z_{i,k}^1\right).$$

Procediendo en la misma forma, se define la siguiente nueva variable como

$$z_{i,k}^2 = x_{i,k}^2 - x_{i\delta,k}^2.$$

Tomando un paso adelante en $z_{i,k}^2$ se tiene que

$$\begin{aligned} z_{i,k+1}^2 &= x_{i,k+1}^2 - x_{i\delta,k+1}^2 \\ &= W_{i,k}^2 \rho_i(x_{i,k}^1, x_{i,k}^2) + W_i'^2 x_{i,k}^3 - x_{i\delta,k+1}^2. \end{aligned}$$

la dinámica deseada para este bloque es impuesta de la forma

$$z_{i,k+1}^2 = W_{i,k}^2 \rho_i(x_{i,k}^1, x_{i,k}^2) + W_i'^2 x_{i,k}^3 - x_{i\delta,k+1}^2 = K_i^2 z_{i,k}^1 \quad (3.8)$$

donde $K_i^2 = \text{diag}\{k_{i1}^2, \dots, k_{in_{i2}}^2\}$ con $|k_{iq}^2| < 1$, $q = 1, \dots, n_{i2}$. Usando la hipótesis A.1, el valor deseado $x_{i\delta,k}^3$ es calculado de (3.8) de la siguiente forma:

$$x_{i\delta,k}^3 = \left(W_i'^2\right)^{-1} \left(-W_{i,k}^2 \rho_i(x_{i,k}^1, x_{i,k}^2) + \chi_{i\delta,k+1}^2 + K_i^2 z_{i,k}^2\right)$$

y así sucesivamente. Como último paso, la variable deseada conocida es $x_{i\delta,k}^r$, y la última nueva variable es definida como

$$z_{i,k}^r = x_{i,k}^r - x_{i\delta,k}^r.$$

De igual forma, tomando un paso adelante se tiene que

$$z_{i,k+1}^r = W_{i,k}^r \rho_i(x_{i,k}^1, \dots, x_{i,k}^r) + W_i'^r u_{i,k} - x_{i\delta,k+1}^r.$$

Se hace énfasis en que las nuevas variables $z_{i,k}^j$, forman una transformación no lineal del estado $x_{i,k}$. El sistema (3.1) puede ser representado en las nuevas variables $z_i = [z_i^{1T} z_i^{2T} \dots z_i^{rT}]^T$ de la forma

$$\begin{aligned} z_{i,k+1}^1 &= K_i^1 z_{i,k}^1 + W_i'^1 z_{i,k}^2 \\ z_{i,k+1}^2 &= K_i^2 z_{i,k}^2 + W_i'^2 z_{i,k}^3 \\ &\vdots \\ z_{i,k+1}^{r-1} &= K_i^{r-1} z_{i,k}^{r-1} + W_i'^{(r-1)} z_{i,k}^r \\ z_{i,k+1}^r &= W_{i,k}^r \rho_i(x_{i,k}^1, \dots, x_{i,k}^r) + W_i'^r u_{i,k} - x_{i\delta,k+1}^r. \end{aligned} \quad (3.9)$$

3.1.2. Ley de control

Para una implementación de control con modos deslizantes discretos [24] cuando las señales de control son acotadas, es decir

$$|u_{i,k}| \leq u_{i0} \quad (3.10)$$

una variedad (*manifold*) y una ley de control son necesarias para llevar los estados hacia tal variedad. La variedad es obtenida del procedimiento del control por bloques, y una elección natural para la variedad deslizante es $S_{i,k} = z_{i,k}^r = 0$. Entonces el sistema (3.9) es representado como sigue

$$(3.11)$$

$$z_{i,k+1}^1 = K_i^1 z_{i,k}^1 + W_i'^1 z_{i,k}^2 \quad (3.12)$$

$$z_{i,k+1}^2 = K_i^2 z_{i,k}^2 + W_i'^2 z_{i,k}^3$$

$$\vdots$$

$$(3.13)$$

$$z_{i,k+1}^{r-1} = K_i^{r-1} z_{i,k}^{r-1} + W_i'^{(r-1)} S_{i,k}$$

$$S_{i,k+1} = W_{i,k}^r \rho_i(x_{i,k}^1, \dots, x_{i,k}^r) + W_i'^r u_{i,k} - x_{i\delta,k+1}^r.$$

Una vez definida la variedad, el paso siguiente es encontrar la ley de control que tome en consideración (3.10), por lo tanto, el control $u_{i,k}$ es seleccionado de la siguiente forma [24]:

$$u_{i,k} = \begin{cases} u_{eqi,k} & \text{para } \|u_{eqi,k}\| \leq u_{i0} \\ u_{i0} \frac{u_{eqi,k}}{\|u_{eqi,k}\|} & \text{para } \|u_{eqi,k}\| > u_{i0} \end{cases} \quad (3.14)$$

donde el control equivalente $u_{eqi,k}$ es calculado de $S_{i,k+1} = 0$ como

$$u_{eqi,k} = (W_i'^r)^{-1} (-W_{i,k}^r \rho_i(x_{i,k}^1, \dots, x_{i,k}^r) + x_{i\delta,k+1}^r) \quad (3.15)$$

3.1.3. Análisis de estabilidad

Es importante garantizar que el sistema al cual se requiere controlar, opere de forma adecuada y físicamente razonable a partir de las entradas de control proporcionadas, para ello, en la parte de control se deben de considerar varias restricciones en el diseño de las entradas de control; por ejemplo, que estén acotadas; esto principalmente se hace porque el sistema físico tiene restricciones en sus componentes (fuentes de alimentación para los actuadores, pares máximos proporcionados por los actuadores, etc.), es decir, no se pueden introducir al sistema entradas con valores muy altos o incluso no acotados.

Ahora se dará la prueba de estabilidad de la ley de control (siendo ésta acotada), para mostrar que el movimiento del sistema sobre $S_{i,k} = 0$, es estable en lazo cerrado. Para ésto se escribe $S_{i,k+1}$ y $u_{eqi,k}$ como

$$\begin{aligned} S_{i,k+1} &= S_{i,k} - x_{i,k}^r + x_{i\delta,k}^r + W_{i,k}^r \rho_i(x_{i,k}^1, \dots, x_{i,k}^r) + W_i'^r u_{i,k} - x_{i\delta,k+1}^r \\ u_{eqi,k} &= - \left(W_i'^r \right)^{-1} \left(S_{i,k} - x_{i,k}^r + x_{i\delta,k}^r + W_{i,k}^r \rho_i(x_{i,k}^1, \dots, x_{i,k}^r) - x_{i\delta,k+1}^r \right) \end{aligned} \quad (3.16)$$

De (3.16) sigue que

$$\|u_{eqi,k}\| \leq \left\| \left(W_i'^r \right)^{-1} \right\| \| (S_{i,k} + f_{s,k}) \| \quad (3.17)$$

donde $f_{s,k} = -x_{i,k}^r + x_{i\delta,k}^r + W_{i,k}^r \rho_i(x_{i,k}^1, \dots, x_{i,k}^r) - x_{i\delta,k+1}^r$.

Note que, cuando $\|u_{eqi,k}\| \leq u_0$, el control equivalente es aplicado, y produce el movimiento sobre la variedad $S_{i,k} = 0$. Cuando $\|u_{eqi,k}\| > u_0$, la estrategia de control propuesta es $u_{i0} \frac{u_{eqi,k}}{\|u_{eqi,k}\|}$ y el sistema en lazo cerrado está expresado por

$$\begin{aligned} S_{i,k+1} &= S_{i,k} - x_{i,k}^r + x_{i\delta,k}^r + W_{i,k}^r \rho_i(x_{i,k}^1, \dots, x_{i,k}^r) + W_i'^r u_{i0} \frac{u_{eqi,k}}{\|u_{eqi,k}\|} - x_{i\delta,k+1}^r \\ &= \left(S_{i,k} - x_{i,k}^r + x_{i\delta,k}^r + W_{i,k}^r \rho_i(x_{i,k}^1, \dots, x_{i,k}^r) - x_{i\delta,k+1}^r \right) \left(1 - \frac{u_0}{\|u_{eqi,k}\|} \right) \\ &= (S_{i,k} + f_{s,k}) \left(1 - \frac{u_0}{\|u_{eqi,k}\|} \right) \end{aligned}$$

Haciendo un análisis de estabilidad con la función de Lyapunov $V_{i,k} = \|S_{i,k}\|$ se tiene que

$$\begin{aligned} \Delta V &= \|S_{i,k+1}\| - \|S_{i,k}\| \\ &= \|S_{i,k} + f_{s,k}\| \left(1 - \frac{u_0}{\|u_{eqi,k}\|} \right) - \|S_{i,k}\| \\ &\leq \|S_{i,k} + f_{s,k}\| \left(1 - \frac{u_0}{\left\| \left(W_i'^r \right)^{-1} \right\| \|S_{i,k} + f_{s,k}\|} \right) - \|S_{i,k}\| \\ &\leq \|S_{i,k} + f_{s,k}\| - \frac{u_0}{\left\| \left(W_i'^r \right)^{-1} \right\|} - \|S_{i,k}\| \\ &\leq \|S_{i,k}\| + \|f_{s,k}\| - \frac{u_0}{\left\| \left(W_i'^r \right)^{-1} \right\|} - \|S_{i,k}\| \\ &\leq \|f_{s,k}\| - \frac{u_0}{\left\| \left(W_i'^r \right)^{-1} \right\|} \end{aligned}$$

donde $f_{s,k} = -x_{i,k}^r + x_{i\delta,k}^r + W_{i,k}^r \rho_i(x_{i,k}^1, \dots, x_{i,k}^r) - x_{i\delta,k+1}^r$.

Si se cumple que $\left\| (W_i^{r'})^{-1} \right\| \|f_{s,k}\| \leq u_{i0}$, entonces, $\Delta V \leq 0$; así, $\|S_{i,k}\|$ decrece monótonamente y de esta forma, al decrecer $\|S_{i,k}\|$, $\|u_{eqi,k}\|$ decrece también monótonamente como se puede apreciar en (3.17).

Además, habrá un tiempo de instante crítico k_{cr} cuando $\|u_{eqi,k}\| \leq u_{i0}$ tenga lugar y entonces $u_{i,k} = u_{eqi,k}$, dando como resultado un movimiento deslizante sobre $S_{i,k} = 0$ en el próximo instante de muestreo, es decir, al aplicar $u_{i,k} = u_{eqi,k}$ como entrada de control, en un solo instante de muestreo se llega a la variedad deslizante. Este movimiento es descrito por el sistema de orden reducido

$$\begin{aligned} z_{i,k+1}^1 &= K_i^1 z_{i,k}^1 + W_i^{1'} z_{i,k}^2 \\ z_{i,k+1}^2 &= K_i^2 z_{i,k}^2 + W_i^{2'} z_{i,k}^3 \\ &\vdots \\ z_{i,k+1}^{r-1} &= K_i^{r-1} z_{i,k}^{r-1} \end{aligned} \quad (3.18)$$

El sistema (3.18) representa la ecuación de modo deslizante (SME, por sus siglas en inglés *Sliding Mode Equation*), donde la matriz K_i^l es *Schur*¹ con $l = 1, \dots, r-1$ y que tiene los eigenvalores deseados. Note que $z_{i,k}^{r-1}$ decaerá asintóticamente a cero y una vez que $z_{i,k}^{r-1} = 0$, $z_{i,k}^{r-2}$ decaerá asintóticamente a cero. Procediendo hacia atrás, finalmente se tiene que $z_{i,k}^1$ decaerá asintóticamente a cero, cumpliéndose el objetivo de control.

La única condición del análisis de Lyapunov, que garantiza la estabilidad en lazo cerrado, es que

$$\left\| (W_i^{r'})^{-1} \right\| \|f_{s,k}\| \leq u_{i0} \quad (3.19)$$

en la cual naturalmente es considerado que las condiciones iniciales admitidas sean tales que la condición (3.19) se mantenga.

¹ Matriz cuadrada con elementos reales, y que tiene eigenvalores con valor absoluto menor a la unidad.

Capítulo 4

Aplicación a un robot de dos DOF

En este capítulo se exponen los diferentes esquemas de identificación neuronal descentralizada propuestos, esto para poder llevar a cabo el control (neuronal descentralizado) en el robot de dos DOF a nivel simulación: el control está diseñado sobre el modelo neuronal usado en la identificación. Las diferentes estructuras neuronales se distinguen entre sí por el número de interconexiones y el orden de éstas.

4.1. Modelos neuronales descentralizados

Para desarrollar la identificación descentralizada del modelo dinámico de un manipulador robótico de N grados de libertad, se propone un conjunto de igual N redes neuronales independientes entre sí. Cada red neuronal identifica a un subsistema (eslabón) en función de las variables locales únicamente (posición, velocidad y par (torque) de entrada al eslabón correspondiente), y además, absorbiendo el término de interconexión del subsistema al considerarlo como perturbación externa. En la Figura 4.1 se muestra la identificación neuronal descentralizada para un robot de N grados de libertad. Los N subsistemas resultantes (uno por grado de libertad) están fuertemente interconectados y cada uno es identificado por N redes neuronales completamente independientes entre sí. En esta tesis se toma $N = 2$ para ilustrar los esquemas de identificación y control descentralizados propuestos.

Los modelos de identificación neuronal que se proponen, son esquemas de identificación de sistemas no lineales con redes neuronales recurrentes. Los modelos propuestos son de primer orden, ya que presentan buen funcionamiento y no se considera que haya justificación para el empleo de redes neuronales de alto orden, pues, cumplen con las necesidades de identificación y de control del robot de dos DOF. Al considerar al robot como sistema descentralizado, la red neuronal hace caso omiso de las interacciones (acoplamientos) entre las articulaciones, aun cuando se sabe que las dinámicas de acoplamiento son importantes en los manipuladores robóticos, esto debido a que la condición (2.14) siempre se cumple.

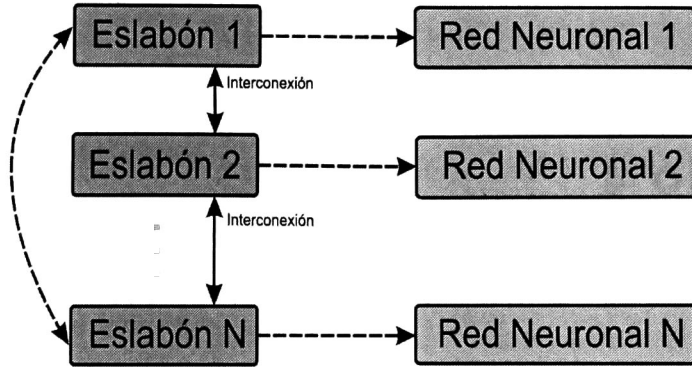


Figura 4.1: Esquema de identificación descentralizada.

Los siguientes esquemas de redes neuronales recurrentes están diseñados con el propósito de llevar a cabo la identificación del modelo dinámico de la posición y la velocidad de cada articulación, y dichas variables son las utilizadas en la red neuronal para conseguir un esquema descentralizado (información local únicamente). Las estructuras de las redes neuronales propuestas se diferencian entre sí por el número de interconexiones en la red, por las variables que se involucran en las interconexiones, así como de considerar las modificaciones presentadas en la Sección 2.2 que se sugieren hacer a la red neuronal para tener un mejor desempeño en el comportamiento y de garantizar algunas propiedades importantes en los modelos neuronales, en particular la propiedad de controlabilidad.

También en los modelos neuronales propuestos, se evita el uso de un gran número de interconexiones en las neuronas con la finalidad de obtener un consumo computacional eficiente para que los algoritmos puedan ejecutarse en tiempo real; de esta manera se puede tener la identificación en línea y se obtienen periodos de muestreo bajos (entre 1 y 10 ms).

Se utiliza el modelo serie-paralelo, descrito por (2.37), para diseñar la red neuronal.

El modelo del robot de dos DOF en la ecuación (2.26), es el *caso escalar* del sistema (2.28) donde $N = 2$, $r = 2$, $l = 1$, $n_{i1} = 1$ y $n_{i2} = 1$. Dado que es el caso escalar de (2.28), es decir $l = 1$, el índice l será omitido por facilidad de notación.

El modelo (2.26) tiene incertidumbres en sus parámetros debido a que son valores aproximados. Este modelo es usado *sólo* para poder llevar a cabo los análisis y pruebas en simulación. Para identificar (2.26) se tiene que de las ecuaciones (2.34), (2.37) y (3.1) se pueden plantear los modelos neuronales que se describen en esta sección. El último modelo neuronal propuesto describe de forma detallada el algoritmo de control. Se resalta que es el mismo procedimiento para el diseño del control en los tres modelos neuronales.

Para actualizar los pesos adaptables en línea se usa el algoritmo descrito por (2.56).

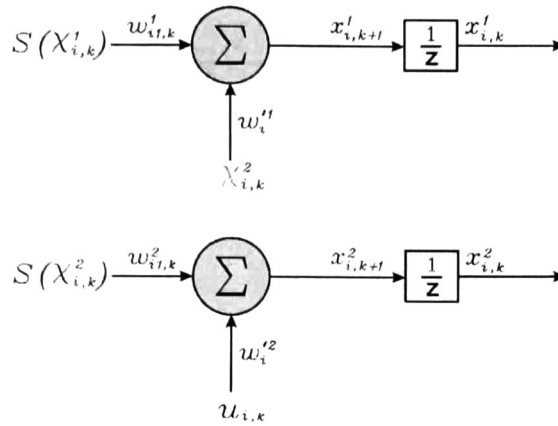


Figura 4.2: Esquema del modelo neuronal 1.

4.1.1. Modelo neuronal 1

Se emplea una sola interconexión por neurona con el fin de identificar la posición y velocidad en cada uno de los eslabones. La estructura de la red neuronal se muestra en la Figura 4.2. El modelo empleado es el siguiente:

$$\begin{aligned} x^1_{i,k+1} &= w^1_{i1,k} S_i(\chi^1_{i,k}) + w_i^1 \chi^2_{i,k} \\ x^2_{i,k+1} &= w^2_{i1,k} S_i(\chi^2_{i,k}) + w_i'^2 u_i \end{aligned} \quad (4.1)$$

donde $x^1_{i,k}$ identifica a $\chi^1_{i,k}$ y $x^2_{i,k}$ identifica a $\chi^2_{i,k}$; $i = 1, 2$; $w^j_{ip,k}$ son los pesos adaptables en línea, p es el número de pesos adaptables en línea con $p = 1$ en ambas neuronas; $j = 1, 2$; w_i^j son los parámetros fijos que garantizan controlabilidad.

4.1.2. Modelo neuronal 2

Se emplea una sola interconexión en la primera neurona y dos interconexiones para la segunda neurona con el fin de identificar la posición y velocidad en cada uno de los eslabones, respectivamente. La estructura de la red neuronal se muestra en la Figura 4.3. El modelo empleado es el siguiente:

$$\begin{aligned} x^1_{i,k+1} &= w^1_{i1,k} S_i(\chi^1_{i,k}) + w_i^1 \chi^2_{i,k} \\ x^2_{i,k+1} &= w^2_{i1,k} S_i(\chi^1_{i,k}) + w^2_{i2,k} S_i(\chi^2_{i,k}) + w_i'^2 u_i \end{aligned} \quad (4.2)$$

donde $x^1_{i,k}$ identifica a $\chi^1_{i,k}$ y $x^2_{i,k}$ identifica a $\chi^2_{i,k}$; $i = 1, 2$; $w^j_{ip,k}$ son los pesos adaptables en línea, p es el número de pesos adaptables en línea con $p = 1$ en la primera neurona y $p = 2$ para la segunda; $j = 1, 2$; w_i^j son los parámetros fijos que garantizan controlabilidad.

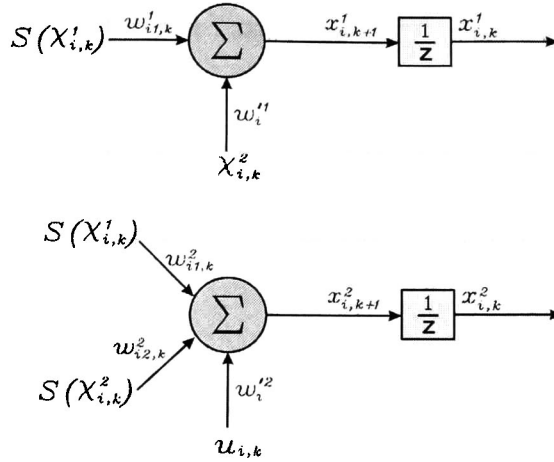


Figura 4.3: Esquema del modelo neuronal 2.

4.1.3. Modelo neuronal 3

Se emplean dos interconexiones por neurona con el fin de identificar la posición y velocidad en cada uno de los eslabones. La estructura de la red neuronal se muestra en la Figura 4.4. El modelo empleado es el siguiente:

$$\begin{aligned} x^1_{i,k+1} &= w^1_{i1,k} S_i(\chi^1_{i,k}) + w^1_{i2,k} + w'^1_i \chi^2_{i,k} \\ x^2_{i,k+1} &= w^2_{i1,k} S_i(\chi^2_{i,k}) + w^2_{i2,k} + w'^2_i u_i \end{aligned} \quad (4.3)$$

donde $x^1_{i,k}$ identifica a $\chi^1_{i,k}$ y $x^2_{i,k}$ identifica a $\chi^2_{i,k}$; $i = 1, 2$; $w^j_{ip,k}$ son los pesos adaptables en línea, p es el número de pesos adaptables en línea con $p = 2$ en ambas neuronas; $j = 1, 2$; w'^j_i son los parámetros fijos que garantizan controlabilidad.

Es importante mencionar que en este último modelo neuronal también se hace uso de las modificaciones de la Sección 2.2, en particular que algunos de los componentes del vector Z_j en (2.32) no pasen necesariamente por una función sigmoideal.

Diseño del control por bloques para el modelo neuronal 3

En esta sección se describe detalladamente el control diseñado para este modelo, ya que es el que mostró mejor desempeño tanto en simulación como en tiempo real. El diseño de control del modelo neuronal 1 y 2 son similares, la diferencia está en el número de interconexiones y las variables involucradas en las funciones sigmoideales.

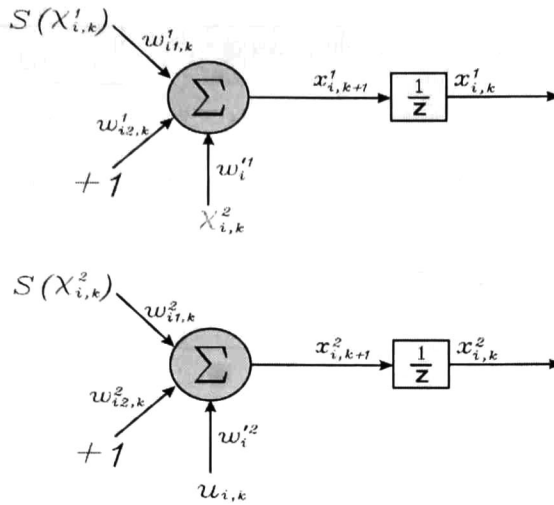


Figura 4.4: Esquema del modelo neuronal 3.

Para la descripción del algoritmo de control primeramente se tiene que el objetivo del control es conseguir que el ángulo de posición $x_{i,k}^1$ siga una señal de referencia deseada $\chi_{i\delta,k}^j$. Esto es logrado al diseñar una ley de control basada en la técnica de modos deslizantes descrita en el Capítulo 3.

Se inicia definiendo el error de seguimiento como

$$z_{i,k}^1 = x_{i,k}^1 - \chi_{i\delta,k}^1.$$

Entonces, usando (4.3) e introduciendo la dinámica deseada para $z_{i,k}^1$ resulta en

$$\begin{aligned} z_{i,k+1}^1 &= w_{i1,k}^1 S_i(\chi_{i,k}^1) + w_{i2,k}^1 + w_i^1 \chi_{i,k}^2 - \chi_{i\delta,k+1}^1 \\ &= k_i^1 z_{i,k}^1 \end{aligned} \quad (4.4)$$

con $|k_i^1| < 1$.

El valor deseado $\chi_{i\delta,k}^2$ para la entrada de pseudo-control $\chi_{i,k}^2$ es calculada de (4.4) como

$$\chi_{i\delta,k}^2 = (w_i^1)^{-1} (-w_{i1,k}^1 S_i(\chi_{i,k}^1) - w_{i2,k}^1 + \chi_{i\delta,k+1}^1 + k_i^1 z_{i,k}^1). \quad (4.5)$$

En el segundo paso se introduce la nueva variable

$$z_{i,k}^2 = x_{i,k}^2 - \chi_{i\delta,k}^2.$$

Tomando un paso adelante, se tiene entonces que

$$z_{i,k+1}^2 = w_{i1,k}^2 S_i(\chi_{i,k}^2) + w_{i2,k}^2 + w_i'^2 u_i - \chi_{i\delta,k+1}^2. \quad (4.6)$$

Ahora, el sistema (4.3) en las nuevas variables $z_{i,k}^1$ y $z_{i,k}^2$ es representado en la siguiente forma:

$$\begin{aligned} z_{i,k+1}^1 &= k_i^1 z_{i,k}^1 + w_i'^1 z_{i,k}^2 \\ z_{i,k+1}^2 &= w_{i1,k}^2 S_i(\chi_{i,k}^2) + w_{i2,k}^2 + w_i'^2 u_i \\ &\quad - \chi_{i\delta,k+1}^2. \end{aligned} \quad (4.7)$$

Una variedad deslizante es elegida como $S_{i,k} = z_{i,k}^2 = 0$. La ley de control es dada en (3.14) donde

$$u_{eq_i,k} = \left(w_i'^2\right)^{-1} \left(-w_{i1,k}^2 S_i(\chi_{i,k}^2) - w_{i2,k}^2 + \chi_{i\delta,k+1}^2\right) \quad (4.8)$$

la cual es obtenida de $S_{i,k+1} = 0$.

Un movimiento sobre $S_{i,k} = 0$ es descrito por la siguiente SME:

$$z_{i,k+1}^1 = k_i^1 z_{i,k}^1$$

la cual es estable si

$$|k_i^1| < 1.$$

4.2. Evaluación experimental

Los modelos neuronales propuestos en este capítulo, para la identificación y control descentralizada, se prueban a través de simulaciones en el modelo del robot planar descrito en la Sección 2.1, el cual tiene dos grados de libertad (con uniones revoluciones).

En lo que respecta a la simulación, computacionalmente se programa el comportamiento dinámico del robot así como el modelo neuronal, ajustándose (en línea) los parámetros de la red a partir del error en la identificación de los estados. Se utilizó la herramienta *Simulink* de MATLAB® para las simulaciones.

Para que el control por bloques diseñado cumpla con el objetivo (seguimiento de trayectoria), se propone una señal de referencia diferente para cada eslabón, y en base al modelo neuronal, se diseña la entrada de control que ha de aplicarse tanto a la RNN como al modelo dinámico del robot. Un diagrama a bloques de la descripción del sistema completo: planta (robot), RNN, controlador, referencias y entrenamiento por el filtro de Kalman extendido puede apreciarse en la Figura 4.5.

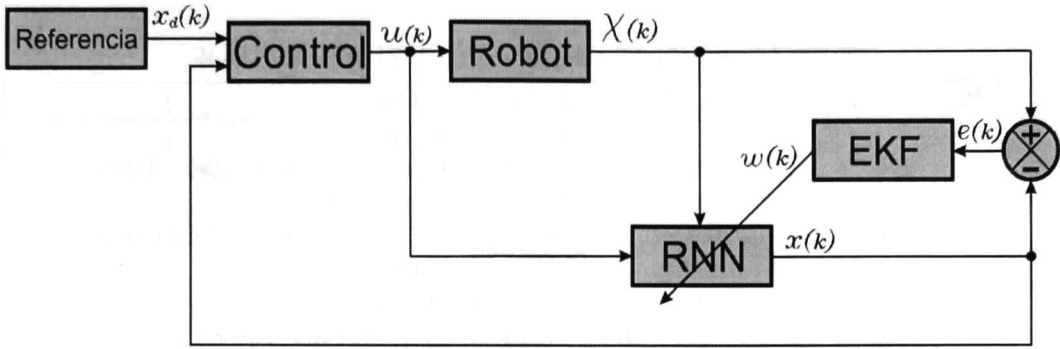


Figura 4.5: Esquema de identificación y control.

Tanto la identificación como el control son llevados a cabo en línea, es decir, desde el inicio del experimento se está identificando y usando dicho resultado para realizar el control, siendo esto lógico en el control de manipuladores físicos. Se concluye que el algoritmo de entrenamiento es eficiente y que adapta los pesos de la red neuronal de manera rápida y adecuada.

Las señales de referencia en las simulaciones (y en tiempo real en el capítulo siguiente) son descritas por

$$\chi_{1\delta,k}^1 = 1,4 \sin(0,25 k T) \text{ rad}$$

$$\chi_{2\delta,k}^1 = 1,4 \sin(0,34 k T) \text{ rad.}$$

4.2.1. Resultados de simulación

Los parámetros aproximados del modelo de la planta real (robot planar) usados sólo para simulación son dados en la Tabla 4.1.

Descripción de resultados: Modelo neuronal 1 (MN1)

El diseño del control se describe en el Capítulo 3. La ley de control es dada en (3.14) donde

$$u_{eqi,k} = \left(w_i^{\prime 2}\right)^{-1} \left(-w_{i1,k}^2 S_i(\chi_{i,k}^2) + \chi_{i\delta,k+1}^2\right) \quad (4.9)$$

con

$$\chi_{i\delta,k+1}^2 = \left(w_i^{\prime 1}\right)^{-1} \left(-w_{i1,k+1}^1 S_i(\chi_{i,k+1}^1) + \chi_{i\delta,k+2}^1 + k_i^1 z_{i,k+1}^1\right) \quad (4.10)$$

y

$$z_{i,k+1}^1 = x_{i,k+1}^1 - \chi_{i\delta,k+1}^1. \quad (4.11)$$

Tabla 4.1: Parámetros del modelo del robot de dos DOF

PARÁMETRO	VALOR	DESCRIPCIÓN
l_1	0.3 m	Longitud del eslabón 1
l_{c1}	0.2 m	Longitud del centro de masa del eslabón 1
l_2	0.25 m	Longitud del eslabón 2
l_{c2}	0.1 m	Longitud del centro de masa del eslabón 2
m_1	1 Kg	Masa del eslabón 1
m_2	0.3 Kg	Masa del eslabón 2
I_{zz1}	0.05 Kg - m ²	Momento de inercia del eslabón 1
I_{zz2}	0.004 Kg - m ²	Momento de inercia del eslabón 2
μ_1	0.005 Kg/s	Coefficiente de fricción en el eslabón 1
μ_2	0.0047 Kg/s	Coefficiente de fricción en el eslabón 2

En la Figura 4.6 se muestra la gráfica del seguimiento de trayectoria de la posición para ambos eslabones, la cual contiene la señal de referencia, la señal de la planta y la señal de identificación. Se aprecia el buen desempeño del modelo neuronal para llevar a cabo la identificación y control; se tiene error de posición significativo sólo en la parte transitoria, debido a que las condiciones iniciales tanto de la planta como de la red neuronal son diferentes. Por otra parte, el tiempo que le toma al control realizar el seguimiento es considerable (3 s), por lo que no es un esquema neuronal muy recomendable en este sentido.

En la Figura 4.7 se muestra la gráfica del error tanto para posiciones como para velocidades de ambos eslabones.

En la Figura 4.8 se muestran las variedades deslizantes S_1 y S_2 .

En la Figura 4.9 se muestran las entradas de control u_1 y u_2 .

Los resultados son aceptables, pues se cumple con el objetivo (seguimiento de trayectoria), además de que el modelo neuronal es bastante simple al requerirse un solo peso adaptable por cada estado a identificar; computacionalmente esto es eficiente, pues se requiere de un número menor de operaciones.

Descripción de resultados: Modelo neuronal 2 (MN2)

El diseño del control se describe en el Capítulo 3. La ley de control es dada en (3.14) donde

$$u_{eqi,k} = \left(w_i^{\prime 2}\right)^{-1} \left(-w_{i1,k}^2 S_i(\chi_{i,k}^2) + \chi_{i\delta,k+1}^2\right) \quad (4.12)$$

con

$$\chi_{i\delta,k+1}^2 = \left(w_i^{\prime 1}\right)^{-1} \left(-w_{i1,k+1}^1 S_i(\chi_{i,k+1}^1) - w_{i2,k+1}^2 S_i(\chi_{i,k+1}^2) + \chi_{i\delta,k+2}^1 + k_i^1 z_{i,k+1}^1\right) \quad (4.13)$$

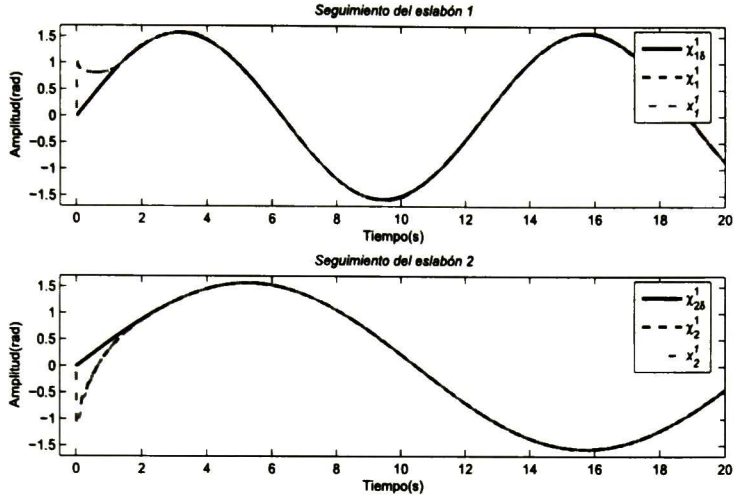


Figura 4.6: Seguimiento de trayectoria de la posición (MN1).

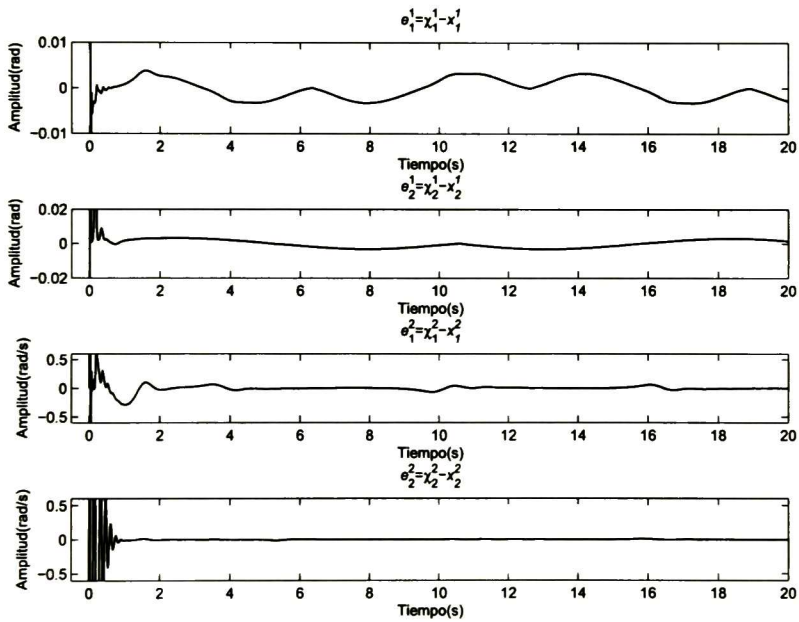
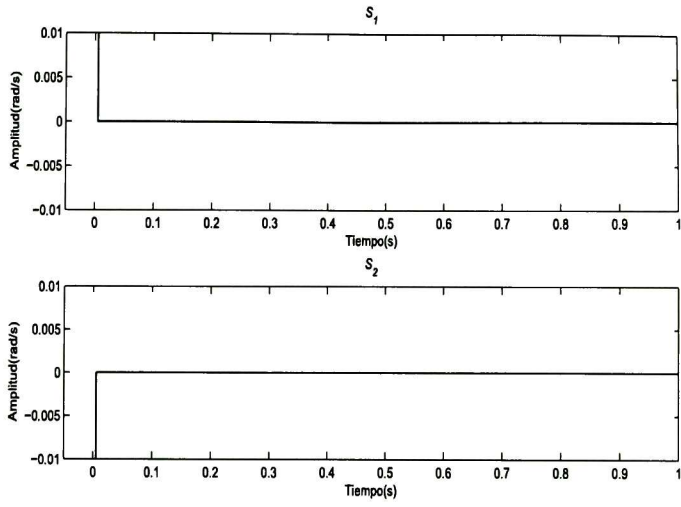
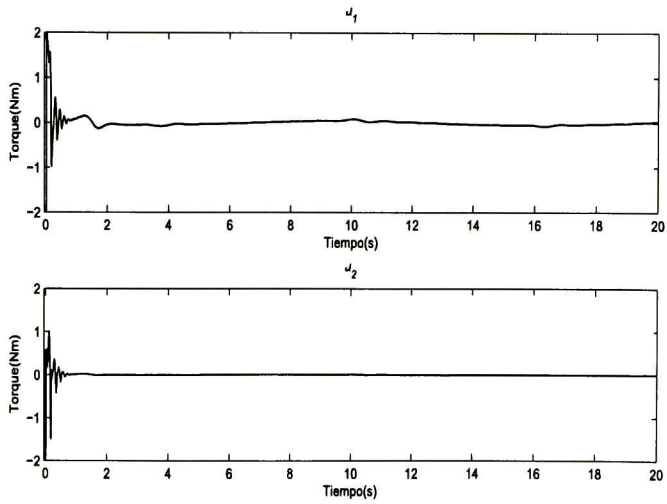


Figura 4.7: Error de posición y velocidad (MN1).

Figura 4.8: *Variedades deslizantes (MN1).*Figura 4.9: *Entradas de control (MN1).*

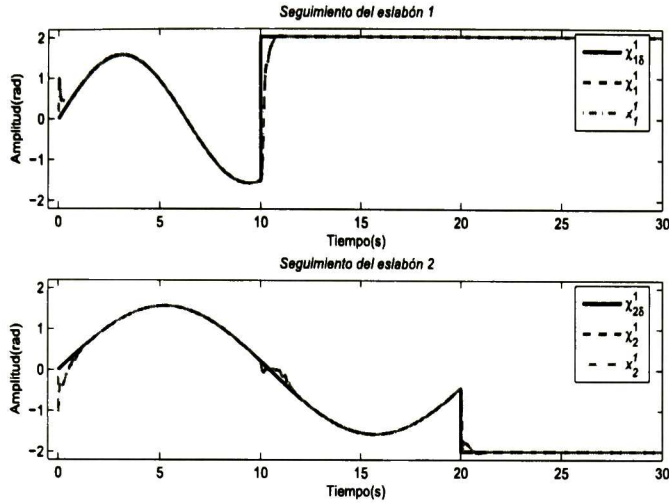


Figura 4.10: Seguimiento de trayectoria de la posición (MN2).

y

$$z_{i,k+1}^1 = x_{i,k+1}^1 - x_{i\delta,k+1}^1. \quad (4.14)$$

En la Figura 4.10 se muestra la gráfica del seguimiento de trayectoria de la posición para ambos eslabones, en ella se tiene la señal de referencia, la señal de la planta y la señal de identificación. Al igual que en el modelo neuronal 1, se observa el buen desempeño de este modelo para llevar a cabo la identificación y control; se tiene error de posición significativo sólo en la parte transitoria, debido a que las condiciones iniciales tanto de la planta como de la red neuronal no son iguales; de igual forma que el modelo neuronal 1 se tiene un tiempo considerable para realizar el seguimiento.

En este experimento se inicia con el seguimiento de una onda sinusoidal y después de unos segundos se cambia por una señal constante; de esta forma, el algoritmo de control es capaz de hacer seguir al robot una amplia variedad de trayectorias.

En la Figura 4.11 se muestra la gráfica del error tanto para posiciones como para velocidades de ambos eslabones. En ella se puede observar que los errores son despreciables.

En la Figura 4.12 se muestran las variedades deslizantes S_1 y S_2 ; también se observa que las variedades son alcanzadas en sólo un instante de muestreo al introducirse un control equivalente sobre la dinámica de las variables z_i^2 y z_i^1 .

En la Figura 4.13 se muestran las entradas de control u_1 y u_2 .

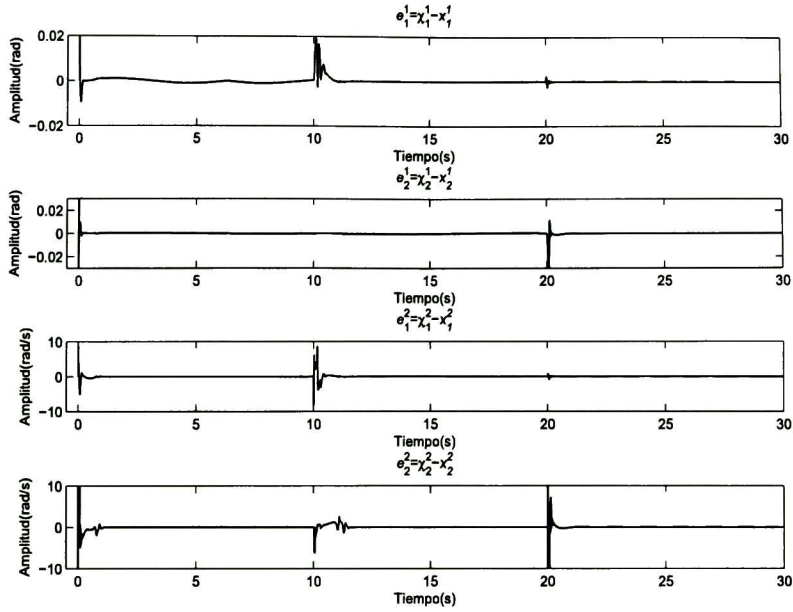


Figura 4.11: Error de posición y velocidad (MN2).

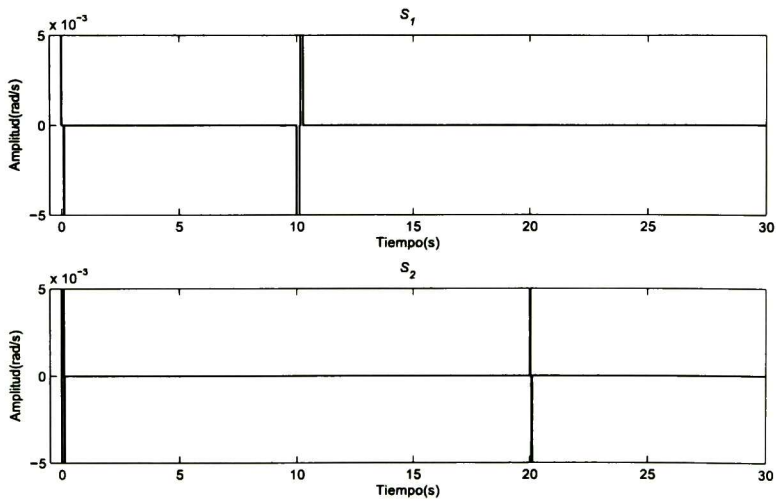


Figura 4.12: Variedades deslizantes (MN2).

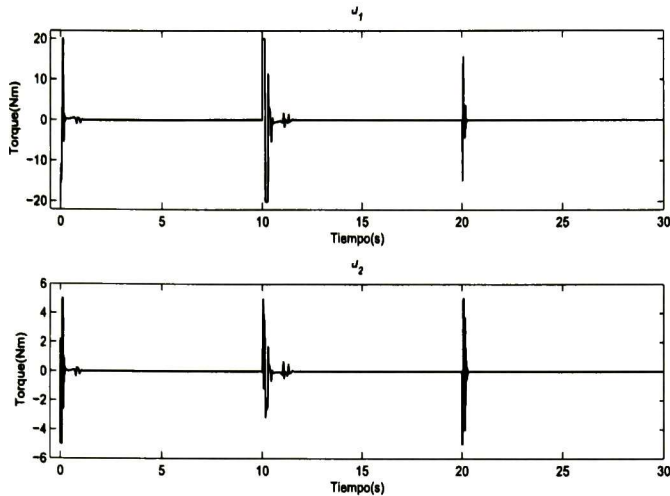


Figura 4.13: *Entradas de control (MN2)*.

Los resultados son bastante aceptables, pues se cumple con el objetivo (seguimiento de trayectoria), además de que el modelo neuronal es simple al requerirse un solo peso adaptable por cada posición a identificar y dos para la identificación de las velocidades; este esquema de identificación no es muy demandante computacionalmente.

Descripción de resultados: Modelo neuronal 3 (MN3)

El diseño del control se describe en el Capítulo 3. La ley de control es dada en (3.14) y (4.8).

En la Figura 4.14 se muestra la gráfica del seguimiento de trayectoria de la posición para ambos eslabones, la cual contiene la señal de referencia, la señal de la planta y la señal de identificación. Se aprecia un excelente desempeño del modelo neuronal para llevar a cabo la identificación y control; se tiene error de posición sólo en la parte transitoria, debido a que las condiciones iniciales tanto de la planta como de la red neuronal son diferentes. Por otra parte, el tiempo que le toma al control realizar el seguimiento es pequeño, por lo que es un esquema neuronal muy eficiente.

En este experimento, al igual que en el modelo neuronal 2, se inicia con el seguimiento de una onda senoidal y después de unos segundos se cambia por una señal constante; con esto se muestra que el algoritmo de control es capaz de hacer seguir al robot una amplia gama de trayectorias.

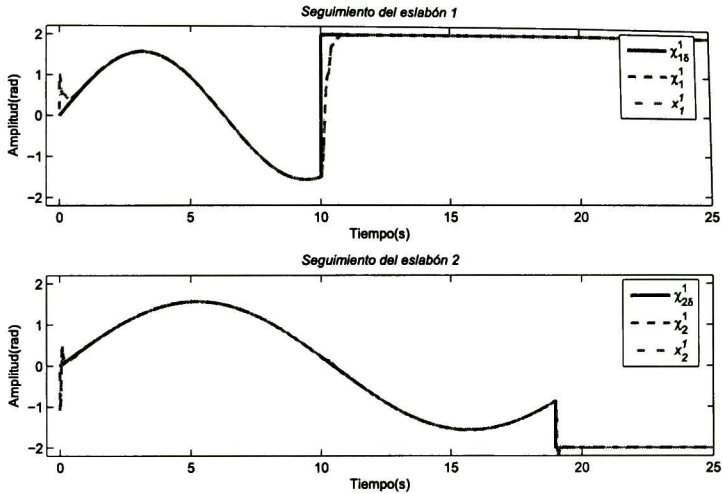


Figura 4.14: Seguimiento de trayectoria de la posición (MN3).

En la Figura 4.15 se muestra la gráfica del error tanto para posiciones como para velocidades de ambos eslabones.

En la Figura 4.16 se muestran las variedades deslizantes S_1 y S_2 .

En la Figura 4.17 se muestran las entradas de control u_1 y u_2 .

Los resultados son bastante aceptables al cumplirse con el objetivo (seguimiento de trayectoria), además de que el modelo neuronal es relativamente simple al requerirse un dos pesos adaptables por cada estado a identificar; aunque este modelo neuronal es el de mayor número de pesos adaptables en la red (en comparación con $MN1$ y $MN2$), computacionalmente no es muy pesado.

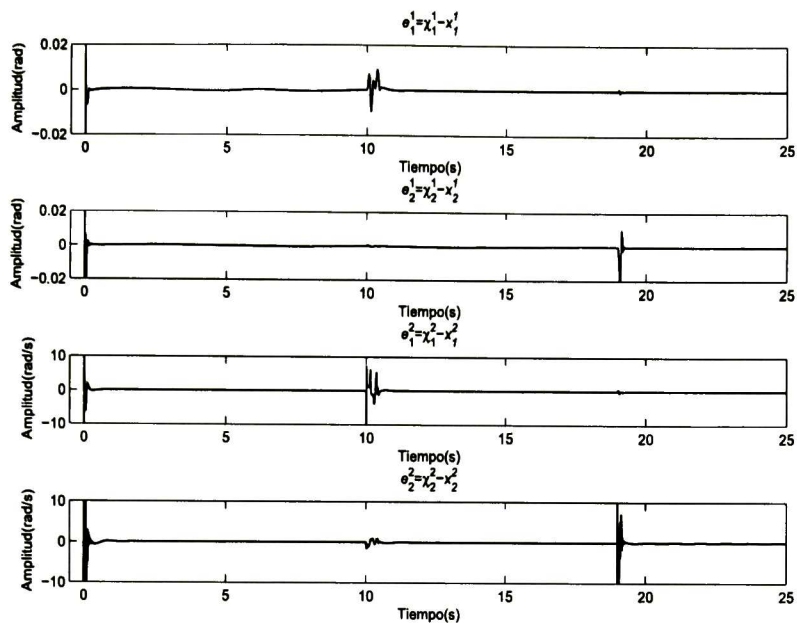


Figura 4.15: *Error de posición y velocidad (MN3).*

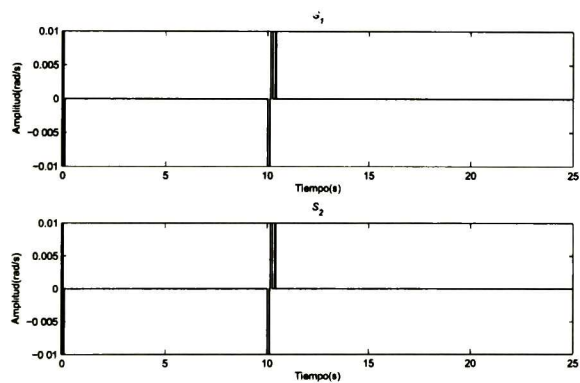


Figura 4.16: *Variedades deslizantes (MN3).*

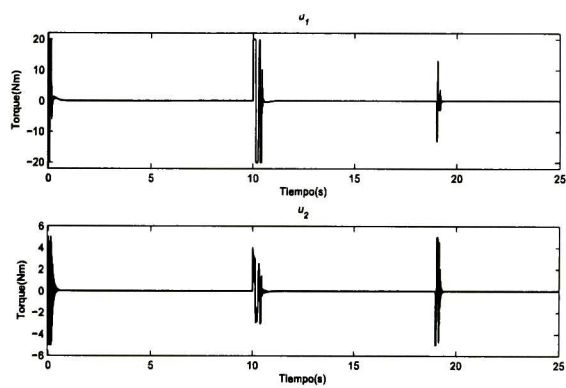


Figura 4.17: Entradas de control (MN3).

Capítulo 5

Implementación en tiempo real

En este capítulo se presentan en detalle los componentes del sistema de control del robot (equipo físico), además de las precauciones que han de tomarse al manejar el equipo para su buen funcionamiento. Es importante la descripción de cada uno de los componentes del equipo a fin de que se puedan realizar las modificaciones para alguna mejora posible o simplemente mantenimiento preventivo correctivo. También se presentan los resultados experimentales en tiempo real.

La Figura 5.1 muestra el sistema completo, mientras que la Figura 5.2 muestra la ubicación de los componentes del robot manipulador (ambos eslabones), los actuadores y sensores.

El equipo se compone de cuatro bloques principales:

Cómputo.

- Interfaz y adquisición de datos.

Fuente de poder y etapa de potencia.

Actuadores, codificadores (sensores de posición) y planta (Robot).

5.1. Cómputo, interfaz y adquisición de datos

En esta sección se considera el *hardware* (componentes físicos). Éste está compuesto por una computadora personal (PC) *HP*[®] (Pentium II), una tarjeta multifuncional de adquisición de datos DAQ PC-1200 (con bus PCI) de *National Instruments*[®], usada para la interfaz entre PC, etapa de potencia y codificadores.

Se incluye también un dispositivo lógico programable (PIC16F877 de *Microchip*[®]) que se encarga de contar los pulsos suministrados por los codificadores y convertirlos en una



Figura 5.1: Sistema físico completo del robot manipulador.

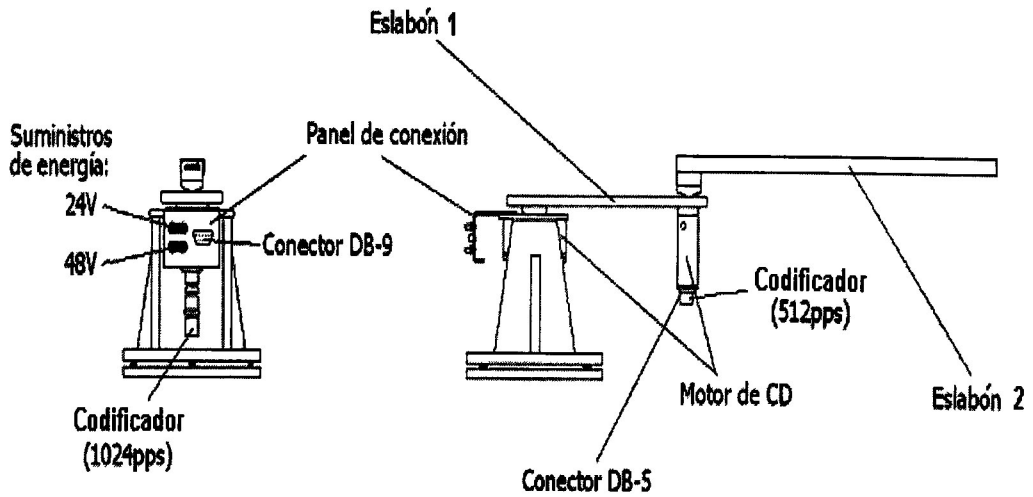


Figura 5.2: Ubicación de los componentes del robot manipulador

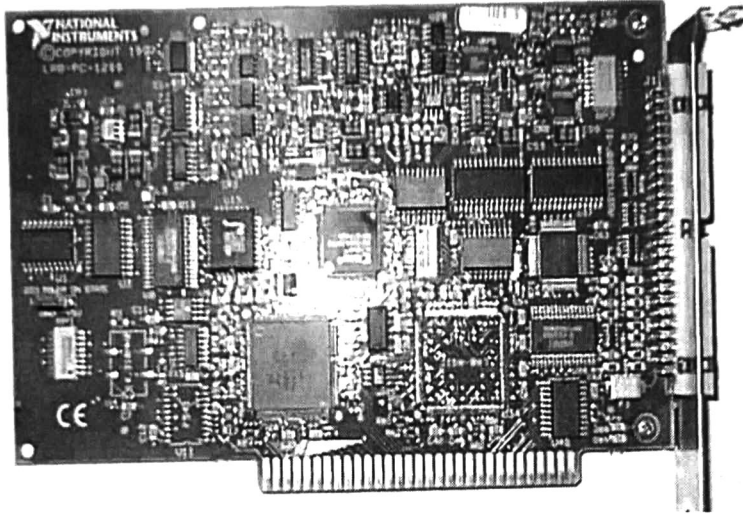


Figura 5.3: Tarjeta de adquisición de datos.

palabra digital para transmitirla por algún puerto digital de la tarjeta de adquisición de datos. La tarjeta (PC-1200) cuenta con tres puertos digitales de entrada y/o salida de datos de 8 bits cada uno, cuenta con dos ADC (Convertidor de Analógico a Digital) de hasta 12 bits de resolución y un grupo de contadores de hasta 1 MHz como frecuencia de oscilación. La Figura 5.3 muestra la tarjeta de adquisición de datos y la Figura 5.4 un diagrama a bloques de los recursos de la misma. Consultar el manual de usuario de la tarjeta para mayor detalle.

También se considera la parte del *software*, que es donde se realiza la programación de los cálculos necesarios para llevar a cabo la identificación y control. El programa implementado está en lenguaje C++.

5.2. Fuente de poder y etapa de potencia

Se utiliza una fuente de poder para alimentar la etapa de potencia (EP), con entrada de corriente alterna (127 AC) y dos salidas de corriente continua: una con 48 Volts para la EP del motor 1 y 24 Volts para la EP del motor 2.

La EP (fabricada por *Servo Systems*[®], modelo 12A8) son los dispositivos que separan la etapa de alta potencia (corrientes y voltajes a los motores) y las señales de baja potencia (las señales de control dadas por la tarjeta de adquisición de datos). La EP, adicionalmente

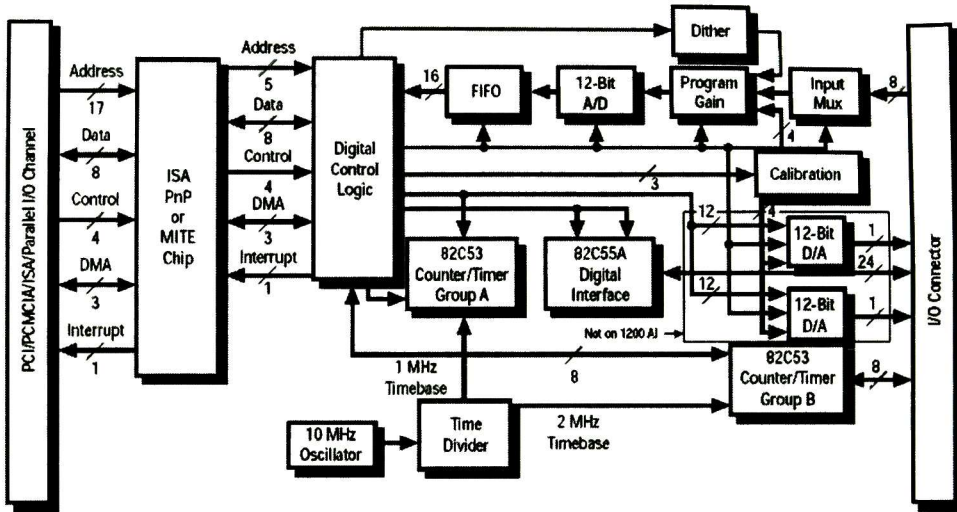


Figura 5.4: Diagrama a bloques de la tarjeta de adquisición de datos.

cuenta con una terminal que habilita o deshabilita el suministro de energía a los motores a través de un botón de seguridad (push-button); éste puede ser utilizado incluso como paro de emergencia; además, cuenta con la configuración de controlar por: *modo corriente* y *modo voltaje*.

En la configuración de modo corriente, la EP se encarga de suministrar la cantidad de corriente que se le ordena sin importar el voltaje que tenga que dar a la salida en la carga (motor en este caso); en la configuración modo voltaje, se suministra el voltaje que se le ordena sin importar la corriente en la carga que tenga que generarse en la EP. Para mayor información consultar las hojas de datos del fabricante. La Figura 5.5 muestra la etapa de potencia instalada.

5.3. Actuadores, codificadores y planta (Robot)

Los actuadores son los motores utilizados para generar el movimiento de los eslabones del brazo manipulador y son mostrados en la Figura 5.6 y Figura 5.7. El primero de ellos es un motor *Yaskawa®*, modelo *P09S3* mientras el segundo es de *Servo Systems®*. En la parte inferior de los motores se encuentran los codificadores para medir la posición del eslabón respectivo. En el motor 1 se encuentra un codificador (*Dynapar®*, series *E14*) de 1024 PPR (Pulsos Por Revolución) y un codificador (*HP®*, *Heds-5640*) de 512 PPR para el motor 2. La planta por su parte, es la unión de los dos eslabones en serie (brazo manipulador de dos

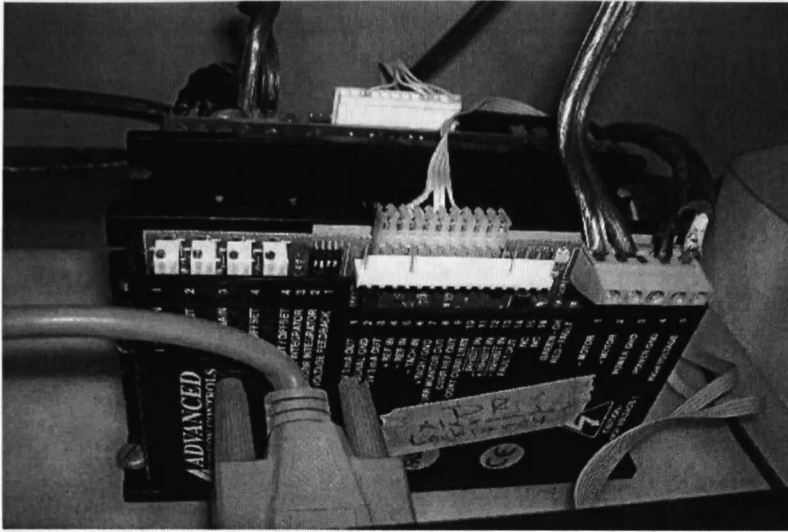


Figura 5.5: Etapa de potencia para los motores del brazo manipulador.

DOF).

La Figura 5.8 muestra un esquema del flujo de las señales de información (datos) involucradas para realizar la tarea de control del brazo manipulador cuando éste se encuentra en operación.

5.4. Instrucciones de uso

Para evitar posibles accidentes al operar el robot se dan las siguientes consideraciones y precauciones:

Verificar que las conexiones estén en perfecto estado, no estén en corto circuito y estén conectadas con la polaridad respectiva entre los elementos del sistema.

Verifique que el botón de seguridad **NO** esté presionado. Éste ha de oprimirse *sólo* cuando el sistema esté bajo el control diseñado por el usuario y que además el control funcione correctamente.

- Cuando se pruebe un algoritmo de control tener en cuenta que, de no funcionar correctamente el algoritmo de control, entonces alguno de los eslabones puede girar a grandes velocidades; por lo tanto, si algún equipo cercano o personas están dentro del alcance

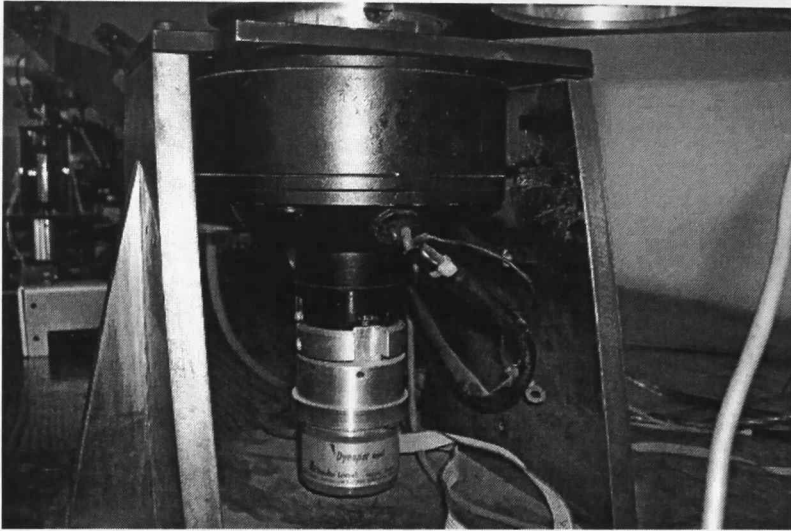


Figura 5.6: *Motor 1 que controla el eslabón 1.*

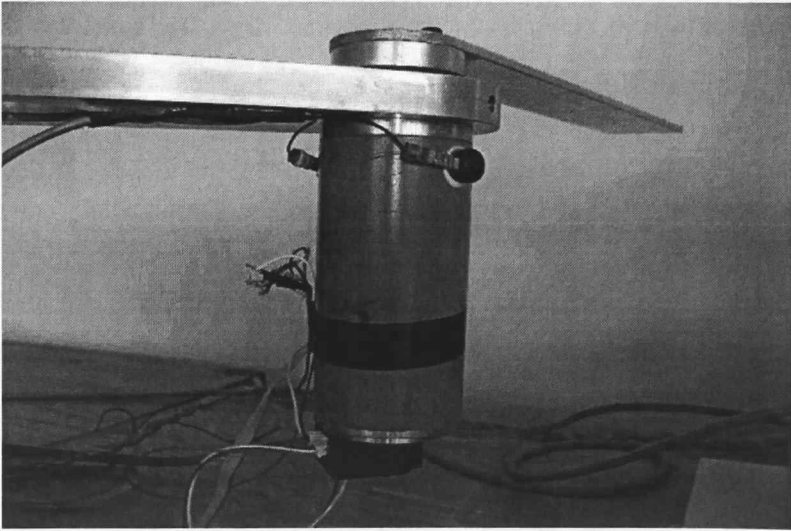


Figura 5.7: *Motor 2 que controla el eslabón 2.*

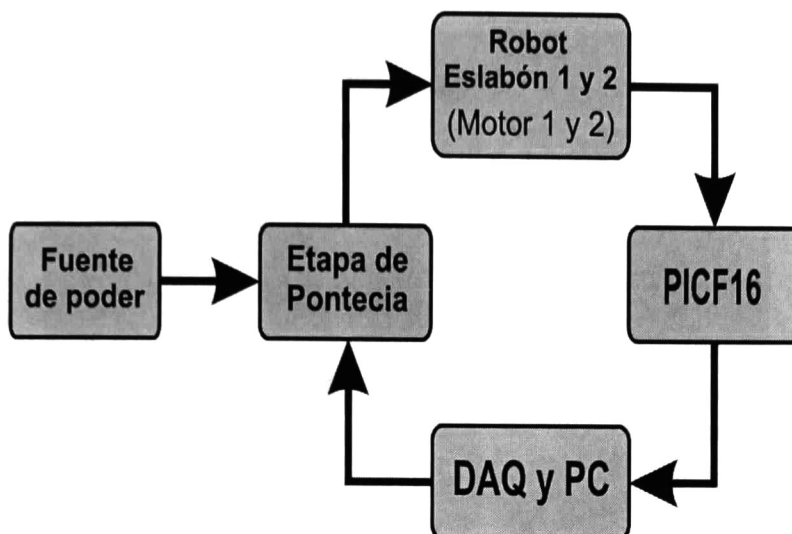


Figura 5.8: Flujo de información.

del brazo (considerar la longitud de la suma de los dos eslabones), éste puede resultar dañado.

- Una vez energizado el equipo, para evitar lesiones, no tocar los dispositivos que manejan las señales de alta potencia. De la misma forma evite tocar las terminales de señales y dispositivos de baja potencia para prevenir un posible daño al equipo producidas por cargas electrostáticas.
- Asegúrese de desenergizar el equipo una vez terminadas las pruebas.

5.5. Evaluación experimental

En esta sección se lleva a cabo la implementación en tiempo real de los modelos neuronales descritos en el Capítulo 4, y son utilizados para la identificación-control descentralizada del robot manipulador de dos grados de libertad con uniones revolutivas.

La implementación es llevada a cabo en el robot planar rígido de dos grados de libertad, disponible en el *Laboratorio de Control Automático (CINVESTAV GUADALAJARA)*. Como se describió anteriormente, este robot cuenta con: la etapa de potencia para los dos motores de corriente continua (CC), brazo mecánico con los dos eslabones, computadora personal con tarjeta de adquisición de datos. El algoritmo de control está programado en lenguaje C++.

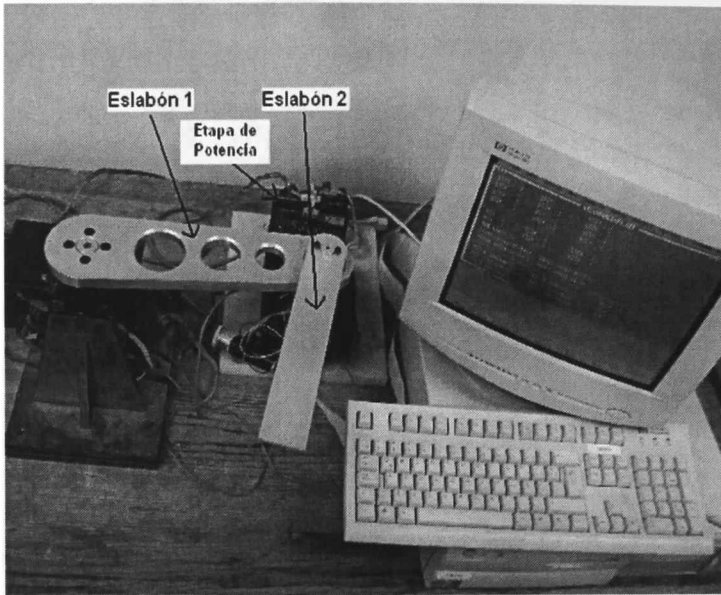


Figura 5.9: Robot planar de dos DOF.

5.5.1. Resultados en tiempo real

El robot físico, en el cual se implementó la identificación y control se muestra en la Figura 5.9. El periodo de muestreo utilizado en estos experimentos es de $T = 0,008 \text{ s}$.

Modelo neuronal 1 (MN1)

La ley de control para este modelo es dada en (3.14) y (4.9)-(4.11).

En la Figura 5.10 se muestra la gráfica del seguimiento de trayectoria de la posición para ambos eslabones, la cual contiene la señal de referencia, la señal de la planta y la señal de identificación. Este esquema de red neuronal no es muy eficiente ya que las condiciones iniciales de posición deben de estar cercanas (± 5 grados para cada eslabón), de lo contrario la posición del robot tiene oscilaciones sobre la señal de referencia que incluso el sistema puede no converger. Además, la frecuencia de la referencia que se puede seguir no es muy alta (un ciclo cada 30 s, 0,033 Hz).

La Figura 5.11 muestra la gráfica del error tanto para posiciones como para velocidades de ambos eslabones.

En la Figura 5.12 se muestran las variedades deslizantes S_1 y S_2 .

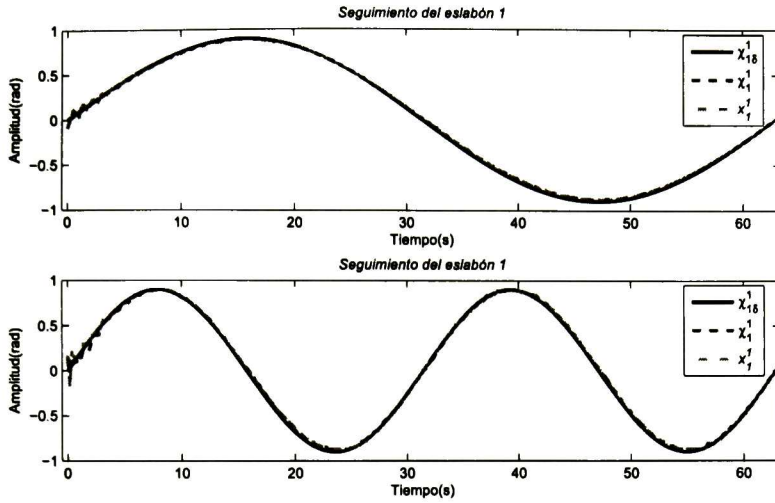


Figura 5.10: Seguimiento de trayectoria de la posición en tiempo real (MN1).

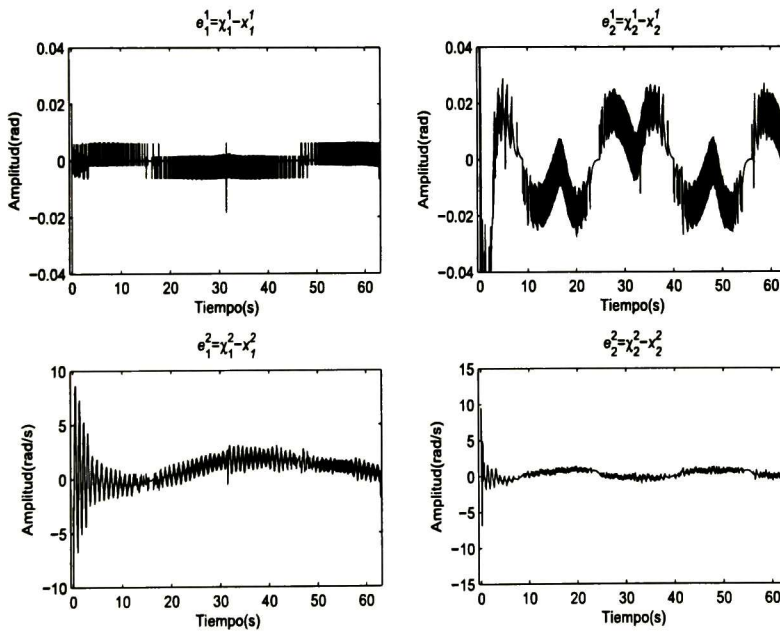


Figura 5.11: Error de posición y velocidad en tiempo real (MN1).

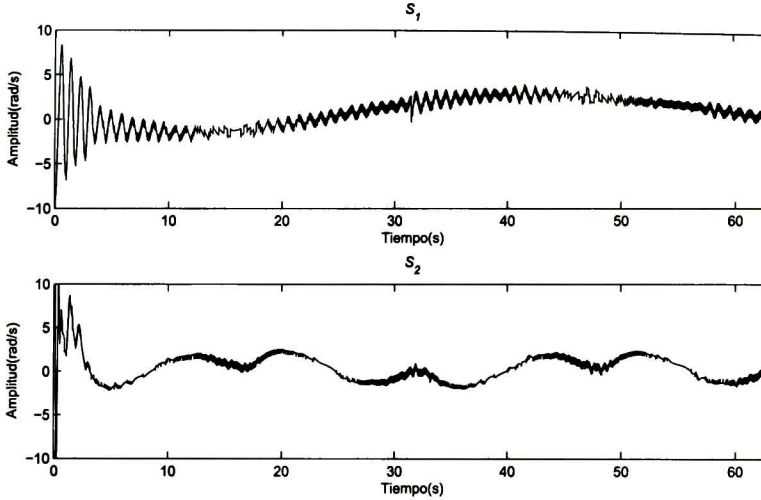


Figura 5.12: *Variedades deslizantes en tiempo real (MN1).*

La Figura 5.13 muestra las entradas de control u_1 y u_2 .

Los resultados son aceptables, pues se cumple con el objetivo (seguimiento de trayectoria), además de que el modelo neuronal es bastante simple al requerirse un solo peso adaptable por cada estado a identificar; computacionalmente esto es muy eficiente, pues se requiere de un número menor de operaciones.

Modelo neuronal 2 (MN2)

La ley de control para este modelo es dada en (3.14) y (4.12)-(4.14).

En la Figura 5.14 se muestra la gráfica del seguimiento de trayectoria de la posición para ambos eslabones, la cual contiene la señal de referencia, la señal de la planta y la señal de identificación. En este esquema se puede iniciar con condiciones iniciales de posición, entre la planta y la red neuronal, alrededor de ± 15 grados para cada eslabón. Es un algoritmo eficiente computacionalmente al requerirse estar entrenando sólo seis pesos (uno por posición y dos para velocidad, esto para cada eslabón).

La Figura 5.15 muestra la gráfica del error tanto para posiciones como para velocidades de ambos eslabones.

En la Figura 5.16 se muestran las variedades deslizantes S_1 y S_2 .

La Figura 5.17 muestra las entradas de control u_1 y u_2 .

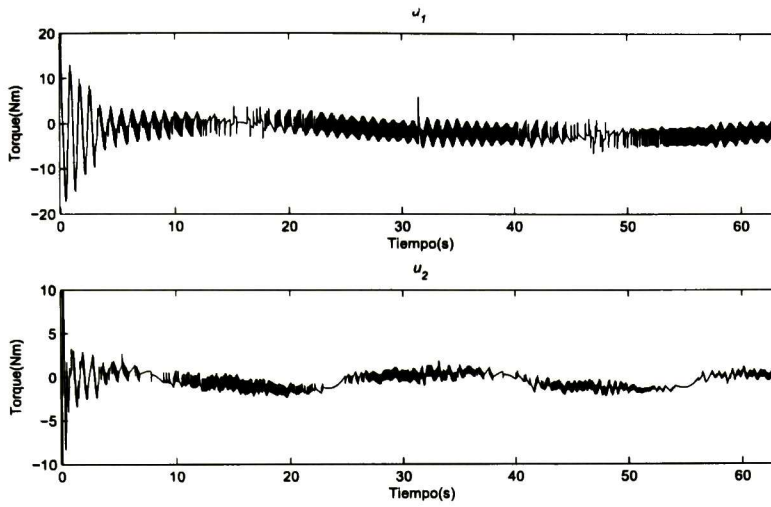


Figura 5.13: Entradas de control en tiempo real (MN1).

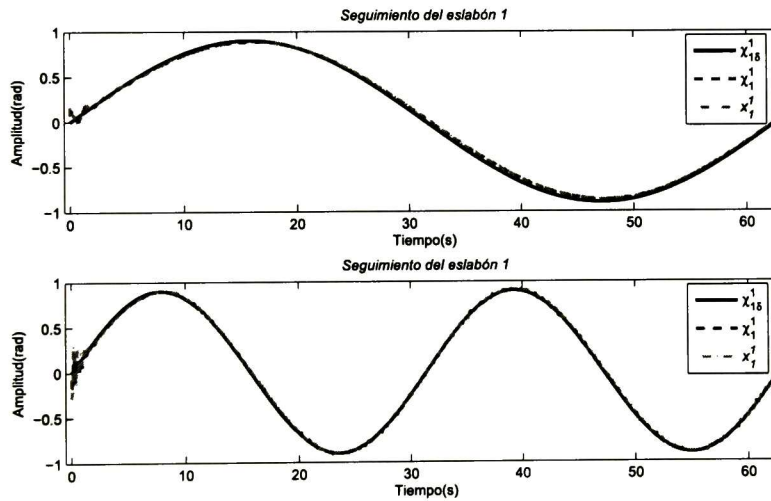


Figura 5.14: Seguimiento de trayectoria de la posición en tiempo real (MN2).

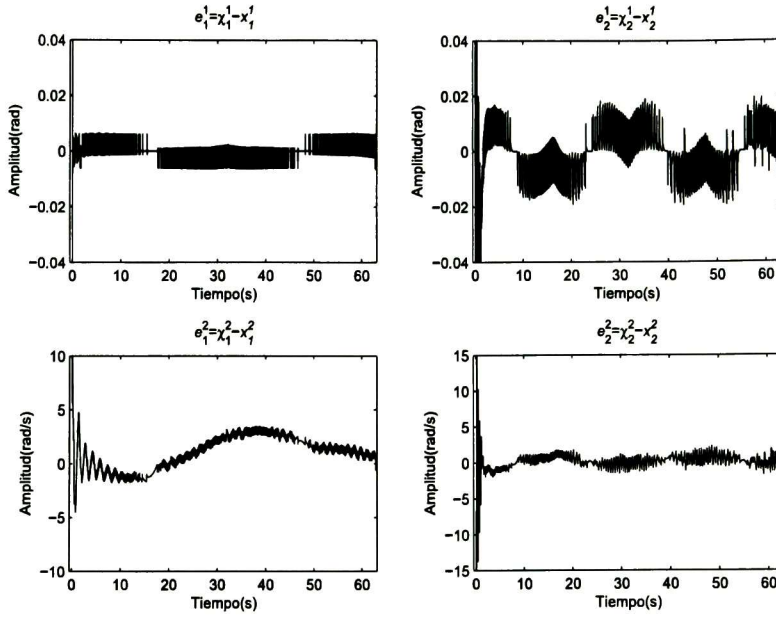


Figura 5.15: Error de posición y velocidad en tiempo real (MN2).

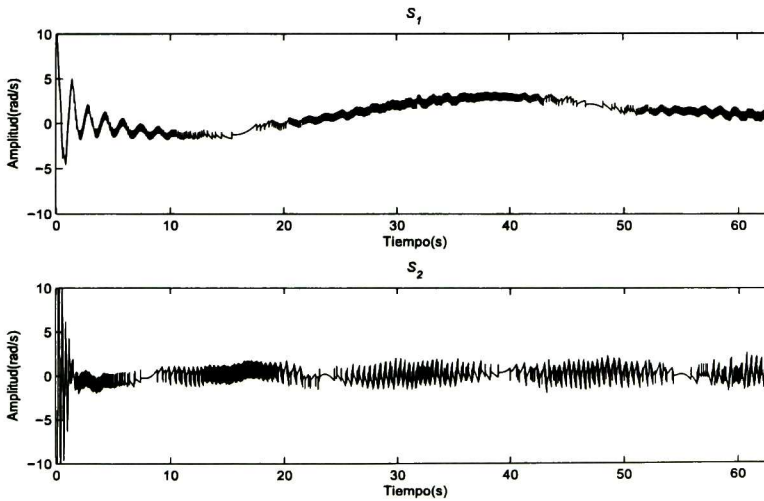


Figura 5.16: Variedades deslizantes en tiempo real (MN2).

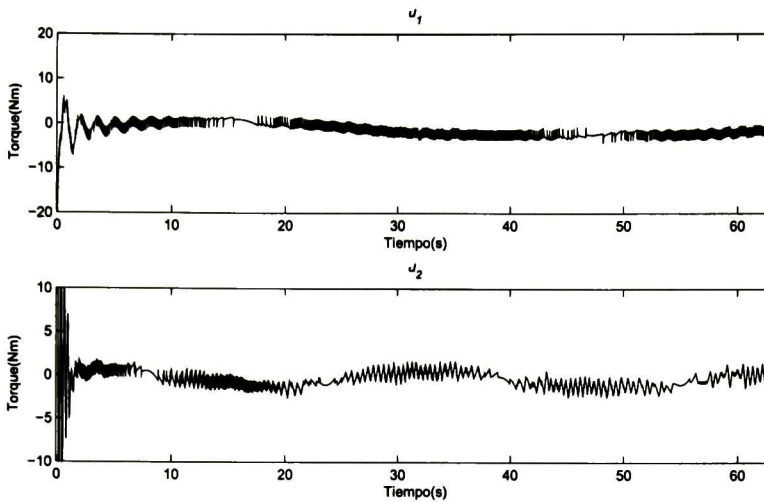


Figura 5.17: Entradas de control en tiempo real (MN2).

Los resultados son bastante aceptables, pues además de poder realizar seguimiento, se puede controlar a una frecuencia considerable (1 ciclo cada 16 s).

Modelo neuronal 3 (MN3)

El diseño del control se describe en el Capítulo 3. La ley de control es dada en (3.14) y (4.8).

En la Figura 5.18 se muestra la gráfica del seguimiento de trayectoria de la posición para ambos eslabones, la cual contiene la señal de referencia, la señal de la planta y la señal de identificación. En este esquema se puede iniciar con condiciones iniciales de posición, entre la planta y la red neuronal, alrededor de ± 30 grados para cada eslabón. Computacionalmente este algoritmo es relativamente más demandante al necesitar ajustar dos pesos por estado.

La Figura 5.19 muestra la gráfica del error tanto para posiciones como para velocidades de ambos eslabones.

En la Figura 5.20 se muestran las variedades deslizantes S_1 y S_2 .

La Figura 5.21 muestra las entradas de control u_1 y u_2 .

Los resultados son bastante aceptables, pues además de poder realizar seguimiento, se puede controlar a una frecuencia considerable (1 ciclo cada 16 s). En este algoritmo se realizó un experimento (aunque no son presentadas en esta tesis) en el cual se pudo realizar

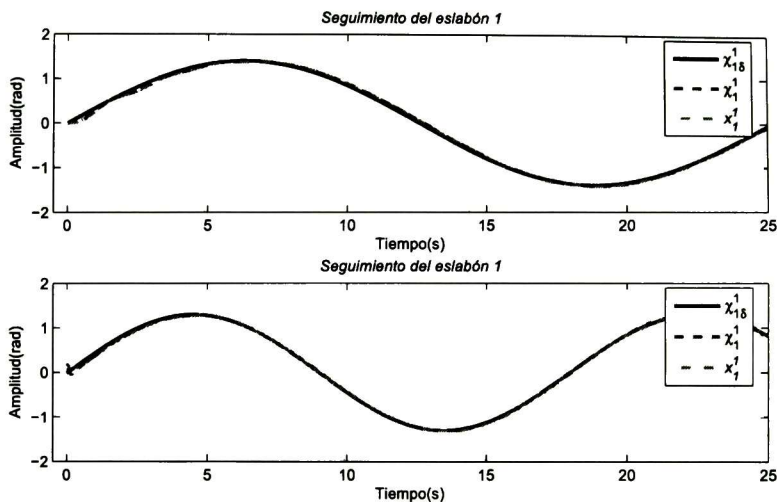


Figura 5.18: Seguimiento de trayectoria de la posición en tiempo real (MN3).

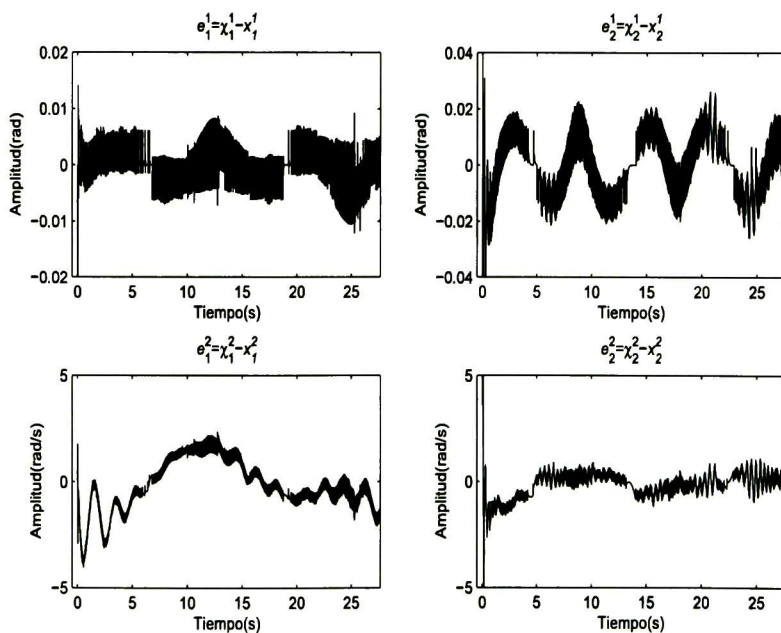


Figura 5.19: Error de posición y velocidad en tiempo real (MN3).

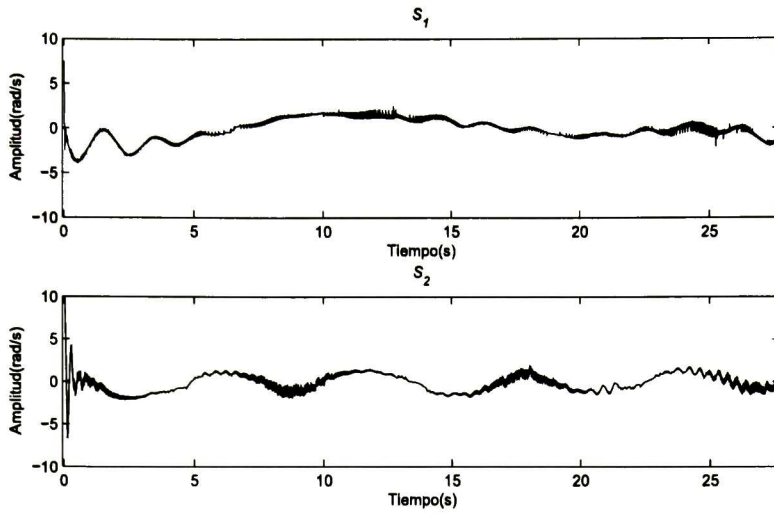


Figura 5.20: Variedades deslizantes en tiempo real (MN3).

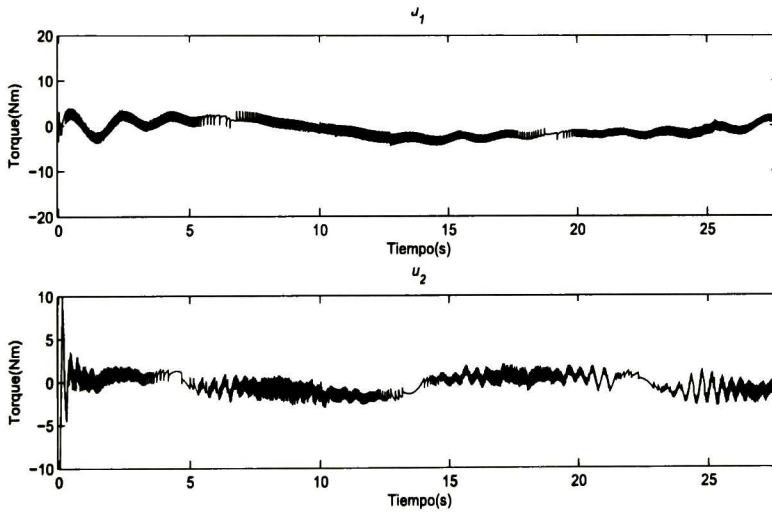


Figura 5.21: Entradas de control en tiempo real (MN3).

Tabla 5.1: Parámetros utilizados en tiempo real

PARÁMETRO	VALOR	PARÁMETRO	VALOR
$\chi_{1,0}^1$	-0.3 rad	$\chi_{2,0}^1$	0.4 rad
$\chi_{1,0}^2$	0 rad/s	$\chi_{2,0}^2$	0 rad/s
$x_{i,0}^1$	0 rad	$x_{i,0}^2$	0 rad/s
w_1^1	0.008	w_2^1	0.008
w_1^2	0.7	w_2^2	0.5
k_1^1	0.73	k_2^1	0.75
η_i^1	0.8	η_i^2	0.85
u_{10}	40 Nm	u_{20}	48 Nm

control de posición para los eslabones entre: ± 90 grados para el primero y ± 115 grados para el segundo¹

Como experimento adicional, se probó en este algoritmo (MN3) un control de posición en el que: las condiciones iniciales son diferentes, se inicia con el seguimiento de una señal senoidal (sobre ella se aplica una perturbación externa² en el eslabón 2 y posteriormente una sobre el eslabón 1), se conmuta a un seguimiento de señal constante en tiempo diferente para cada eslabón; dado que este esquema neuronal fue el que presento mejor desempeño, la Tabla 5.1 presenta algunas de las condiciones iniciales y parámetros de la red neuronal utilizadas en tiempo real en el experimento. La Figura 5.22 muestra los resultados de seguimiento.

5.5.2. Algoritmo para la planeación de trayectorias

Los esquemas presentados en la sección anterior realizan seguimiento sobre referencias constantes o sinusoidales. Sin embargo, en aplicaciones de robots manipuladores comúnmente se tienen tareas con referencias distintas a las mencionadas. Por ejemplo, para un robot manipulador industrial, que tiene que realizar un trabajo de corte (en línea recta) con su efector final sobre una superficie metálica, se debe diseñar una trayectoria ya sea en función de los ángulos de sus articulaciones o en función de las coordenadas en el espacio (x, y, z) , tal que el efector final se mueva sobre dicha trayectoria; diseñar trayectorias en el espacio es el más empleado y por ello es el que se expone en esta sección.

En el Apéndice B y Apéndice C se dan los detalles para la planeación de la trayectoria, así como la solución de las referencias (posiciones angulares calculadas por la cinemática inversa), las cuales han de ser aplicadas al algoritmo de control para que se realice dicho

¹ Estas son las cotas máximas con las que se experimento, aunque pueden ser mayores. Éstas vendrán limitadas por la construcción física del robot.

² Se aplico una fuerza manualmente sobre el eslabón correspondiente.

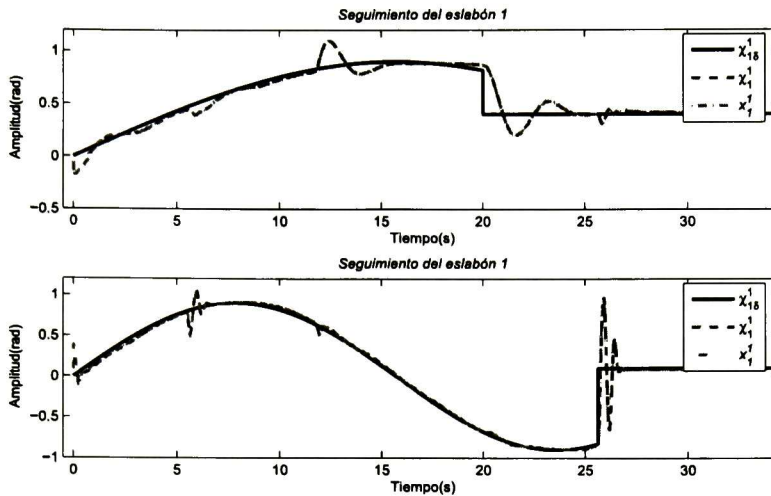


Figura 5.22: Seguimiento de trayectoria de la posición en tiempo real (MN3).

movimiento (trayectoria).

Ahora se describen los pasos principales del algoritmo para la generación de trayectorias en un robot manipulador:

1. Determinar el polinomio que describe la trayectoria $X_d(t)$ a partir del punto inicial y final de la trayectoria, así como velocidad y aceleración en los tiempos t_0 y t_f .
2. Encontrar la velocidad en el tiempo t_k al derivar el polinomio $X_d(t)$ dado en el Apéndice B.
3. Encontrar \dot{q}_k a partir de la cinemática inversa del robot como se describe en el Apéndice C.
4. Encontrar q_k por medio de un integrador.
5. Los valores de q_k son los utilizados de referencia para el algoritmo de control por bloques diseñado en el Capítulo 3.
6. Calcular el valor de $X(q)$ actual por medio de la cinemática directa del robot y retroalimentarlo en el cálculo de \dot{q}_k .

Cabe señalar que en caso de que sea una trayectoria compuesta por tramos o puntos intermedios, basta con hacer que el punto final sea el inicial del siguiente tramo; de esta

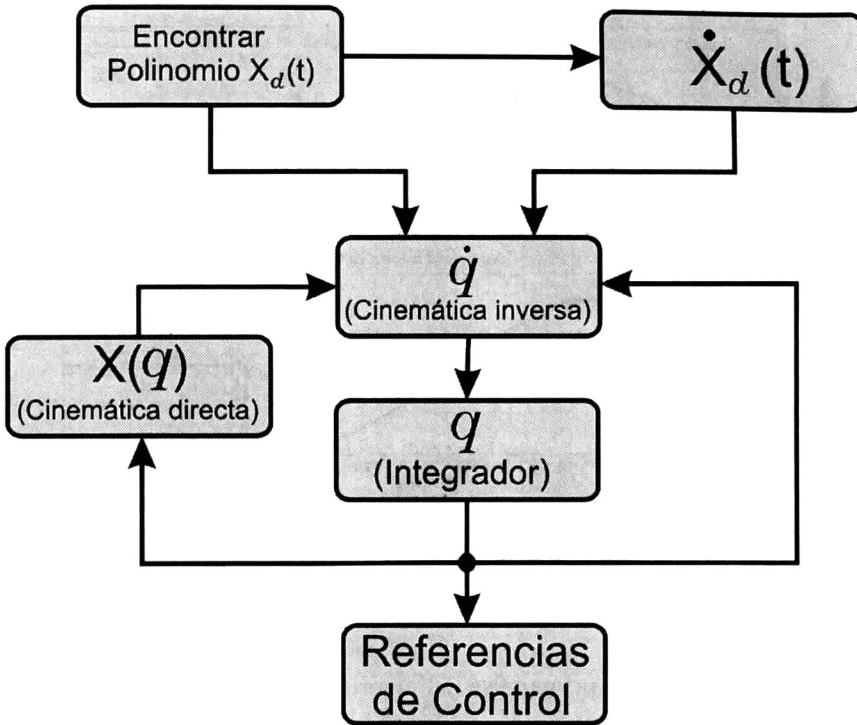


Figura 5.23: Diagrama a bloques para la generación de trayectoria.

forma se repite el algoritmo hasta que se recorran los n tramos que conforman la trayectoria total.

La Figura 5.23 muestra un diagrama a bloques de los pasos antes mencionados.

Ahora se aplica el algoritmo descrito en la Figura 5.23 para el robot planar de dos DOF, considerando que este robot sólo puede realizar un movimiento en coordenadas (x, y) , pues por su construcción (uniones revolutivas) no puede realizar un movimiento en el eje z . En base al Apéndice B y Apéndice C, se tiene que los parámetros son:

$$X(q) = \begin{bmatrix} x(q) \\ y(q) \end{bmatrix} = \begin{bmatrix} l_1 \cos(q_1) + l_2 \cos(q_1 + q_2) \\ l_1 \sin(q_1) + l_2 \sin(q_1 + q_2) \end{bmatrix} \quad (5.1)$$

$$J^{-1}(q) = \frac{1}{l_1 l_2 \sin(q_2)} \begin{bmatrix} l_2 \cos(q_1 + q_2) & l_2 \sin(q_1 + q_2) \\ -l_1 \cos(q_1) - l_2 \cos(q_1 + q_2) & -l_1 \sin(q_1) - l_2 \sin(q_1 + q_2) \end{bmatrix} \quad (5.2)$$

En la Figura 5.24 se muestra la planeación de una trayectoria (un triángulo) a la cual el

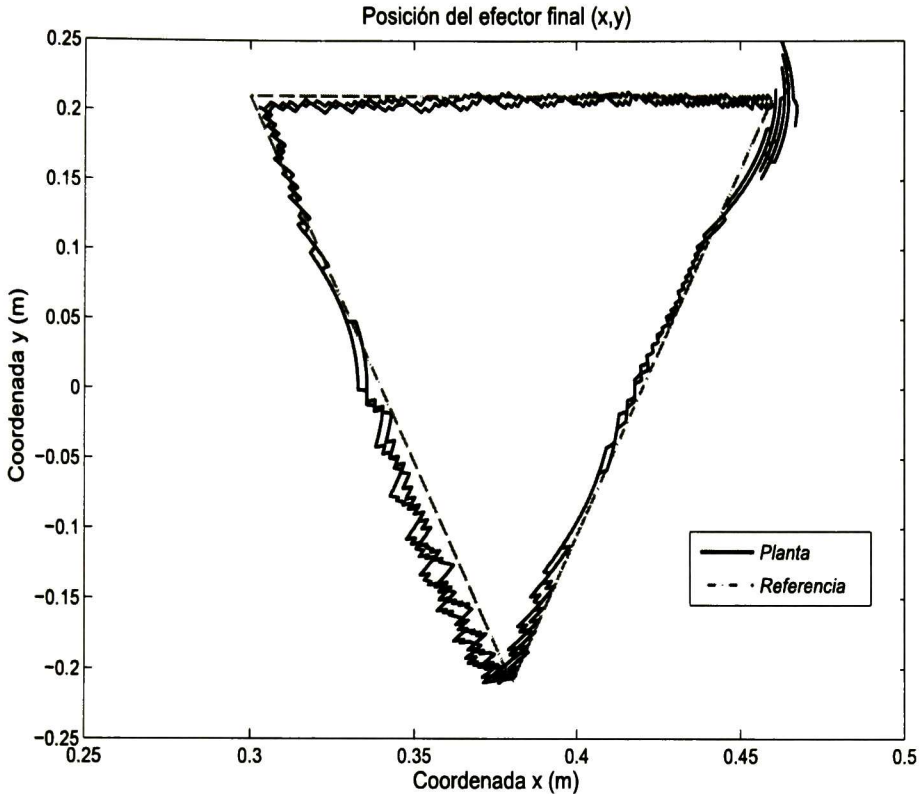


Figura 5.24: Posición de la referencia y del efector final del robot.

robot (efector final) hace seguimiento. La trayectoria que forma el triángulo está descompuesta en tres tramos: el primero va de $(0.46, 0.21)$ a $(0.38, -0.21)$, el segundo de $(0.38, -0.21)$ a $(0.30, 0.21)$ y el tercero de $(0.30, 0.21)$ a $(0.46, 0.21)$; como se puede apreciar, la parte final del último tramo de la trayectoria es el inicial de la misma, así, repitiendo n veces la trayectoria se estará siguiendo la trayectoria de un triángulo n veces. La Figura 5.25 muestra la posición que debe de recorrer el efector final en la coordenada $(x(t), y(t))$. La Figura 5.26 muestra la posición angular que debe tener cada eslabón para poder realizar la trayectoria deseada.

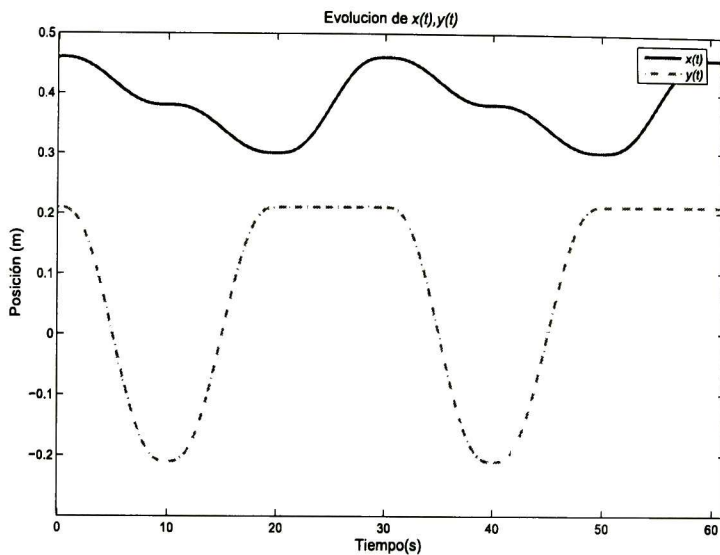


Figura 5.25: *Evolución de las coordenadas.*

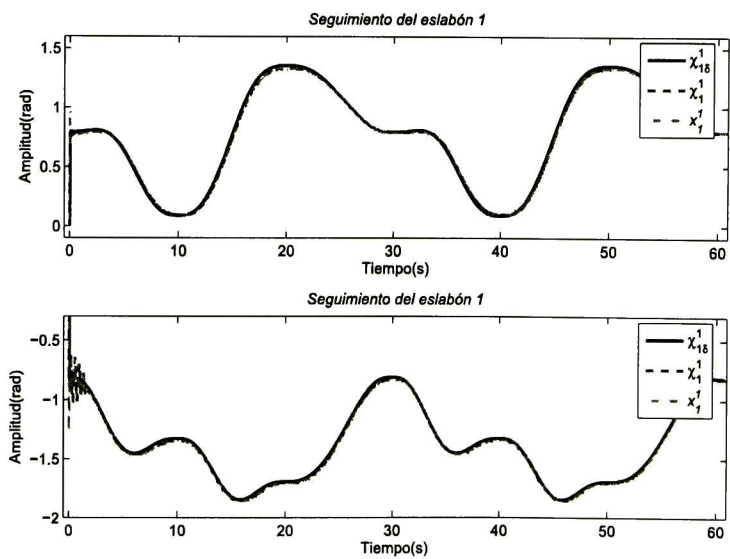


Figura 5.26: *Referencias angulares para realizar la trayectoria de un triángulo.*

Capítulo 6

Conclusiones y trabajo futuro

6.1. Conclusiones

Dada la complejidad para obtener el modelo dinámico de un robot manipulador, en particular la obtención de los parámetros que forman parte del modelo, se diseñó una alternativa eficaz para modelar de forma artificial el comportamiento dinámico del sistema (robot). Para ello, se hizo uso de las redes neuronales recurrentes, las cuales tienen la capacidad de modelar sistemas no lineales complejos y de gran escala.

Los resultados obtenidos en el desarrollo de esta tesis permiten concluir que los sistemas que describen el comportamiento de los robots manipuladores, completamente actuados (un actuador por grado de libertad), pueden ser tratados como una descomposición de subsistemas de menor dimensión, en donde cada uno de los subsistemas corresponde a un eslabón o grado de libertad; de esta manera se tendrán tantos subsistemas como grados de libertad constituyan al robot. A este enfoque se le conoce como sistema descentralizado, en el cual cada subsistema puede ser tratado de forma independiente. De lo anterior, es natural de igual forma, diseñar un sistema neuronal descentralizado.

Se diseñaron e implementaron diferentes estructuras neuronales descentralizadas (Modelo neuronal 1, 2 y 3) con el fin de modelar la dinámica de cada grado de libertad del robot; la topología usada en las redes neuronales presentadas es la serie-paralelo.

La parte principal de las redes neuronales es la forma en la que éstas aprenden (por medio del valor de sus pesos o parámetros de ajuste); en este sentido, se usó el filtro de Kalman extendido como algoritmo de entrenamiento, dando excelentes resultados para que la red neuronal lograra la identificación (modelar) adecuadamente la dinámica del robot. Otra característica importante del filtro de Kalman es que es un algoritmo eficiente computacionalmente, permitiendo realizar los ajustes de los pesos en la red neuronal en línea. Esto último permitió realizar una implementación en tiempo real sin requerirse de equipo con velocidades de procesamiento muy alto.

Se diseñó el esquema de control (usando la técnica de control por bloques y modos deslizantes discretos) basado en el modelo neuronal para poder controlar la posición de cada eslabón y realizar seguimiento de trayectorias; además de que por medio de la ley de control obtenida, se garantiza la estabilidad del sistema completo en lazo cerrado.

Tanto el identificador (red neuronal) como el controlador fueron tratados en tiempo discreto, lo cual permite realizar de forma inmediata una implementación del algoritmo de control en algún equipo digital (PC, microcontrolador, etc.) disponible para la programación del algoritmo. Los algoritmos de identificación y control propuestos fueron implementados en una computadora personal bajo el lenguaje de C++.

Los últimos algoritmos de control diseñados e implementados (modelo neuronal 2 y 3) presentaron excelentes comportamientos tanto en simulación como en tiempo real, al permitir tener condiciones iniciales diferentes entre la planta y la red neuronal, lo cual es un caso real; además de ser algoritmos de control robusto ante perturbaciones internas y externas.

El enfoque descentralizado es muy atractivo, pues permite tener un conjunto de controladores locales trabajando en paralelo (un controlador por eslabón o grado de libertad), lo cual computacionalmente es muy eficiente.

6.2. Trabajo futuro

- Implementar los algoritmos de control descentralizado en un robot humanoide.
- En la parte de control, implementar modos deslizantes de alto orden tomando en cuenta la dinámica no modelada de los actuadores.
- Eliminar los codificadores como sensores de posición y utilizar la visión artificial (por medio de cámaras) para medir la posición angular de los eslabones.
- Diseñar e implementar un observador neuronal para las velocidades de los eslabones (usando solamente sensores de posición).

Apéndice A

Sintonización de los parámetros para el EKF

El algoritmo EKF para el entrenamiento de RNN requiere dar valores a ciertos parámetros. En este apéndice se dan algunas recomendaciones y consideraciones para la sintonización del EKF.

- Los valores iniciales de los pesos $w(0)$ deben de ser pequeños y pueden tomar valores arbitrarios.
- $P(0)$ de igual forma toma un valor inicial pequeño (del orden de 0.01-1).
- Usualmente se requieren valores pequeños de la razón de aprendizaje η_i para lograr un buen comportamiento en el entrenamiento. Se propone un rango para este parámetro entre 0.001-1.
- La matriz de covarianza del ruido del proceso Q está representado por una matriz diagonal qI con el factor de escala q , con un valor alrededor de 1000.

El valor de la matriz de covarianza del ruido en la medición R , usualmente toma valores en magnitud similar o mayor a Q .

También es posible ajustar (en línea) los parámetros de Q y R , empleando otro filtro de Kalman.

Es importante notar que:

1. Si la covarianza del error de medición se aproxima a cero, la ganancia de $K(k)$ otorga mayor peso a la innovación de la medición, es decir, se confía más en la medición:

$$\lim_{R(k) \rightarrow 0} K(k) = H(k)^{-1}$$

2. Si el estimador de la covarianza del error se aproxima a cero, la ganancia $K(k)$ otorga menor importancia a la innovación de la medición:

$$\lim_{P(k) \rightarrow 0} K(k) = 0$$

La elección apropiada de los valores de las matrices Q y R va de acuerdo al comportamiento que se desee tener: mejor seguimiento o ahorro de energía en la entrada de control. Si se está interesado en un mejor seguimiento de la trayectoria, se puede seleccionar un $\|Q\|$ alto, o reducir $\|R\|$. Por el contrario, si se desea un ahorro de la energía de control, se puede dar un valor bajo de $\|Q\|$ o incrementar $\|R\|$. Para más detalles se puede consultar [9] y [18].

Apéndice B

Generación de trayectorias

Este apéndice da una explicación general sobre el planeamiento de trayectorias en un espacio multidimensional, el cual describe el movimiento deseado del manipulador, para más detalles ver [4] y [5]. Así, por trayectoria se hace referencia a la historia en el tiempo de la posición, velocidad y aceleración para cada grado de libertad, de tal forma que se tenga un movimiento en el efector final¹ sobre una trayectoria específica en el espacio. Por otro lado, se debe de dar una interfaz adecuada entre el sistema y el usuario, para que este último pueda especificar de forma sencilla el movimiento deseado del manipulador (sólo especificar la posición y orientación del efector final), y dejar al sistema la deducción de los detalles de cálculo para dicha tarea.

En forma general, el problema puede ser visto en como diseñar el movimiento del manipulador para que éste pueda alcanzar una posición final a partir de una inicial. Dependiendo de la trayectoria que se desee seguir, algunas veces es necesario especificar el movimiento en forma más detallada y no simplemente establecer la configuración final (posición y orientación). Una forma de incluir dichos detalles en la descripción de la trayectoria, es dar una secuencia de puntos de ésta (puntos intermedios entre la posición inicial y final).

Otro aspecto también a considerar es que, se debe de especificar el tiempo para que se lleve a cabo la tarea de seguimiento de trayectoria completa, en donde usualmente se debe realizar de una forma suave. Para este propósito se diseñan funciones que tengan primera y segunda derivada, que es velocidad y aceleración respectivamente.

Existen diferentes tipos de funciones para la generación de trayectorias en espacio-tiempo como se muestra en la Figura B.1, en la cual se puede apreciar la evolución del movimiento de X^2 con respecto al tiempo t . Aquí se aborda una función de quinto orden para realizar dicho movimiento, ya que en esta función se puede especificar la posición, velocidad y aceleración en el punto inicial y final de la trayectoria (lo mismo aplica para puntos intermedios de una

¹ Por efector final se hace referencia a la parte final del último eslabón.

² X contiene las coordenadas (x, y, z) .

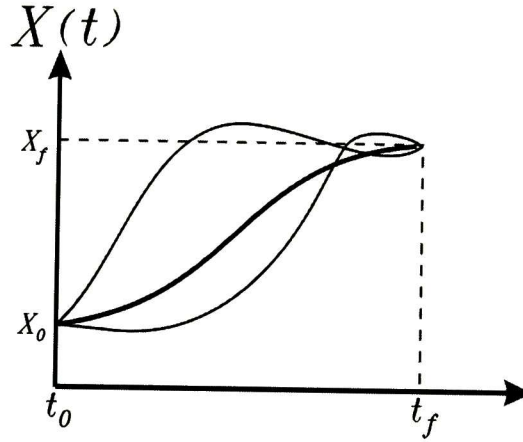


Figura B.1: *Formas de trayectorias.*

trayectoria), de esta forma el polinomio requerido es

$$X(t) = a_0 + a_1t + a_2t^2 + a_3t^3 + a_4t^4 + a_5t^5 \quad (\text{B.1})$$

donde las restricciones son dadas como

$$\begin{aligned} X_0 &= a_0 \\ X_f &= a_0 + a_1t_f + a_2t_f^2 + a_3t_f^3 + a_4t_f^4 + a_5t_f^5 \\ \dot{X}_0 &= a_1 \\ \dot{X}_f &= a_1 + 2a_2t_f + 3a_3t_f^2 + 4a_4t_f^3 + 5a_5t_f^4 \\ \ddot{X}_0 &= 2a_2 \\ \ddot{X}_f &= 2a_2 + 6a_3t_f + 12a_4t_f^2 + 20a_5t_f^3. \end{aligned} \quad (\text{B.2})$$

Las restricciones en (B.2) especifican un conjunto lineal de seis ecuaciones con seis incógni-

tas en donde la solución viene dada por

$$\begin{aligned}
 a_0 &= X_0 \\
 a_1 &= \dot{X}_0 \\
 a_2 &= \frac{\ddot{X}_0}{2} \\
 a_3 &= \frac{20X_f - 20X_0 - (8\dot{X}_f + 12\dot{X}_0)t_f - (3\ddot{X}_0 - \ddot{X}_f)t_f^2}{2t_f^3} \\
 a_4 &= \frac{30X_0 - 30X_f + (14\dot{X}_f + 16\dot{X}_0)t_f + (3\ddot{X}_0 - 2\ddot{X}_f)t_f^2}{2t_f^4} \\
 a_5 &= \frac{12X_f - 12X_0 - (6\dot{X}_f + 6\dot{X}_0)t_f - (\ddot{X}_0 - \ddot{X}_f)t_f^2}{2t_f^5}
 \end{aligned} \tag{B.3}$$

Es evidente que al diseñar las trayectorias del efector final se deben evitar ciertos problemas geométricos, restricciones físicas, entre otros, tales como: trazar una trayectoria dentro de la cual se tengan puntos inalcanzables, altas velocidades, restricciones dadas por la construcción del mecanismo del robot, etc.

Apéndice C

Cinemática

La cinemática es la ciencia del movimiento de un cuerpo sin tomar en cuenta las fuerzas que lo ocasionan [4] y [22]. Dentro de la cinemática se estudia la posición, velocidad, aceleración, y todas las derivadas de alto orden de las variables de posición (con respecto al tiempo o cualquier otra variable). El principal análisis de la cinemática en los robots manipuladores se divide en dos: *cinemática directa* y *cinemática inversa*.

En lo que respecta a la cinemática directa, el análisis que se desarrolla está en función de la posición y orientación de cada uno de los eslabones del manipulador en condiciones estáticas, esto por medio de marcos de referencia (sistema de coordenadas en cada eslabón) y así poder determinar la posición y orientación del efector final.

En la cinemática inversa se hace un análisis en el cual dada la posición y orientación del efector final, se calcula el conjunto de ángulos de las uniones de los eslabones para lograr el resultado deseado.

El enfoque que se da a la cinemática es con el fin de obtener información suficiente para llevar a cabo el diseño y realización de trayectorias, como se vio en el Apéndice B. Consultar [4] y [5] para más detalles sobre cinemática de robots manipuladores.

Se hace uso del vector

$$q = [q_1 \ q_2 \ \dots \ q_N]^T \quad (\text{C.1})$$

para representar el valor (grados o radianes según corresponda) de las N uniones (DOF) en el robot manipulador. Para representar el valor del efector final (posición de la parte final del último eslabón) en el espacio (x, y, z) , con respecto a algún marco de referencia (sistema de coordenadas), se usa el vector

$$e = [e_x \ e_y \ e_z]^T \quad (\text{C.2})$$

La función de cinemática directa calcula el punto en el espacio (con sistema de coordenadas usualmente en el origen) del efector final a partir de los ángulos en las uniones del

manipulador por medio de

$$e = f(q). \quad (\text{C.3})$$

Por su parte la cinemática inversa calcula el vector de ángulos en las uniones que causarían que el efector final alcance algún punto en el espacio por medio de la función inversa, es decir,

$$q = f^{-1}(e). \quad (\text{C.4})$$

Encontrar la función inversa f^{-1} es bastante complicado analíticamente, y a mayor grados de libertad, mayor será la complejidad. Es claro que dar solución a la cinemática directa (evaluación de $f(q)$) es mucho más sencillo que dar solución a la cinemática inversa (evaluación de $f^{-1}(e)$).

Para encontrar la solución de (C.4) existen varios métodos, el más usual es uso del Jacobiano $J(q)$, el cual es una forma multidimensional (matriz) de la derivada de $f(q)$ con respecto a q , es decir, $J(q) = df(q)/dq$.

Ahora, para resolver el problema de la cinemática inversa se hace uso de un algoritmo [6], el cual toma en cuenta las restricciones de los límites de velocidad permitidos en las uniones de los eslabones mientras se realiza el seguimiento de trayectoria del efector final; este algoritmo está diseñado para poder llevarse a cabo en tiempo real ya que requiere un ligero esfuerzo computacional.

Considere la cinemática diferencial

$$\dot{e} = J(q) \dot{q} \quad (\text{C.5})$$

y su inversión con la aproximación llamada *cinemática inversa en lazo cerrado* [6] en tiempo discreto

$$\dot{q}_k = J_{k-1}^{-1}(q) [\dot{e}_d(t_k) + A_p(e_d(t_k) - e_{k-1})] \quad (\text{C.6})$$

donde A_p es una matriz definida positiva constante, con valores apropiados para garantizar que $e_d(t_k) - e_{k-1}$ tienda a cero; \dot{q}_k es la velocidad angular, \dot{e}_d es la velocidad de la posición deseada. El objetivo es encontrar, en el tiempo t_k , un vector de velocidad \dot{q}_k , que satisfaga las restricciones de velocidad o al menos generar una velocidad en el efector final cuya dirección sea lo más cercana posible a la deseada. De esta forma, si al menos una de las velocidades de las uniones de los eslabones está fuera de su límite, entonces se debe calcular los coeficientes α_i dados por

$$\alpha_i = \frac{|\dot{q}_i|}{\dot{q}_{i,lim}} \quad i = 1, \dots, N. \quad (\text{C.7})$$

Entonces, definiendo $\alpha = \max_i(\alpha_i)$, la ecuación (C.6) queda escalada como

$$\dot{q}_k = \frac{1}{\alpha} J_{k-1}^{-1}(q) [\dot{e}_d(t_k) + A_p(e_d(t_k) - e_{k-1})]. \quad (\text{C.8})$$

Note que esta solución corresponde a una velocidad en el efector final la cual es paralela al vector $\dot{e}_d(t_k) + A_p(e_d(t_k) - e_{k-1})$, es decir, el efector final se está moviendo con la dirección deseada. Posteriormente para encontrar q_k como solución de (C.8), basta con un integrador (en tiempo discreto, ya que todos los cálculos están dados de esa forma). El integrador que se implementó tanto en simulación como en tiempo real es el *método directo de Euler*.

Para esta integración, el integrador en tiempo continuo $1/s$ es aproximado en tiempo discreto por $Ts/(z - 1)$. La expresión resultante para la salida del integrador discreto en el paso k es

$$q(k) = q(k - 1) + Ts \dot{q}_{k-1} \quad (C.9)$$

donde Ts es el periodo de muestreo. Es importante mencionar que en (C.9) se deben de establecer las condiciones iniciales para posición y velocidad angular.

Más integradores en tiempo discreto pueden ser encontrados en las bibliotecas y utilerías de *MATLAB*[®].

Bibliografía

- [1] Alanis A. Y. Entrenamiento de redes neuronales con el filtro de kalman. Master's thesis, Centro de Investigación y Estudios Avanzados del IPN, Guadalajara, México, Julio 2004.
- [2] Alanis A. Y. *Control Neuronal Discreto: Aplicación a Motores de Inducción*. PhD thesis, Centro de Investigación y Estudios Avanzados del IPN, Guadalajara, México, Septiembre 2007.
- [3] Anderson B. D. O. and Moore J. B. *Optimal Filtering*. Dover Publications, New York, 1979.
- [4] Craig J. J. *Introduction to Robotics: Mechanics and Control*. Addison-Wesley, USA, 1989.
- [5] Fu K. S., Gonzalez R. C., and Lee C. S. G. *Robótica: Control, detección, visión e inteligencia*. McGraw-Hill, España, 1998.
- [6] Gianluca A., Stefano C., and Giuseppe F. An algorithm for on-line inverse kinematics with path tracking capability under velocity and acceleration constraints. *IEEE Conference on Decision and Control*, pages 5079–5084, 2000.
- [7] Hassan K. *Nonlinear Systems*. Prentice Hall, New Jersey, 1996.
- [8] Haykin S. *Neural Networks: A comprehensive foundation*. Prentice Hall, New Jersey, USA, 1999.
- [9] Haykin S. *Kalman Filtering and Neural Networks*. Wiley, New Jersey, USA, 2001.
- [10] Huang S., Kok K. T., and Tong H. L. Decentralized control desing for large-scale systems with strong interconecctions using neural networks. *IEEE Transactions on Automatic Control*, 48(5):805–810, 2003.
- [11] Kazantzis N. and Kravaris C. Time-discretization of nonlinear control systems via taylor methods. *Computers & Chemical Engineering*, 23:763–784, 1999.

- [12] Loukianov A. G. Nonlinear block control with sliding modes. *Automation and Remote Control*, 57(7):916–933, 1997.
- [13] Marino R. and Tomei P. *Nonlinear Control Design*. Prentice Hall, Great Britain, 1995.
- [14] Pham D. T. and Liu X. *Neural Networks for Identification, Prediction and Control*. Springer-Verlag, Great Britain, 1995.
- [15] Rovothakis G. A. and Christodoulou M. A. *Adaptive Control with Recurrent High-Order Neural Networks*. Springer, 2000.
- [16] Sanchez E., Gaytan A., and Maarouf S. Decentralized neural identification and control for robotics manipulators. *International Symposium on Intelligent Control*, pages 1614–1619, 2006.
- [17] Sanchez E. N. and Alanis A. Y. *Redes neuronales: Conceptos fundamentales y aplicaciones a control automático*. Pearson Educación, S. A., Madrid, 2006.
- [18] Sarangapani J. *Neural Network Control of Nonlinear Discrete-Time Systems*. Taylor & francis, New York, 2006.
- [19] Sarangapani J., Lewis F. L., and Yesildirek A. *Neural Network Control of Robot Manipulators and Nonlinear Systems*. Taylor & francis, 1999.
- [20] Silician B., Bastin G., and Canudas C. *Theory of Robot Control*. Springer-Verlag, New York, 1996.
- [21] Su-Hau H. and Li-Chen F. Globally adaptive decentralizad control of robot manipulators. *SICE*, pages 402–407, 2002.
- [22] Symon K. R. *Mechanics*. Addison-Wesley, 3rd edition, 1971.
- [23] Utkin V. I. *Sliding Modes in Control and Optimization*. Springer-Verlag, New York, USA, 1992.
- [24] Utkin V. I. Sliding mode control design principle and applications to electrical drives. *IEEE Transactions on Industrial Electronics*, 40:23–36, 1993.
- [25] Welch G. and Bishop G. *An Introduction to the Kalman Filter*. ACM, USA, 2001.



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

Guadalajara, Jalisco, a 30 de Junio de 2008

Lic. Alfredo Alejandro Aguilar
Subdirector de posgrado
CINVESTAV del IPN
P r e s e n t e .

La presente es para informar a usted que el Comité Tutorial aprueba el contenido de la tesis de **MAESTRÍA** del (la) C.

Fernando ORNELAS TELLEZ

titulada

**Control Robusto Discreto de Robots Manipuladores con Redes
Neuronales**

Dr. Edgar Nelson Sánchez Camperos
Investigador CINVESTAV 3D
CINVESTAV Unidad Guadalajara

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

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

Dr. Antonio Ramírez Treviño
Investigador CINVESTAV 3A
CINVESTAV Unidad Guadalajara



CINVESTAV
BIBLIOTECA CENTRAL



SSIT000006889