

UNIVERSIDAD NACIONAL DE INGENIERIA

FACULTAD DE INGENIERIA ELECTRICA Y ELECTRONICA



“CONTROL DE SERVO SISTEMAS NO LINEALES USANDO REDES NEURONALES”

TESIS

PARA OPTAR EL TITULO PROFESIONAL DE :

INGENIERO ELECTRONICO

PRESENTADO POR :

LUIS ALEXANDER PONCE DIOSES

PROMOCION 1996-II

**LIMA - PERU
1998**

***Dedico este trabajo a mis padres
Justo y Consuelo los cuales con
su paciencia, amor y confianza
siempre están a mi lado.***

**CONTROL DE SERVO SISTEMAS NO LINEALES USANDO
REDES NEURONALES**

SUMARIO

En este trabajo de tesis se resuelve el problema de identificación y seguimiento de trayectoria de sistemas dinámicos no lineales empleando redes neuronales. En la identificación se utiliza un modelo dinámico para la representación de la planta no lineal. Las redes neuronales forman parte de este modelo como subsistemas. Para el control, se linealiza la planta no lineal con los subsistemas encontrados mediante una REALIMENTACION DE COMPENSACION de forma tal que este sistema linealizado pueda ser controlado mediante un controlador lineal PID. Se realiza la simulación y implementación del sistema de control e identificación de la planta no lineal con la finalidad de mostrar sus ventajas.

INDICE

PROLOGO	1
CAPITULO I	
PLANEAMIENTO DEL PROBLEMA	3
1.1 Enunciado del problema.....	3
1.2 Descripción de símbolos y variables.....	4
CAPITULO II	
CONCEPTOS FUNDAMENTALES	6
2.1 Neuronas biológicas y sus modelos artificiales	6
2.1.1 Fisiología de una neurona individual	7
2.1.2 La unión sináptica	9
2.1.3 Circuitos neuronales	10
2.2 Modelo neuronal	12
2.2.1 Tipos de funciones de activación	15
2.3 Arquitecturas de redes	15
2.3.1 Red de alimentación directa de una capa simple	15
2.3.2 Red de alimentación directa multicapa	16
2.3.3 Redes periódicas	17
2.3.4 Estructuras enrejadas	18
CAPITULO III	
ESTRUCTURA Y MODELO DEL SISTEMA	21
3.1 Estructura del sistema	21
3.1.1 Motor y driver (Actuador)	21
3.1.2 Codificador óptico y PLD (Sensores)	21
3.1.3 Tarjeta de interface	25
3.1.4 Software	26
3.2 Modelamiento	26
3.2.1 Modelamiento del Brazo Robótico de un Grado de Libertad (BR1L)	26
3.2.2 Modelo del driver y motor	29
CAPITULO IV	
ALGORITMO DE ENTRENAMIENTO Y SISTEMAS DE IDENTIFICACION USANDO REDES NEURONALES	30
4.1 Redes neuronales de alimentación directa multicapa	30

4.1.1	Notaciones	31
4.1.2	Derivación del algoritmo de retropropagación	32
4.1.3	No linealidad signoidal	38
4.1.4	Modelos de entrenamiento Pattern y Batch	39
4.1.5	Criterio de fin de entrenamiento	41
4.2	Resumen del algoritmo de retropropagación	42
4.3	Modelos de identificación	45
4.3.1	Modelo de identificación paralelo	45
4.3.2	Modelo serie-paralelo	46
CAPITULO V		
IDENTIFICACION Y CONTROL USANDO REDES NEURONALES		48
5.1	Identificación de los parámetros del motor utilizando redes neuronales...	48
5.1.1	Identificación de $f(n\omega_k)$	50
5.1.2	Identificación de $f(n\omega_k)$ y los parámetros lineales del motor sin carga ..	53
5.1.3	Identificación de los parámetros del motor con carga	58
5.2	Compensación del sistema no lineal	65
5.2.1	Compensación del motor sin carga	65
5.2.2	Compensación del motor con carga	65
5.3	Control no lineal del BR1L	67
5.3.1	Análisis del sistema linealizado	67
5.3.2	Diseño del servo controlador	69
5.4	Control PID del BR1L	74
5.5	Control del BR1L utilizando realimentación de estado	79
CAPITULO VI		
DESARROLLO DEL PROYECTO		83
6.1	Diseño e implementación del driver del motor DC	83
6.1.1	Circuito generador de la lógica de disparo	83
6.1.2	Circuito amplificador PWM	84
6.2	Calculo de los parámetros del sistema	85
6.2.1	Amplificador de potencia del PWM	85
6.2.2	Compensación de la fricción estática y de Coulomb	86
6.2.3	Identificación en el dominio de la frecuencia del sistema linealizado	88
6.3	Identificación neuronal	92
6.3.1	Adquisición de datos	92
6.3.2	Entrenamiento de la red neuronal	94
6.4	Software de control	96
CAPITULO VII		
RESULTADOS		100
7.1	Adquisición de datos	100
7.2	Identificación de las no linealidades del BR1L	100
7.2.1	Identificación de la fricción estática y de Coulomb	101

7.2.2	Identificación de la componente sinusoidal del peso de la varilla	103
7.3	Compensación del sistema no lineal	105
7.3.1	Compensación del motor sin carga	105
7.3.2	Compensación del motor con carga	105
7.4	Sistema de control implementado	107
7.4.1	Control PID del motor sin varilla	107
7.4.2	Control PID del sistema BR1L	107
	CONCLUSIONES Y FUTUROS TRABAJOS	109
	ANEXO A	
	CIRCUITOS DEL DRIVER DEL MOTOR	111
	BIBLIOGRAFIA	116

PROLOGO

Redes neuronales multicapa se han utilizado exitosamente en problemas de reconocimiento de patrones y en numerosas aplicaciones que son sugeridas en la literatura [2], [3]. Retropropagación estática ¹es uno de los métodos estándar usados en estos casos para el ajuste de pesos (parámetros) de la red neuronal.

Las propiedades de mapeo de una red neuronal juegan un rol importante en todos los estudios analíticos de tales redes. En [2] comentan que una red neuronal de dos capas y con un número arbitrario de nodos en las capas ocultas puede aproximar algunas funciones continuas $f \in C(\mathbb{R}^n, \mathbb{R}^m)$ sobre un subconjunto compacto de \mathbb{R}^n . Esto proporciona la motivación de asumir que las redes neuronales son adecuadas para tratar con una gran clase de problemas en la teoría de sistemas no lineales. Por tanto, dada la habilidad de una red neuronal para aproximar diferentes clases de funciones no lineales con suficiente precisión, las redes neuronales se convierten en primeras candidatas para su uso en modelos dinámicos para la representación de plantas no lineales. El hecho que los métodos de retropropagación estática y dinámica pueden ser usados para los ajustes de los parámetros internos de una red neuronal, lo hace atractivo en identificadores y controladores.

Al ser todos los sistemas físicos del mundo real no lineales, éstos pueden aproximarse a un modelo dinámico ideal, despreciando y aproximando las no linealidades dentro de un pequeño rango de operación, en el cual se diseña el controlador. Si el controlador diseñado se utiliza fuera de su rango de operación, el control de seguimiento de trayectorias, por ejemplo, sería pobre y en algunos casos inestable. Por tanto, el objetivo de esta Tesis es mostrar teóricamente y experimentalmente un tipo de control no lineal el cual hace uso de redes neuronales, las cuales han modelado las no linealidades más significativas del sistema real a controlar. Por tanto, el uso de estas en el control aumenta el rango de operación del sistema.

Se eligió al Brazo Robótico de Un Grado de Libertar por ser un sistema inherentemente no lineal. Por tanto, adecuado para mostrar el funcionamiento del control no lineal. El costo de implementar este sistema con el objeto de estudio es económico, sencillo y puede ser realizado en corto plazo.

¹Static Backpropagation.

El desarrollo de la presente trabajo ha sido posible gracias al apoyo técnico, teórico y crítico de los Drs. Luis G. Herrera B. y Arturo Rojas M. La financiación del trabajo fue posible gracias al programa de apoyo a proyectos de investigación realizada por el CONCYTEC. Igualmente se agradece al Laboratorio de Electrónica de la Facultad por el préstamo de equipos y dispositivos y al Laboratorio de Microelectrónica dirigida por el Dr. Paz Campaña el cual desarrollo un circuito en un PLD que mide la posición absoluta del BR1L. También quiero hacer mención a los Srs. William Astocondor y Carlos Kobashikawa quienes ayudaron también en la implementación del sistema.

CAPITULO I

PLANEAMIENTO DEL PROBLEMA

En el presente capítulo se enuncia el problema que se pretende resolver y se realiza una breve explicación de los pasos seguidos para la resolución del problema. Finalmente, se describen los símbolos y variables utilizados en la Tesis.

1.1 Enunciado del problema

El enunciado del problema es el siguiente:

Se desea resolver el problema de identificación y control no lineal de seguimiento de trayectorias utilizando redes neuronales de un motor DC controlado por voltaje de armadura el cual tiene acoplado a su eje una varilla a manera de un brazo robótico de un grado de libertad. Este sistema no lineal será llamado BR1L por facilidad. La trayectoria a seguir debe de ser continua, diferenciable y definida en todo el rango de la posición del BR1L, q , ($0 \leq q \leq 2\pi$). La entrada al sistema es el voltaje de armadura del motor DC y la salida es la posición del BR1L, debiéndose estimar otras variables del sistema si son requeridas.

La solución del problema propuesto se ha dividido en dos etapas. La primera etapa consiste en la identificación de los elementos no lineales del BR1L que son la fricción estática y de Coulomb y la carga no lineal de la varilla (componente sinusoidal), que es una función de la posición angular de la carga. Estas dos no linealidades son identificadas por dos redes neuronales multicapa de alimentación directa, entrenadas mediante el algoritmo de retropropagación (back-propagation) debido a la característica estática de las funciones a identificar. El objetivo de la identificación es linealizar el sistema BR1L utilizando una compensación por realimentación con la red neuronal. El sistema utilizado se puede considerar como un sistema de primer orden si se desprecia el efecto de la inductancia de la armadura. Esta consideración se utiliza con el fin de tener acceso a los elementos no lineales del sistema BR1L mediante una simple traslación de los mismos hacia la tensión de entrada del servomotor, pudiendo éstos eliminarse mediante una realimentación de compensación.

La segunda etapa consiste en la realización del control. Una vez linealizado el sistema, podemos diseñar cualquier tipo de control clásico; el controlador utilizado en esta tesis es un PID.

El sistema experimental BR1L esta constituido, además del servomotor, de un sensor óptico de posición, un decodificador de cuadratura diseñado en un PLD (Dispositivo Lógico Programable), que también integra contadores y la interfaz a la PC para medir la posición absoluta del BR1L. El sistema es controlado mediante un driver PWM (Pulse Width Modulation) de potencia con configuración tipo H.

Se utilizó una PC con un procesador Pentium de 100 Mhz y una tarjeta de adquisición de datos. El Software se desarrolló en lenguaje C y consiste de los siguientes módulos: (1) identificación y control utilizando redes neuronales; (2) algoritmo de entrenamiento de la red neuronal (retropropagación); (3) visualización en tiempo real de los parámetros identificados.

1.2 Descripción de símbolos y variables

Los parámetros del motor DC, driver y los símbolos mas importantes a la cual se hace referencia en capítulos posteriores de la tesis se describen a continuación:

Símbolo	Descripción
RNA	Redes Neuronales Artificiales
$\Pi_{i_1, i_2, \dots, i_{N+1}}^N$	Red Neuronal de N capas con i_1 entradas y i_{N+1} salidas
T	Torque del Motor
J_M	Momento de inercia del motor
J_L	Momento de inercia de la carga
b_M	Coefficiente de fricción viscosa del motor
b_L	Coefficiente de fricción viscosa de la carga
M	Masa de la esfera
m	Masa de la varilla
L_0	Longitud de la varilla
R_0	Radio de la esfera
E	constante de la fuerza contraelectromotriz del motor
$f^*(.)$	Modela el torque de fricción estática y de Coulomb
w_M	Velocidad angular del motor
w	Velocidad angular de la varilla
q_M	Posición angular del motor
q	Posición angular de la varilla
n	Factor de reducción de velocidad
V_a	Tensión de entrada en los bornes del motor
I	Corriente de armadura del motor
R	Resistencia de armadura
L	Inductancia de armadura

e_b	Fuerza contra-electromotriz
K_{act}	Ganancia del amplificador de potencia
$\xi(n)$	Suma instantánea del cuadrado del error en la iteración n
ξ_{av}	Error cuadrático promedio de $\xi(n)$ sobre el número total de muestras
w_{ij}	Peso sináptico entre la salida neurona i - entrada neurona j
ν_j	Nivel de actividad interna de la neurona j
$\varphi_j(\cdot)$	Función de activación asociada con la neurona j
η	Parámetro de la razón de aprendizaje
$\delta_j(n)$	Gradiente local
E_f	Error cuadrático por fase
ECM	Error cuadrático por muestra
ECFE	Error cuadrático por fase de entrenamiento
ECPE	Error cuadrático promedio por fase de entrenamiento
T_s	Periodo de muestreo

Tabla 1.1: Tabla de símbolos y variables

CAPITULO II

CONCEPTOS FUNDAMENTALES

Existen diferentes respuestas posibles a la pregunta de cómo definir una red neuronal. En un extremo, una respuesta podría ser que las redes neuronales son simplemente una clase de algoritmos matemáticos; desde que una red puede conceptuarse esencialmente como una notación gráfica para una gran clase de algoritmos. Tales algoritmos producen soluciones a un número de problemas específicos. En el otro extremo, una definición sería, que son redes sintéticas que simulan las redes neuronales biológicas encontradas en los organismos vivientes. En vista del conocimiento limitado de las redes neuronales biológicas en organismos vivientes; la respuesta más recomendable parece estar más cerca de la algorítmica.

En la búsqueda de mejores soluciones para la ingeniería y tareas computacionales, muchas rutas se han seguido. Hay una larga historia respecto al interés en las ciencias biológicas por parte de ingenieros, matemáticos y físicos empeñados por ganar nuevas ideas, inspiraciones y diseños. Las redes neuronales artificiales (RNA) han sido indudablemente una inspiración de la fisiología neuronal, pero la correspondencia entre la RNA y un sistema real neuronal son aún débiles. Existen vastas discrepancias en la arquitectura y funcionamiento de ambas. El conocimiento sobre las funciones reales del cerebro son limitadas. Ninguno de los modelos han tenido éxito en replicar el desempeño del cerebro humano. Por lo tanto el cerebro es y será solo una metáfora para una gran variedad de configuraciones de redes desarrolladas. A pesar de la pérdida de analogía entre las RNA y un natural sistema neuronal, se revisará brevemente el modelo neuronal biológico. En este capítulo se mostrará un modelo sintético neuronal y ejemplos de clases de redes neuronales, introduciéndose también las notaciones y símbolos usados.

2.1 Neuronas biológicas y sus modelos artificiales

El cerebro humano consiste de aproximadamente 10^{10} elementos computacionales llamadas neuronas. Ellas se comunican a través de una red de conexiones sinápticas teniendo una densidad de aproximadamente 10^4 sinapsis por neurona. Nuestra hipótesis con respecto al modelo del sistema nervioso natural, es que las neuronas se comunican con otras por medio de impulsos eléctricos. Las neuronas operan en un entorno químico que es aún más importante en términos del comportamiento real del cerebro. Nosotros

podemos así, considerar al cerebro como una densa red de conmutadores eléctricos condicionada en su mayor parte por los procesos bioquímicos. La vasta red neuronal tiene una estructura elaborada con interconexiones muy complejas. La entrada a la red es provista por receptores sensoriales y motoras. Los receptores reciben estímulos originados dentro del cuerpo, como también desde órganos sensores (vista, olfato, etc) cuando los estímulos se originan en el mundo externo. Los estímulos son en forma de impulsos eléctricos que transmiten la información a la red de neuronas. Como resultado del procesamiento de la información en el sistema nervioso central, los efectores son controlados originando una respuesta humana en la forma de acciones. Así tenemos un sistema de tres estados: (1)receptores, (2)redes neuronales y (3)efectores para el control del equilibrio del medio interno y externo del organismo.

2.1.1 Fisiología de una neurona individual

La figura 2.1 representa los componentes principales de una célula nerviosa típica perteneciente al sistema nervioso central. La membrana de la neurona separa el plasma intracelular del fluido extracelular que se encuentra fuera de la célula. La membrana es permeable para ciertas especies iónicas, y actúa del tal forma que se mantenga una diferencia de potencial y de concentración iónica entre el fluido intracelular y el fluido extracelular. Este efecto se consigue primordialmente mediante la acción de una bomba de sodio-potasio. También están presentes otras especies iónicas, como son los iones cloruro e iones orgánicos negativos.

Todas las especies iónicas se pueden difundir a través de la membrana, con la excepción de los iones orgánicos que son demasiado grandes. Dado que los iones orgánicos no pueden salir de la célula por difusión, su carga negativa neta dificulta la entrada en la célula de iones cloro por difusión. Por tanto, habrá una concentración más alta de iones cloro fuera de la célula. La bomba sodio-potasio determina una concentración más alta de potasio y de sodio dentro y fuera de la célula respectivamente.

La membrana celular es selectivamente más permeable para los iones de potasio que para los iones de sodio. El gradiente químico del potasio tiende a hacer que los iones de potasio salgan de la célula por difusión, pero la fuerte atracción de los iones orgánicos negativos tienden a mantener dentro el potasio. El resultado de estas fuerzas opuestas alcanzan un equilibrio en el cual hay más iones de sodio y cloro fuera de la célula, y más iones orgánicos y de potasio dentro de ella. Además, el equilibrio resultante produce una diferencia de potencial de unos 70 a 100 milivoltios (mV) a través de la membrana de la célula, siendo el más negativo el fluido intracelular. Este potencial que se denomina **potencial de reposo** de la célula, está representado esquemáticamente en la figura 2.2

La figura 2.3 ilustra la neurona con varias conexiones de entrada y los potenciales que presenta en distintas posiciones.

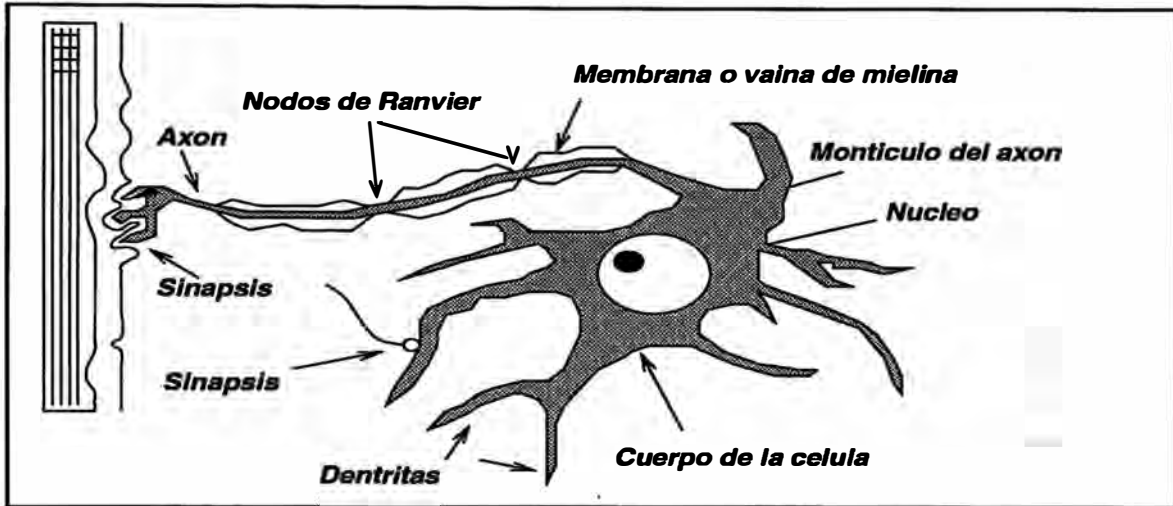


Figura 2.1: Estructura fundamental de una célula nerviosa típica mostrando las dendritas, el cuerpo de la célula y un único axón. El axón está rodeado por una membrana que se denomina vaina de mielina. Los nodos de Ranvier interrumpen periódicamente la banda de mielina a lo largo del axón. Las sinapsis conectan el axón de la neurona con las distintas partes de otras neuronas.

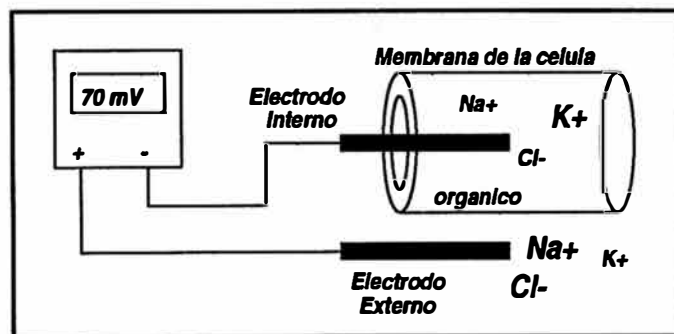


Figura 2.2: Esta figura ilustra el potencial de reposo que se establece en ambos lados de la membrana de una neurona. Los tamaños de las letras que representan las especies iónicas denotan aproximadamente la concentración relativa de cada especie en las regiones internas y externas de las células.

Las entradas excitatorias que llegan a la célula reducen o aumentan la diferencia de potencial que existe entre los dos lados de la membrana celular. La despolarización resultante en el **montículo del axón** altera la permeabilidad de la membrana celular a efectos de los iones de sodio. Como resultado hay un fuerte flujo entrante de iones sodio positivos, que penetran en la célula, contribuyendo aún más a la despolarización. Este efecto autogenerado da lugar al **potencial de acción**.

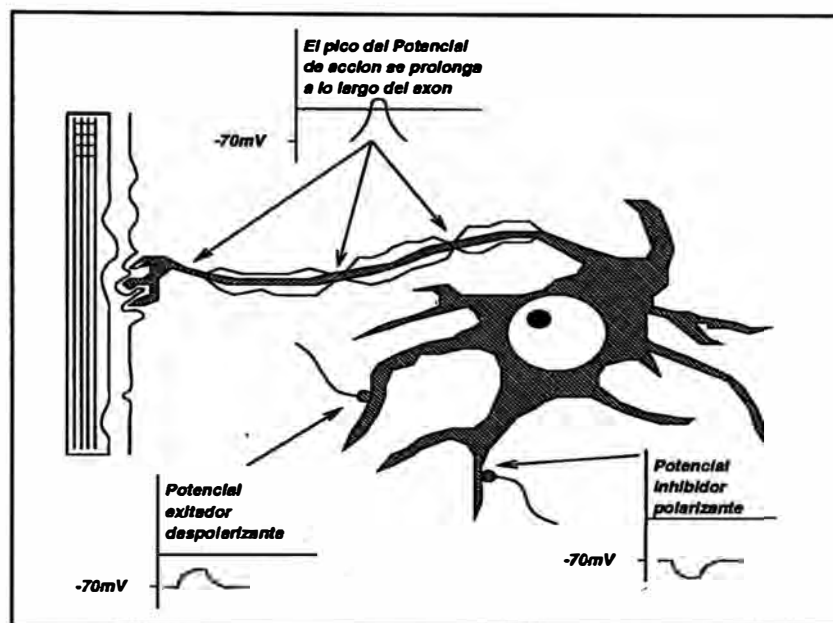


Figura 2.3: Neurona con varias conexiones de entrada y los potenciales que presenta.

Las fibras nerviosas en sí son malos conductores. La transmisión del potencial de acción a lo largo del axón es el resultado de una serie de despolarizaciones que tienen lugar en los nodos de Ranvier. Cuando uno de los nodos se despolariza, se desencadena la despolarización del siguiente nodo. El potencial de acción viaja a lo largo de la fibra en forma discontinua, de un nodo a otro. Una vez que un potencial de acción ha pasado por un cierto punto, ese punto no puede volver a ser excitado durante el tiempo de 1 milisegundo, que es el tiempo que tarda en volver a su potencial de reposo. Este **período refractario** limita la frecuencia de transmisión de los impulsos nerviosos a unos 1000 por segundo.

2.1.2 La unión sináptica

Examinemos brevemente la actividad que se desarrolla en la unión existente entre dos neuronas, la que se denomina unión sináptica o **sinapsis**. La comunicación de las neuronas tiene lugar como resultado de la liberación de sustancias llamadas **neurotransmisores** por parte de la célula presináptica, al ser absorbidas estas sustancias por la célula postsináptica, La figura 2.4 muestra dicha actividad. Cuando el potencial de

acción llega a la membrana presináptica, los cambios de permeabilidad de la membrana dan lugar a un flujo entrante de iones de calcio. Estos iones dan lugar a que las vesículas que contienen los neurotransmisores se fundan con la membrana sináptica, liberando así sus neurotransmisores en la separación sináptica.

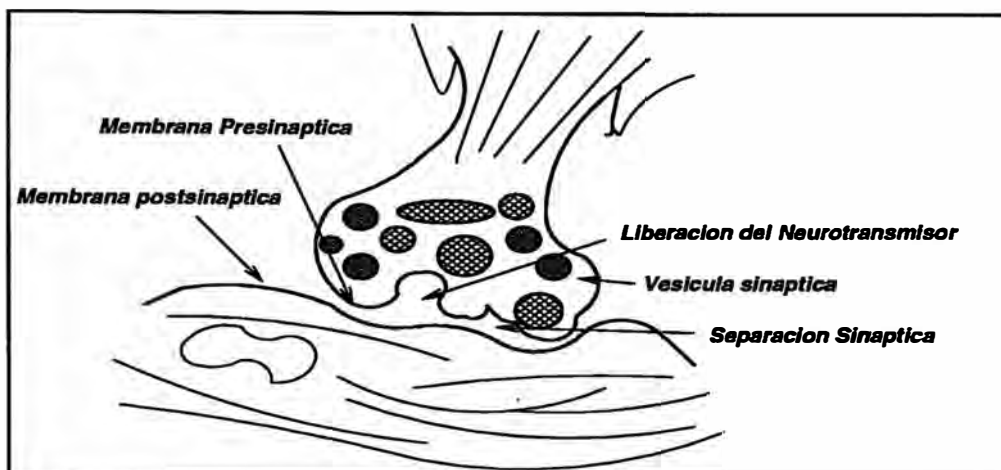


Figura 2.4: Los neurotransmisores están almacenados en la vesícula, próximas a la membrana presináptica. Estas sustancias se liberan en la separación sináptica y se difunden hacia la membrana postsináptica, en la cual son absorbidas posteriormente.

Los neurotransmisores se difunden a través de la unión y se unen a la membrana postsináptica en ciertos lugares llamados receptores, La acción química que se produce en los receptores dan a lugar a cambios de permeabilidad de la membrana postsináptica para ciertas especies iónicas. Un flujo entrante de especies positivas hacia la célula tenderá a despolarizar el potencial de reposo; este efecto es excitatorio. Si entran iones negativos, se producirá un efecto hiperpolarizante; este efecto es inhibitorio. Estos dos efectos son locales, y actúan tan sólo a lo largo de una pequeña distancia hacia el interior de la célula; se suman en el montículo del axón . Si la suma es mayor que un cierto valor umbral se genera un potencial de acción.

2.1.3 Circuitos neuronales

La figura 2.5 muestra varios circuitos neuronales básicos que se encuentran en el sistema nervioso central. Las figuras 2.5 (a) y (b) ilustra los principios de convergencia y divergencia respectivamente en la circuitería neuronal. Cada neurona envía impulsos a muchas otras neuronas (divergencia) y recibe impulsos procedentes de muchas neuronas (convergencia). Esta sencilla idea parece ser el fundamento de toda la actividad del sistema nervioso central, y forma la base de la mayoría de los modelos de redes neuronales.

Obsérvese las vías de realimentación que aparecen en los circuitos de las figuras 2.5

(b), (c) y (d). Dado que las conexiones sinápticas pueden ser tanto excitatorias como inhibitorias, estos circuitos hacen posible que los sistemas de control pueden tener tanto realimentación positiva como de realimentación negativa. Por supuesto, estos circuitos tan sencillos no describen adecuadamente la gran complejidad de la neuroanatomía.

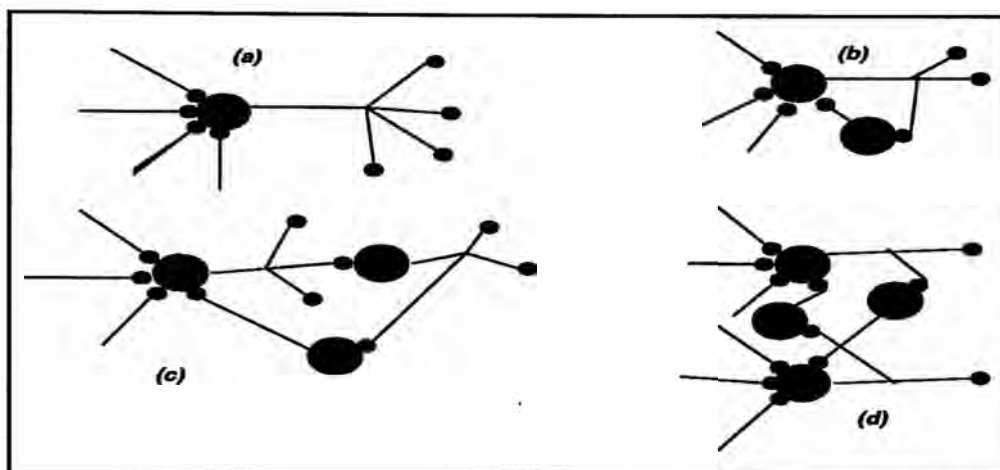


Figura 2.5: Los esquemas muestran ejemplos de circuitos neuronales del sistema nervioso central. Los cuerpos de las células (incluyendo dendritas) se han representado mediante círculos grandes. Los círculos pequeños se encuentran al final de los axones. En (a), (b) se ilustran los conceptos de divergencia y de convergencia. En (b), (c), (d) se pueden ver ejemplos de circuitos con vías de realimentación.

Ahora que ya tenemos una idea de la forma en que operan las neuronas individuales, y de la forma en que están conectadas entre sí, podemos formularnos una pregunta fundamental: ¿Cómo se combinan estos conceptos relativamente sencillos para dar al cerebro sus enormes capacidades?. El primer intento significativo para resolver esta pregunta se hizo en 1943, a través del trabajo seminal de McCulloch-Pitts. Este trabajo fue el primero en tratar el cerebro como un organismo computacional. La teoría de McCulloch-Pitts se basa en cinco suposiciones:

1. La actividad de la neurona es un proceso todo-nada (esta suposición indica que la neurona es binaria).
2. Es preciso que un número fijo de sinapsis (> 1) sean excitadas dentro de un período de adición latente, para que se excite una neurona.
3. El único retraso significativo dentro del sistema nervioso es el retardo sináptico.
4. La actividad de cualquier sinapsis inhibitoria impide por completo la excitación de la neurona en ese momento.
5. La estructura de la red de interconexiones no cambia con el transcurso del tiempo.

2.2 Modelo neuronal

Una neurona es una unidad de procesamiento de información, fundamental en la operación de la red neuronal. En la figura 2.6 se presenta un modelo neuronal. Nosotros podemos identificar tres elementos básicos de una neurona tal como se describe a continuación:

1. Un *conjunto de sinapsis o enlaces de conexiones*, donde cada una es caracterizada por su propio peso. Específicamente, una señal x_j a la entrada de la sinapsis j conectada a la neurona k es multiplicada por el peso sináptico w_{kj} . Es importante hacer una nota de la manera en que los subíndices de los pesos sinápticos w_{kj} son escritos. El primer subíndice se refiere a la neurona en cuestión y el segundo a la entrada y fin de la sinapsis a la cual el peso se refiere. El peso w_{kj} es positivo si la sinapsis asociada es excitatoria y es negativa si la sinapsis es inhibitoria.
2. Un *sumador*, para sumar las señales de entrada, ponderadas por la respectiva sinapsis de la neurona. La operación descrita aquí constituye un combinador lineal.
3. Una *función de activación*, para limitar la amplitud de la salida de la neurona.

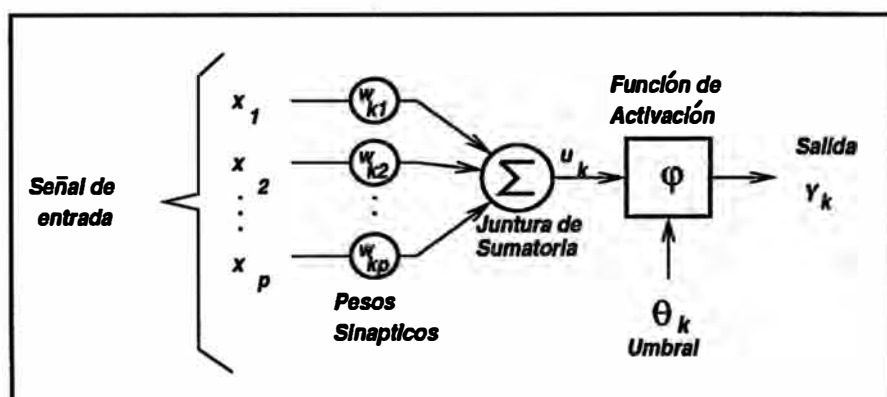


Figura 2.6: Modelo no lineal de una neurona.

El modelo de una neurona se muestra en la figura 2.6. También se incluye un umbral externo θ_k que tiene el efecto de rebajar el aporte neto de la función de activación. Por otra parte, el aporte neto de la función de activación puede ser aumentada por emplear un término *bias* que al contrario del umbral es el negativo de este. En terminos matemáticos, nosotros podemos describir la neurona k por las siguientes ecuaciones:

$$u_k = \sum_{j=1}^p w_{kj} x_j \quad (2.1)$$

$$y_k = \varphi(u_k - \theta_k) \quad (2.2)$$

donde x_1, x_2, \dots, x_p es la señal de entrada; $w_{k1}, w_{k2}, \dots, w_{kp}$ son los pesos sinápticos de la neurona k , u_k es la salida del combinador lineal, θ_k es el umbral; $\phi(\cdot)$ es la función de activación, e y_k es la señal de salida de la neurona. El uso del umbral θ_k tiene el efecto de aplicar una transformación afin a la salida u_k del combinador lineal en el modelo de la figura 2.6, visto por:

$$v_k = u_k - \theta_k \quad (2.3)$$

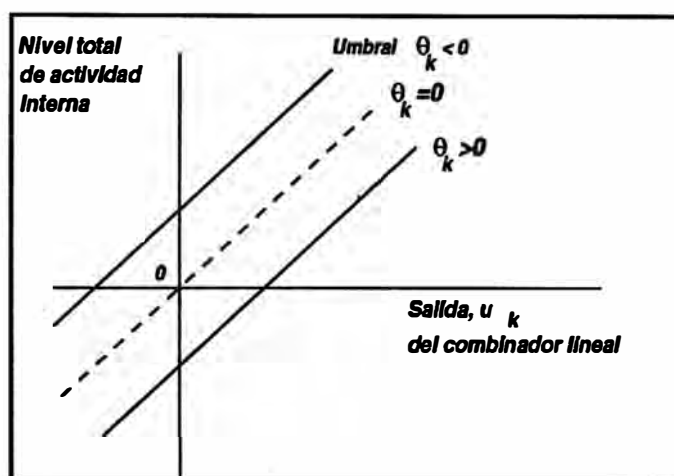


Figura 2.7: Transformación afin producida por la presencia del umbral

En particular, dependiendo si el umbral θ_k es positivo o negativo, la relación entre los efectivos niveles de actividad interna o potencial de activación v_k de la neurona k y la salida del combinador lineal u_k es modificado como se ilustra en la figura 2.7. Notar que como resultado de esta transformación afin, la gráfica de v_k versus u_k , no pasa a través del origen. El umbral θ_k es un parámetro externo de la neurona artificial k . Este umbral se muestra en la ecuación 2.2. Equivalentemente, reformulando las ecuaciones 2.1 y 2.2 como:

$$u_k = \sum_{j=0}^p w_{kj} x_j \quad (2.4)$$

$$y_k = \varphi(v_k) \quad (2.5)$$

Nosotros podemos ahora presentar el modelo de la neurona k como en la figura 2.8 (a). En esta figura el efecto del umbral está representado como sigue: (1) añadiendo una nueva señal de entrada fija en $x_0 = -1$, y (2) adicionado un nuevo peso sináptico w_{k0} igual al umbral θ_k . Un modelo alternativo se muestra en la figura 2.8 (b) donde la entrada del combinador lineal x_0 es fijada en $+1$ y el peso w_{k0} en el valor referencial b_k .

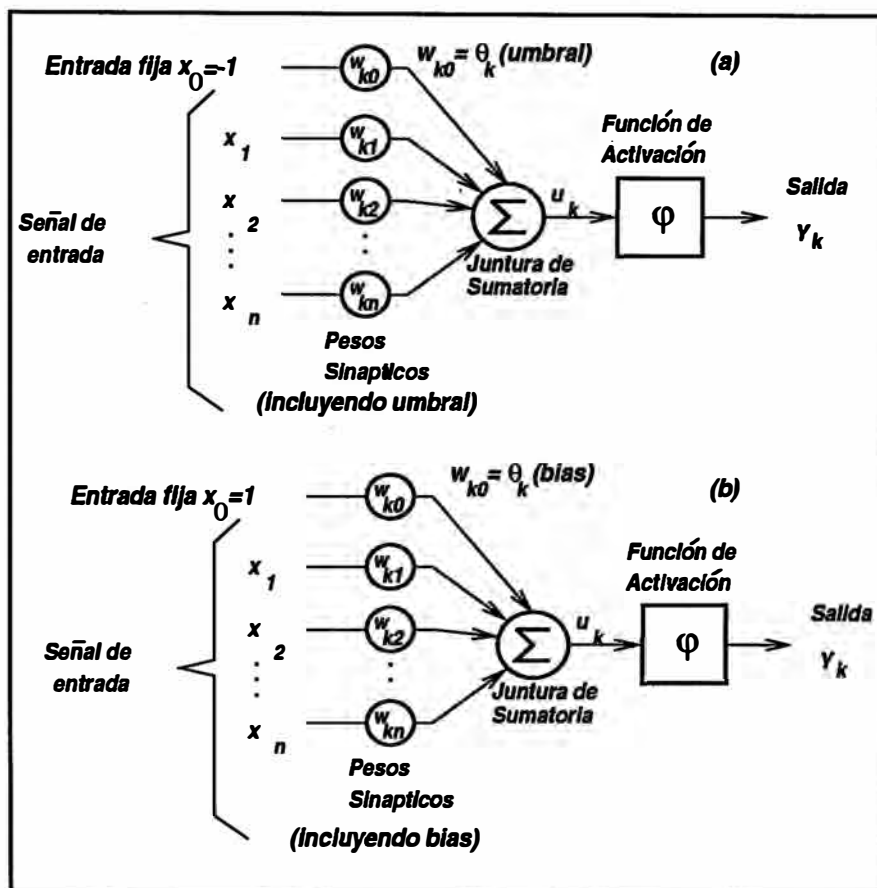


Figura 2.8: Dos modelos no lineales de una neurona

2.2.1 Tipos de funciones de activación

La función de activación denotada por $\varphi(\cdot)$. Define la salida de una neurona en términos del nivel de actividad en su entrada. Nosotros podemos identificar tres tipos básicos de funciones de activación:

Función Umbral. Para este tipo de función de activación, tenemos:

$$\varphi(v_k) = \begin{cases} 1 & \text{si } v_k \geq 0. \\ 0 & \text{si } v_k < 0. \end{cases} \quad (2.6)$$

Función Continua por partes. Para este tipo de función de activación, tenemos:

$$\varphi(v_k) = \begin{cases} 1 & \text{si } v_k > \frac{1}{2}. \\ v_k & \text{si } -\frac{1}{2} > v_k > \frac{1}{2}. \\ 0 & \text{si } v_k \leq -\frac{1}{2} \end{cases} \quad (2.7)$$

donde el factor de amplificación dentro de la región de operación lineal es la unidad. Esta forma de función de activación puede ser vista como una aproximación de un amplificador no lineal.

Función Sigmoidal. La función sigmoidal es la función de activación más común usada en la construcción de redes neuronales. Se define como una función suave y de propiedades asintóticas. Un ejemplo de sigmoide es la función *logistic* definida por:

$$\varphi(v) = \frac{1}{1 + e^{-av}} \quad (2.8)$$

donde a es un parámetro que varía la pendiente en la función sigmoide. La función tangente hiperbólica se usa a menudo también como función de activación sigmoide, la cual está definida por:

$$\varphi(v) = \tanh\left(\frac{v}{2}\right) = \frac{1 - e^{-v}}{1 + e^{-v}} \quad (2.9)$$

Esta función permite salidas positivas como negativas variando entre ± 1 .

2.3 Arquitecturas de redes

La manera en la cual las neuronas de una red neuronal se encuentran estructuradas, está íntimamente vinculada con el algoritmo de aprendizaje para entrenar la red. Nosotros podemos hablar por lo tanto de algoritmos de aprendizaje usados en el diseño de redes neuronales, como si fueran estructuras. En general, se puede identificar 4 diferentes clases de arquitecturas de redes descritas a continuación.

2.3.1 Red de alimentación directa de una capa simple

Se denomina capa a la agrupación de un conjunto de neuronas. Entonces una red neuronal con capas, es una red de neuronas organizadas en forma de capas. El tipo de

red más simple, es la que se muestra en la figura 2.9, la cual está formada por una capa de entrada de nodos fuente que se proyectan a una capa de salida de neuronas (nodos computacionales). Esta red es de tipo *alimentación directa* es llamada *red neuronal de una capa simple*, porque solo contiene una capa computacional y no se toma en cuenta en el conteo, la capa de nodos fuente.

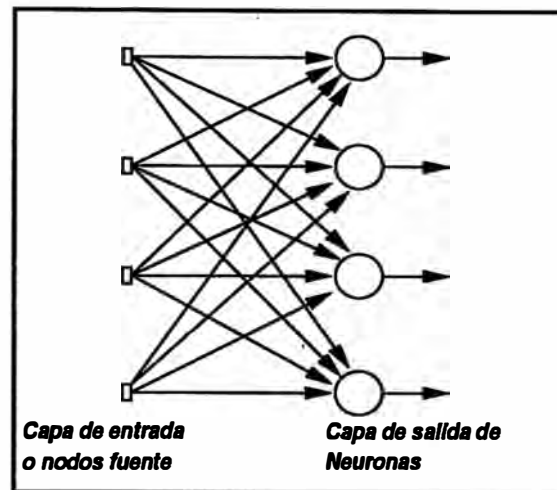


Figura 2.9: Red de alimentación directa con una capa simple de neuronas

Una memoria asociativa lineal es un ejemplo de una red neuronal de una capa simple, la cual asocia un patrón de salida con uno de entrada. La información es almacenada en virtud de las modificaciones hechas a los pesos sinápticos de la red.

2.3.2 Red de alimentación directa multicapa

Una segunda clase de redes neuronales se distingue por la presencia de una o más capas ocultas, cuyos nodos computacionales son correspondientemente llamados neuronas ocultas o capas ocultas. La función de las neuronas ocultas es intervenir entre una entrada externa y salida de la red. Por poseer una o mas capas ocultas, la red es capaz de extraer estadísticas de alto orden. Es decir, la red adquiere una perspectiva global, a pesar, de su conectividad local, en virtud del conjunto extra de conexiones sinápticas y la adicional dimensión de las interacciones neuronales.

Los nodos fuente en la capa de entrada de la red suministran los respectivos elementos del patrón de activación (el vector de entrada), el cual constituye la señal de entrada aplicada a las neuronas (nodos computacionales) de la segunda capa (por ejemplo primera capa oculta). La señal de salida de la segunda capa son utilizadas como entrada a la tercera capa, y así para el resto de la red. Típicamente, las neuronas en cada capa de la red tienen como entradas sólo la señal de salida de la capa precedente. El conjunto de señales de salida de la capa final de la red constituye la respuesta de la red al patrón de entrada suministrado por los nodos fuente de la capa de entrada. La

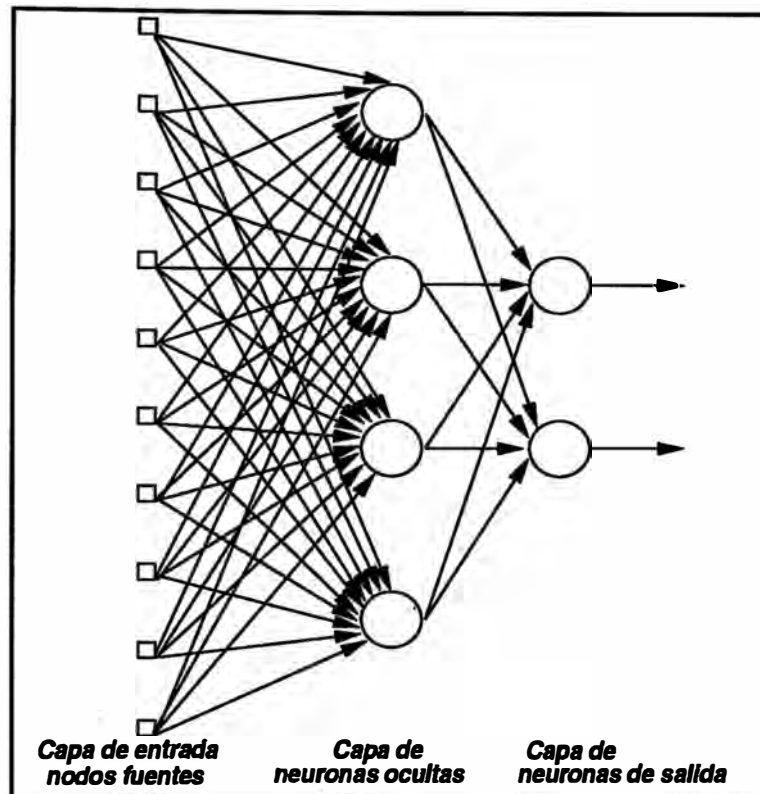


Figura 2.10: Red de alimentación directa de una capa oculta y una de salida

arquitectura gráfica de una red neuronal de alimentación directa multicapa para el caso de una capa oculta simple se ilustra en la figura 2.10. Por brevedad, esta red contiene 10 nodos fuente, 4 neuronas ocultas y 2 neuronas de salida. En posteriores referencias a una red neuronal de N capas se empleará el símbolo $\Pi_{i_1, i_2, \dots, i_{N+1}}^N$ donde la red tiene i_1 entradas, i_{N+1} salidas, $(N-1)$ conjuntos de neuronas en las capas ocultas, cada una conteniendo $i_2, i_3, i_4, \dots, i_N$ nodos respectivamente.

2.3.3 Redes periódicas

Una red neuronal periódica, se distingue de una red neuronal de alimentación directa, en que la primera incluye un lazo de realimentación. Por ejemplo, una red periódica puede consistir de una simple capa de neuronas, donde la salida de una neurona alimenta a las entradas de las otras neuronas, tal como se ilustra en la figura 2.11. En la estructura descrita en esta figura no hay un lazo de auto realimentación en la red. La red periódica mostrada no tiene neuronas ocultas. En la figura 2.12 se ilustra otra clase de red periódica con neuronas ocultas. Las conexiones de realimentación vistas en la figura 2.12 se originan en las capas ocultas, así como también de las neuronas de salida. Los lazos de realimentación presentados en las figuras 2.11 y 2.12 tienen un profundo impacto en la capacidad de aprendizaje de la red y su funcionamiento. Además los lazos

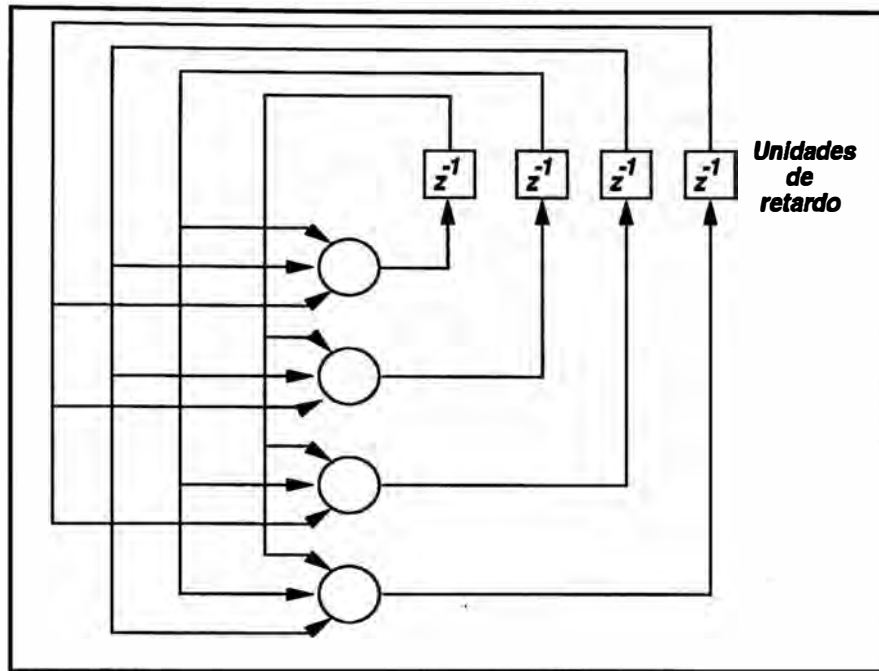


Figura 2.11: Red periódica sin lazos de realimentación ni neuronas ocultas

de realimentación involucran el uso de particulares ramas compuestas con unidades de retardo (denotadas por z^{-1}), las cuales dan un comportamiento dinámico no lineal a la red en virtud de la naturaleza no lineal de las neuronas.

2.3.4 Estructuras enrejadas

Una estructura enrejada consiste de un arreglo de una o más dimensiones de neuronas con un correspondiente conjunto de nodos fuente que suministran la señal de entrada al arreglo. La dimensión del enrejado se refiere al número de dimensiones del espacio, en cual el gráfico se extiende. La arquitectura gráfica de la figura 2.13a describe un enrejado dimensional de tres neuronas alimentadas desde una capa de tres neuronas fuentes, mientras que la arquitectura gráfica de la figura 2.13b describe un enrejado de dos dimensiones de 3 por 3 neuronas alimentadas desde una capa de 3 nodos fuente. Notar que en ambos casos cada nodo fuente es conectado a cada neurona en el enrejado. Una red de enrejado es realmente una red de realimentación directa con las neuronas de salida arregladas en filas y columnas.

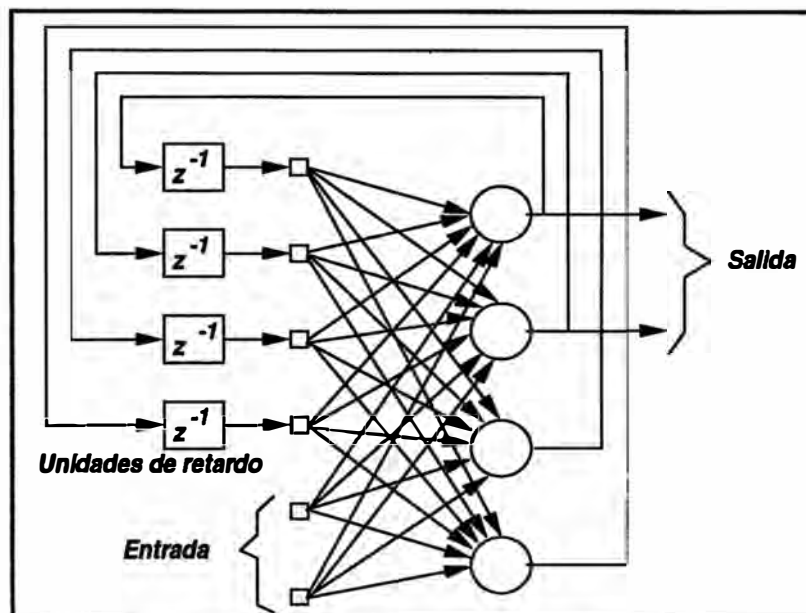


Figura 2.12: Red periódica con neuronas ocultas

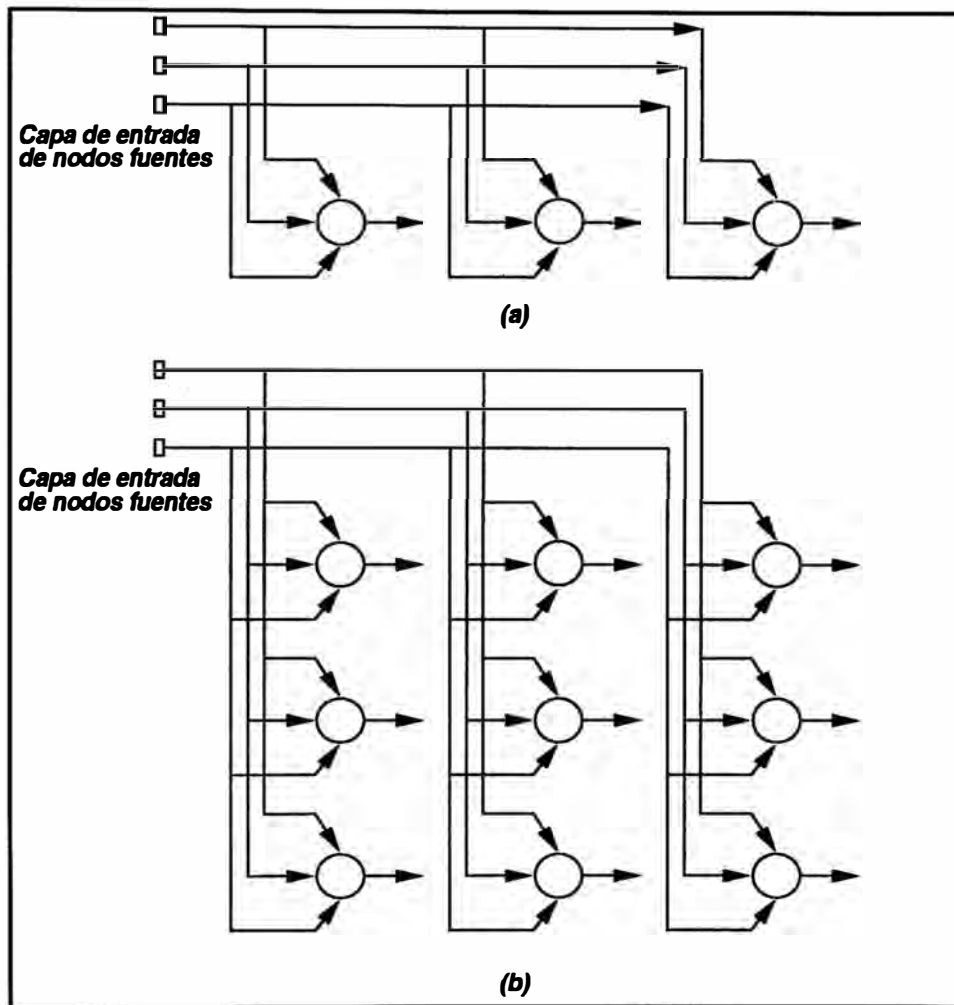


Figura 2.13: (a)Estructura de una dimensión con tres neuronas. (b)Estructura de 2 dimensiones de 3 por 3 neuronas.

CAPITULO III

ESTRUCTURA Y MODELO DEL SISTEMA

En este capítulo se presenta la descripción de la estructura del sistema y su modelamiento matemático que será utilizado en las simulaciones realizadas.

3.1 Estructura del sistema

En la figura 3.1 se muestra un esquema general del hardware implementado. A continuación se describe al actuador, el sensor, la interconexión entre los distintos dispositivos electrónicos y el software empleado.

3.1.1 Motor y driver (Actuador)

El actuador empleado es un servo motor DC de campo magnético permanente y escobillas conmutadas para el control de un brazo robótico de un grado de libertad.

Como el motor no responde inmediatamente a una tensión de entrada, le toma entonces un pequeño tiempo para llegar a la máxima velocidad. Si conmutamos a "off" la fuente, un tiempo antes que el motor llegue a la máxima velocidad, éste comienza a moverse a una velocidad menor. Un generador de PWM + Driver realizan exactamente lo descrito anteriormente; conmuta al motor con una serie de pulsos, logrando así, variar su velocidad de acuerdo al ancho del pulso. El LM3524 genera una modulación por ancho de pulsos de 15.4Khz (la cual se puede variar), esta señal es amplificada por un driver, que alimenta al motor.

En la figura 3.2 se muestra el esquema del funcionamiento del driver. Un sistema de disparo y dos pares de conmutadores A y B conforman el driver. El objetivo del sistema de disparo es realizar la conmutación de cada par A y B, de tal forma que se eviten cortocircuitos durante el tiempo de conmutación de A a B. Esto se ilustra en la figura 3.3.

3.1.2 Codificador óptico y PLD (Sensores)

El sensor de posición y sentido de giro del motor cuenta con dos entradas de alimentación y dos salidas seriales de tren de pulsos, desfasados 90 grados. Básicamente el codificador óptico rotatorio (sensor) consiste de 4 componentes: (1) una fuente de luz, el cual puede ser un diodo emisor de luz (LED) o una lámpara incandescente. Esto depende las consideraciones de diseño; (2) un disco opaco con segmentos transparentes,

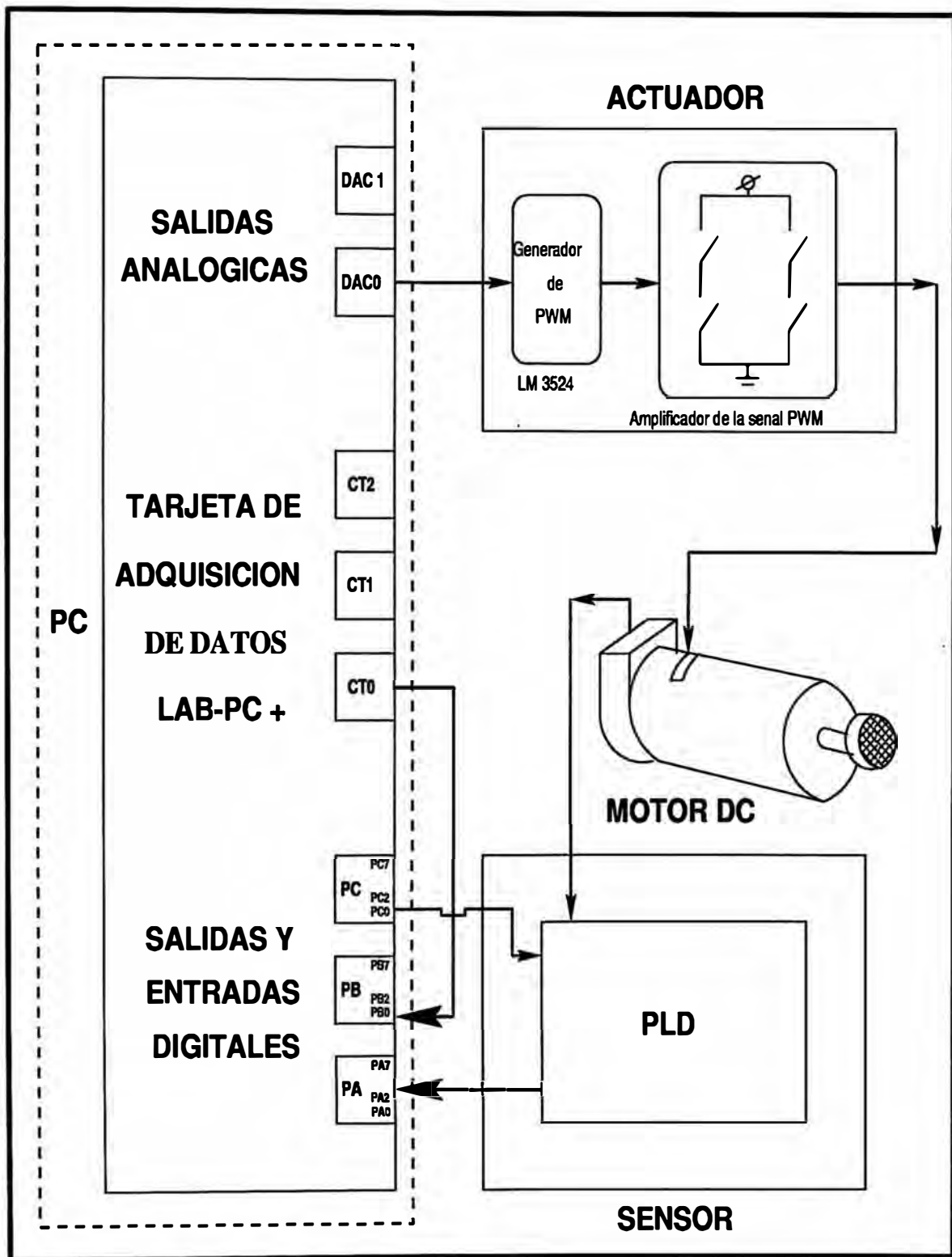


Figura 3.1: Esquema general del Sistema de Control

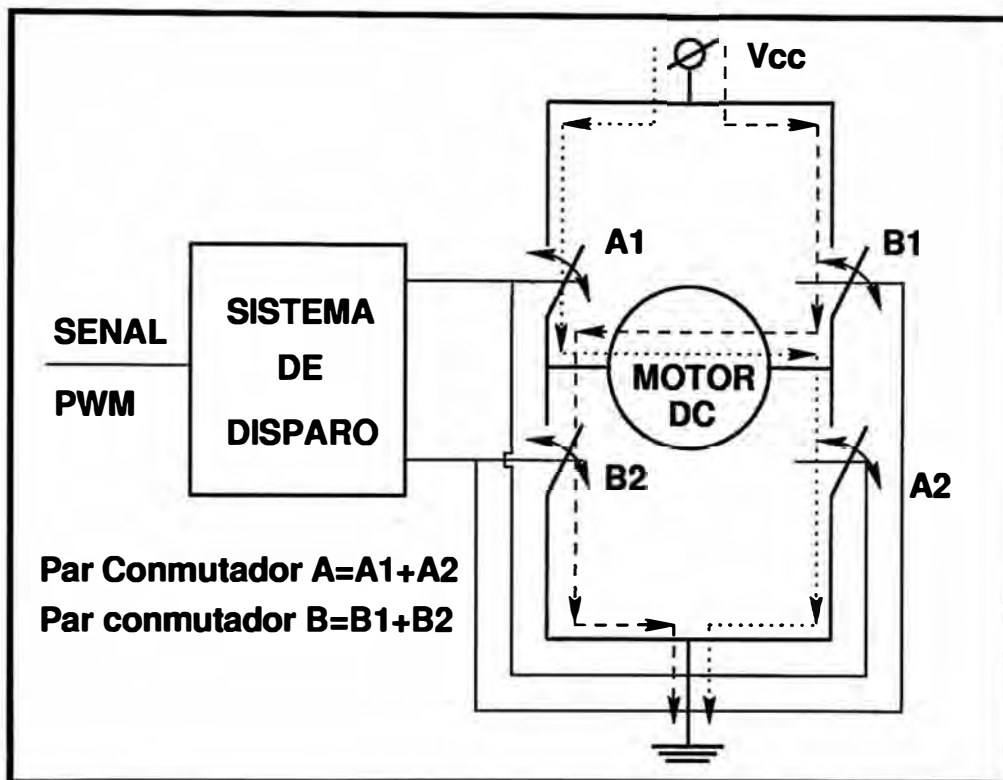


Figura 3.2: Esquema general del sistema de disparo y los conmutadores. Cuando el sistema de disparo cierra el conmutador A y abre B, el sentido de la corriente es la línea con puntos, induciendo de esta forma una tensión $+V_{CC}$ en el motor. Luego, si el sistema de disparo abre el conmutador A y cierra el B, el sentido de la corriente es la línea segmentada, induciendo así una tensión $-V_{CC}$ en el motor. Por consiguiente, el motor ve en sus bornes una onda de voltaje cuadrada, variando entre $\pm V_{CC}$ y la corriente que puede absorber dependerá de los conmutadores.

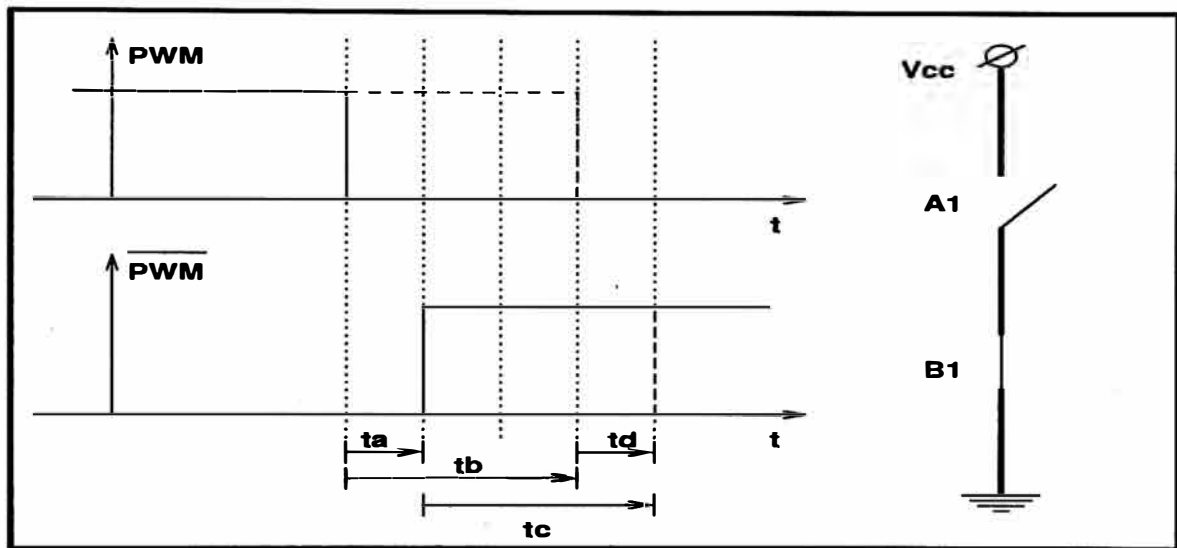


Figura 3.3: Cortocircuitos producidos con un sistema de disparo inadecuado. La figura muestra una señal PWM y otra \overline{PWM} que se a derivado por ejemplo, de una simple negación de la señal PWM, las cuales controlan los conmutadores A1 y B1 respectivamente. t_a es el retardo de la compuerta y la circuiteria involucrada hasta los conmutadores, t_b es el tiempo que demora la compuerta A1 en cerrarse y t_c es el tiempo que demora la compuerta B1 en abrirse; como puede observarse $t_d = t_a + t_c - t_b$ es el tiempo en que las compuertas A1 y B1 están cerradas, cortocircuitando la fuente. El efecto explicado se observa en el osciloscopio como respuestas transitoria subamortiguadas durante el tiempo de conmutación, esta se pueden atenuar colocando un condensador adecuado. El efecto dañino ocurre cuando por el conmutador circula una corriente grande por un tiempo t_d que mayormente es de unos cuantos nanosegundos; esta corriente va deteriorando poco a poco el dispositivo conmutador.

localizado entre la fuente de luz y su asociado sensor de luz, este disco se coloca de tal manera que siga el movimiento del dispositivo a ser sensado; la resolución del disco utilizado es de 512 pulsos por revolución; (3) un sensor de luz, frecuentemente se utiliza un fototransistor; (4) un circuito que convierta la salida del sensor a un conveniente formato de información necesaria para la interface (PLD). En la figura 3.4 se muestra un esquema del funcionamiento de un codificador óptico rotatorio.

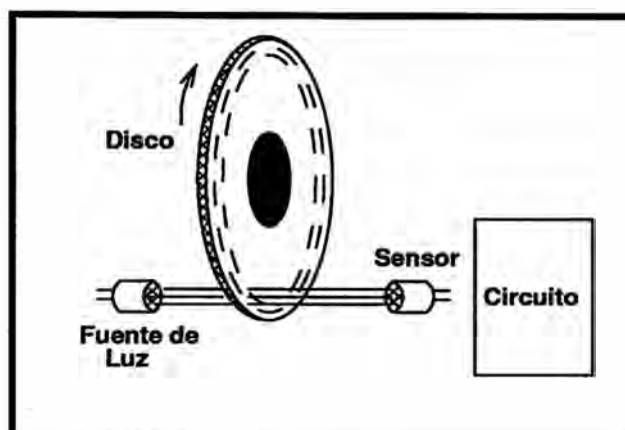


Figura 3.4: Esquema general del Codificador Óptico Rotatorio

Un PLD (Dispositivo lógico programable) es un circuito integrado digital, configurable por el usuario; es utilizado para implementar expresiones Booleanas o funciones que se han construido con estructuras lógicas. Esta es la principal diferencia con respecto a los dispositivos lógicos tales como los TTL's, que proveen una específica función lógica y no pueden ser modificados. En un EPLD (PLD que utiliza un EEPROM como elemento reconfigurable) EPM7128E de la familia MAX 7000 se ha integrado un circuito digital sensor de posición que consiste de un LS7083 y 4 contadores (74193), esto ocupa un 20% de la capacidad del EPLD.

3.1.3 Tarjeta de interface

Una tarjeta LAB-PC+ es utilizada para la adquisición de datos y envío de la señal de control. Para el control se cuentan con tres elementos: (1) dos salidas analógicas (DAC0, DAC1); la señal de control se envía por el DAC0 al LM3524 (Generador de PWM). Esta tiene una resolución de 12 bits para generar la señal analógica; (2) tres puertos de entrada-salida de 8 bits cada uno; los cuales básicamente consisten de un 8255 (PPI) que nos permite programar los puertos. El puerto A trabaja como entrada recibiendo los datos de salida de los contadores del PLD en bloques de 8 bits; el puerto B es configurado también como entrada. Solo se utiliza el bit B0 para la recepción de una señal de reloj (OUTB0) que sirve de base de tiempos en los programas; el puerto C se ha configurado como salida, el bit C0 envía una señal de control al PLD para

que envíe el bloque de 8 bits mas o menos significativos de los contadores del PLD; (3) 3 contadores-temporizadores integrados en el 8253 (Timer), donde un contador es utilizado para generar la señal de reloj a partir de un clock interno de 2MHz.

3.1.4 Software

Se utilizó una PC con un procesador Pentium de 100Mhz. El Software se desarrollo en C, el cual consiste en programas de identificación y control utilizando redes neuronales. Para el aprendizaje de la red se programaron módulos de aprendizaje con el algoritmo de retropropagación; los parámetros identificados se presentan en tiempo real y en modo gráfico observándose la evolución de los parámetros de la red. Para las operaciones con matrices se utilizan rutinas optimizadas en Fortran (Blas) las cuales se tradujeron a C. Las operaciones se realizan utilizando el tipo de dato "Double" para obtener mejor precisión. Las simulaciones se realizaron en "Matlab" y luego se tradujeron a C para aumentar la velocidad de simulación en el entrenamiento de la red.

3.2 Modelamiento

En esta sección se realiza el modelamiento matemático del brazo robótico de un grado de libertad y del driver + motor DC.

3.2.1 Modelo del Brazo Robótico de un Grado de Libertad (BR1L)

En la figura 3.5 se muestra un esquema general del BR1L. En este sistema se observan 2 partes importantes: (1) Un sistema mecánico y (2) Un sistema eléctrico.

Sistema Mecánico. Las ecuaciones del sistema mecánico se deducen usando la segunda ley de Newton aplicada al movimiento rotacional de K partículas de momento de inercia $J_j, j = 1, \dots, K$; sometidas a N torques externos $\tau_i, i = 1, \dots, N$; dada por:

$$\sum_{j=1}^K J_j \ddot{\theta} = \sum_{i=1}^N \tau_i$$

Para esta ecuación, tomando como referencia el eje del motor como se muestra en la figura 3.6 y considerando como positivo la dirección de las flechas, tenemos:

$$\begin{aligned} nT = & n(J_M \dot{\omega}_M + b_M \omega_M) + (ML_0^2 + \frac{2}{5}MR_0^2 + \frac{1}{3}mL_0^2 + J_L)\dot{\omega} + \\ & + gL_0(M + \frac{m}{2}) \sin q + b_L \omega + n f^*(\omega_M) \end{aligned} \quad (3.1)$$

Como $\omega_M = n\omega$ es la relación entre la velocidad motor-carga y de 3.1 se obtiene:

$$\begin{aligned} T = & (J_{eff} + \frac{1}{n}(ML_0^2 + \frac{2}{5}MR_0^2 + \frac{1}{3}mL_0^2))\dot{\omega} + b_{eff}\omega + \\ & + gL_0(M + \frac{m}{2}) \sin q + f^*(n\omega) \end{aligned} \quad (3.2)$$

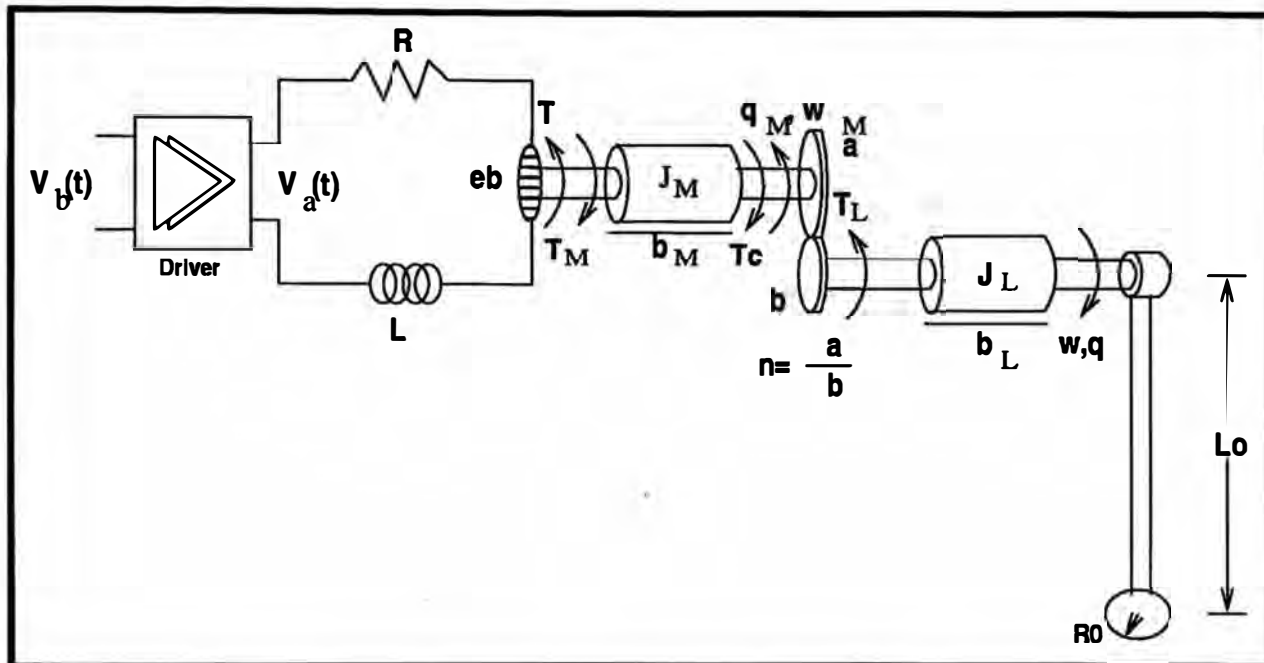


Figura 3.5: Esquema de la planta a controlar

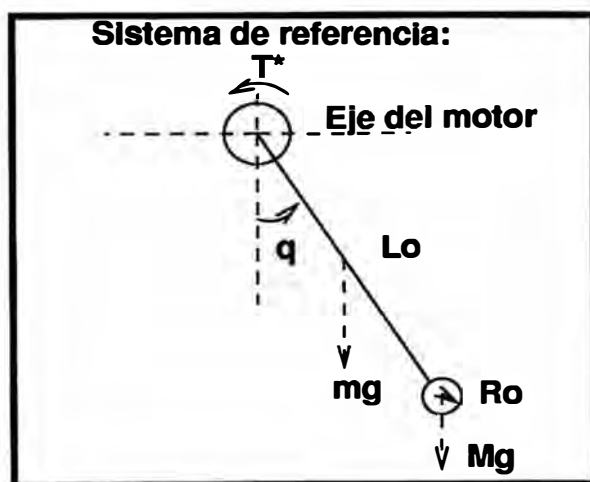


Figura 3.6: Sistema de referencia

donde:

$$J_{eff} = nJ_M + \frac{J_L}{n} \quad (3.3)$$

$$b_{eff} = nb_M + \frac{b_L}{n} \quad (3.4)$$

Sistema eléctrico. Como se muestra en la figura 3.5, la velocidad del motor esta controlada por la tensión de armadura, la cual se puede modelar con la siguiente ecuación:

$$V_a = IR + L\dot{I} + e_b \quad (3.5)$$

Cuando la armadura esta girando, se induce en ella una tensión proporcional al producto del flujo por la velocidad angular, deduciéndose $e_b = E\omega_M$, donde E es la constante de fuerza contraelectromotriz.

El driver del motor se modela como un amplificador de ganancia K_{act}^1 , entonces $V_a = V_b K_{act}$, donde V_b es la tensión de entrada en el driver.

Las ecuaciones (3.1) y (3.5), se encuentran relacionadas por la ecuación de conversión de energía eléctrica en energía mecánica, la cual esta dada por:

$$T = KI \quad (3.6)$$

donde K es la constante del par motriz. Por tanto, los sistemas descritos se pueden integrar en uno solo juntando las ecuaciones (3.2), (3.5) y (3.6), obteniéndose la siguiente ecuación de estado, no lineal del BRIL:

$$\begin{aligned} \dot{\mathbf{x}} &= F(\mathbf{x}) + G(\mathbf{x})U \\ y &= H(\mathbf{x}) \end{aligned} \quad (3.7)$$

Donde \mathbf{x} es un vector de estado con las siguientes variables:

$$\begin{aligned} x_1 &= q \\ x_2 &= \dot{q} = w \\ x_3 &= I \end{aligned} \quad (3.8)$$

$F(\cdot)$, $G(\cdot)$, $H(\cdot)$ estan descritos por las siguientes ecuaciones:

$$F(\mathbf{x}) = \begin{bmatrix} -\frac{A \sin x_1}{N} - \frac{f^*(nx_2)}{N} - \frac{b_{eff}}{N} x_2 + \frac{K}{N} x_3 \\ -\frac{nE}{L} x_2 - \frac{R}{L} x_3 \end{bmatrix} \quad (3.9)$$

¹Esto se muestra en la sección 6.2.1

$$G(x) = \begin{bmatrix} 0 \\ 0 \\ \frac{K_{ac}}{L} \end{bmatrix}; \quad H(x) = x_1; \quad U = V_b \quad (3.10)$$

donde:

$$\begin{aligned} A &= gL_0(M + \frac{m}{2}) \\ N &= J_{eff} + \frac{1}{n}(ML_0^2 + \frac{2}{5}MR_0^2 + \frac{1}{3}mL_0^2) \end{aligned} \quad (3.11)$$

3.2.2 Modelo del driver y motor

Las ecuaciones del driver del motor se deducen fácilmente haciendo la masa de la esfera y la varilla igual a cero, entonces para el sistema de ecuaciones de estado (3.7) las ecuaciones (3.9) , (3.10) llegan a ser:

$$F(x) = \begin{bmatrix} -\frac{f^*(nx_2)}{J_{eff}} - \frac{b_{eff}}{J_{eff}}x_2 + \frac{K}{J_{eff}}x_3 \\ -\frac{nE}{L}x_2 - \frac{R}{L}x_3 \end{bmatrix} \quad (3.12)$$

$$G(x) = \begin{bmatrix} 0 \\ 0 \\ \frac{K_{ac}}{L} \end{bmatrix}; \quad H(x) = x_1; \quad U = V_b \quad (3.13)$$

donde J_{eff} y b_{eff} vienen dado por las ecuaciones (3.3) y (3.4) .

CAPITULO IV

ALGORITMO DE ENTRENAMIENTO Y SISTEMAS DE IDENTIFICACION USANDO REDES NEURONALES

La identificación y control se realiza usando modelos dinámicos los cuales utilizan como subsistemas a las redes neuronales de alimentación directa multicapa. Este tipo de red es descrita en este capítulo, derivándose también, su algoritmo de entrenamiento denominado “Back-Propagation ¹”. Para la identificación se describen dos esquemas generales utilizando redes neuronales como subsistemas, se muestran las ventajas uno con respecto al otro y los algoritmos para la actualización de los parámetros internos del modelo de identificación utilizado en esta tesis se deriva en el siguiente capítulo.

4.1 Redes neuronales de alimentación directa multicapa

En la figura 2.10 se observa la arquitectura gráfica de una red neuronal multicapa con dos capas ocultas. Esta figura muestra una red neuronal con todas sus conexiones. Esto significa que una neurona de alguna capa de la red esta conectada a todos los nodos/neuronas de las capas anteriores. Una señal sigue a través de la red desde la izquierda a la derecha (hacia adelante) y de una capa a otra. La figura 4.1 muestra una porción de una red neuronal multicapa. En esta red se identifican dos tipos de señales:

1. **Señal Función.** Es una señal de entrada (estímulo); esta señal se propaga a través de la red hacia adelante y emerge en el fin de la red como una señal de salida. Se refiere a esta señal como una “señal función” por dos razones. Primero, la salida de la red es una función útil para una señal de entrada dada. Segundo, por cada neurona de la red donde la señal función pasa, la señal es calculada como una función de las entradas y está asociada a los pesos aplicados a las neuronas.
2. **Señal de Error.** Esta señal se origina en la neurona de salida de la red, propagándose hacia atrás (capa por capa) a través de la red. Se refiere a esta señal como una “señal error”, porque su cálculo por todas las neuronas de la red involucra una dependiente función de error.

¹En adelante se le denominará *retropropagación*.

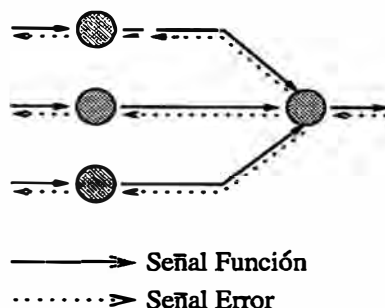


Figura 4.1: Se muestra las direcciones seguidas por dos señales básicas en una red neuronal multicapa, llamadas, propagación hacia adelante para la función señal y retropropagación para la señal error

Las neuronas de salida (nodos computacionales) constituye la capa de salida de la red. Las otras neuronas (nodos computacionales) constituyen las capas ocultas de la red. De esta manera las unidades ocultas no forman parte de la entrada o salida de la red; es por esto su designación de ocultas. La primera capa es alimentada por una capa de entrada constituida por unidades sensoras (nodos fuentes), la salida de la primera capa oculta es aplicada a la próxima capa oculta y así para el resto de la red.

Cada neurona de salida ó oculta de la red esta diseñada para realizar dos cálculos: (1) El cálculo de la señal función que aparece a la salida de la neurona, la cual se expresa como una función continua no lineal de la señal de entrada y los pesos sinápticos asociados con esta neurona. (2) El cálculo de la estimación del vector gradiente en un instante (p.e., el gradiente de la superficie de error con respecto a los pesos conectados a las entradas de una neurona). Este cálculo es necesario para la retropropagación a través de la red.

4.1.1 Notaciones

Para la derivación del algoritmo de retropropagación se presenta un sumario con la notaciones que se usarán en la derivación.

1. Los índices i, j, k se refieren a las distintas neuronas de la red, con señales propagándose a través de la red de izquierda a derecha, la neurona j se encuentra en una capa a la derecha de la neurona i y la neurona k se encuentra en una capa a la derecha de la neurona j cuando la neurona j es una capa oculta.
2. la iteración n se refiere al n –ésimo patrón de entrenamiento (ejemplo) presentado a la red.
3. El símbolo $\xi(n)$ se refiere a la suma instantánea del cuadrado del error en la iteración n . El promedio de $\xi(n)$ sobre todos los valores de n (por ejemplo, todo el conjunto

de entrenamiento) es el error cuadrático promedio ξ_{av} .

4. El símbolo $e_i(n)$ se refiere a la señal de error a la salida de la neurona j para la iteración n .
5. El símbolo $d_i(n)$ se refiere a la respuesta deseada de la neurona j y es usada para calcular $e_i(n)$.
6. El símbolo $y_i(n)$ se refiere a la señal función que aparece a la salida de la neurona j en la iteración n .
7. El símbolo $w_{ij}(n)$ denota el peso sináptico conectado de la salida de la neurona i a la entrada de la neurona j en la iteración n . La corrección aplicada a este peso en la iteración n es denotada por $\Delta w_{ij}(n)$.
8. El nivel de actividad interna de la neurona j en la iteración n es denotada por $\nu_i(n)$; este constituye la señal aplicada a la no linealidad asociada con la neurona j .
9. La función de activación describe la relación funcional de la entrada y salida y esta dada por una no linealidad asociada con la neurona j esta es denotada por $\varphi_j(\cdot)$.
10. El umbral aplicado a la neurona j es denotado por Θ_j ; su efecto es representado por un peso sináptico $w_{j0} = \Theta_j$ conectada a una entrada fija igual a -1.
11. El i –ésimo elemento del vector de entrada (patrón) esta denotado por $x_i(n)$.
12. El k –ésimo elemento del vector de salida (patrón) esta denotado por $o_k(n)$.
13. El parámetro del razón de aprendizaje esta denotado por η .

4.1.2 Derivación del algoritmo de retropropagación

La señal de error a la salida de la neurona j en la iteración n (p.e., presentación del n –ésimo patrón de entrenamiento) está definido por:

$$e_j(n) = d_j(n) - y_j(n), \quad \text{la neurona } j \text{ es un nodo de salida} \quad (4.1)$$

El valor instantáneo del error cuadrático para la neurona j está dado por $\frac{1}{2}e_j^2(n)$. Correspondientemente, el valor instantáneo de la suma de errores cuadráticos $\xi(n)$ se obtiene por la suma de $\frac{1}{2}e_j^2(n)$ sobre todas las neuronas en la capa de salida (las neuronas “visibles”), para las cuales la señal de error puede ser calculada. La suma instantánea de los errores cuadráticos de la red se escribe como sigue:

$$\xi(n) = \frac{1}{2} \sum_{j \in C} e_j^2(n) \quad (4.2)$$

donde el conjunto \mathbf{C} incluye todas las neuronas en la capa de salida de la red. Sea N el número total de patrones (ejemplos) contenidos en el conjunto de entrenamiento. El *error cuadrático promedio* se obtiene por la suma de $\xi(n)$ sobre todos los n ; entonces, normalizando con respecto a N tenemos que

$$\xi_{av}(n) = \frac{1}{N} \sum_{n=1}^N \xi(n) \quad (4.3)$$

La suma instantánea de los errores cuadráticos $\xi(n)$ y por consiguiente el error cuadrático promedio ξ_{av} , es una función de todos los parámetros libres (p.e., los pesos sinápticos y los umbrales) de la red. Aquí, ξ_{av} representa la función de costo que mide el aprendizaje para un conjunto dado de entrenamiento. El objetivo del proceso de aprendizaje es ajustar los parámetros libres de la red de forma tal que minimice ξ_{av} . Para hacer esta minimización se consideró un simple método de entrenamiento, en el cual los pesos se actualizan después de la presentación de un patrón de entrenamiento. El promedio aritmético de estos pesos individuales cambia en todo el conjunto de entrenamiento. Por tanto, este cambio es un estimado del verdadero cambio que podría resultar de la modificación de los pesos que minimizan la función de costo ξ_{av} sobre todo el conjunto de entrenamiento.

Consideremos la figura 4.2, en la cual se muestra a la neurona j alimentada por un conjunto de señales funciones producidas por la capa de neuronas de su izquierda.

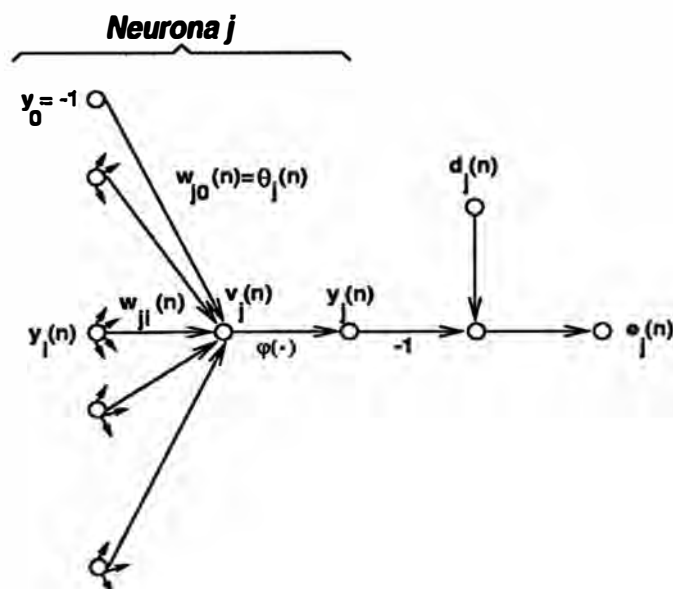


Figura 4.2: Gráfico del flujo de señales de la neurona j

El nivel de actividad interna de la red $v_j(n)$ producida a la entrada de la no linealidad

asociada la neurona j es

$$\nu_j(n) = \sum_{i=0}^p w_{ji}(n) y_i(n) \quad (4.4)$$

donde p es el número total de entradas (excluyendo el umbral) aplicada a la neurona j . El peso sináptico w_{j0} (correspondiente a la entrada fija $y_0 = -1$) es igual al umbral Θ_j aplicada a la neurona j . Así la señal función $y_j(n)$ que aparece a la salida de la neurona j en la interacción n es

$$y_j(n) = \varphi_j(\nu_j(n)) \quad (4.5)$$

El algoritmo de retropropagación aplica una corrección $\Delta w_{ji}(n)$ a los pesos sinápticos $w_{ij}(n)$, la cual es proporcional al gradiente instantáneo $\partial \xi(n) / \partial w_{ji}(n)$. De la regla de la cadena, este gradiente se puede expresar como sigue:

$$\frac{\partial \xi(n)}{\partial w_{ji}(n)} = \frac{\partial \xi(n)}{\partial e_j(n)} \frac{\partial e_j(n)}{\partial y_j(n)} \frac{\partial y_j(n)}{\partial \nu_j(n)} \frac{\partial \nu_j(n)}{\partial w_{ji}(n)} \quad (4.6)$$

El gradiente $\partial \xi(n) / \partial w_{ji}(n)$ representa un *factor de sensibilidad* determinado por la dirección de la búsqueda en el espacio de pesos para el peso sináptico w_{ij} .

Diferenciando ambos lados de la ecuación (4.2) con respecto a $e_j(n)$, se obtiene:

$$\frac{\partial \xi(n)}{\partial e_j(n)} = e_j(n) \quad (4.7)$$

Diferenciando ambos lados de la ecuación (4.1) con respecto a $y_j(n)$:

$$\frac{\partial e_j(n)}{\partial y_j(n)} = -1 \quad (4.8)$$

Diferenciando la ecuación (4.5) con respecto a $\nu_j(n)$ se obtiene:

$$\frac{\partial y_j(n)}{\partial \nu_j(n)} = \varphi'_j(\nu_j(n)) \quad (4.9)$$

Finalmente, diferenciando la ecuación (4.4) con respecto a $w_{ji}(n)$ resulta:

$$\frac{\partial \nu_j(n)}{\partial w_{ji}(n)} = y_i(n) \quad (4.10)$$

Entonces, usando las ecuaciones (4.7) a (4.10) en la ecuación (4.6) resulta:

$$\frac{\partial \xi(n)}{\partial w_{ji}(n)} = -e_j(n) \varphi'_j(\nu_j(n)) y_i(n) \quad (4.11)$$

La corrección $\Delta w_{ji}(n)$ aplicada a $w_{ji}(n)$ es definida por la *regla delta*

$$\Delta w_{ji}(n) = -\eta \frac{\partial \xi(n)}{\partial w_{ji}(n)} \quad (4.12)$$

donde η es una constante que determina la razón de aprendizaje llamada parámetro de razón-apredizaje del algoritmo de retropropagación. El uso del signo menos en la ecuación (4.12) define la gradiente descendente en el espacio de pesos. De las ecuaciones (4.11) y (4.12) tenemos:

$$\Delta w_{ji}(n) = \eta \delta_j(n) y_j(n) \quad (4.13)$$

donde el *gradiente local* es autodefinido por

$$\begin{aligned} \delta_j(n) &= - \frac{\partial \xi(n)}{\partial e_j(n)} \frac{\partial e_j(n)}{\partial y_j(n)} \frac{\partial y_j(n)}{\partial \nu_j(n)} \\ &= e_j(n) \varphi'_j(\nu_j(n)) \end{aligned} \quad (4.14)$$

De acuerdo a la ecuación (4.14), el gradiente local de $\delta_j(n)$ para la neurona de salida j es igual al producto de la correspondiente señal de error $e_j(n)$ por la derivada $\varphi'_j(\nu_j(n))$ de la asociada función de activación.

De las ecuaciones (4.13) y (4.14) se observa que el principal factor involucrado en el cálculo del ajuste de pesos $\Delta w_{ji}(n)$ es la señal de error $e_j(n)$ a la salida de la neurona j . En este contexto se puede identificar dos casos distintos dependiendo de la localización de la neurona j en la red. En el primer caso la neurona j es el nodo de salida; este caso es simple de manejar, porque cada nodo de salida de la red es suministrado con la respuesta deseada del mismo. De esta manera es fácil calcular la señal de error asociada. En el segundo caso la neurona j es un nodo oculto. Todas las neuronas ocultas no son directamente accesibles; ellas comparten la responsabilidad de contribuir con alguna fracción del error a la salida de la red. La pregunta, sin embargo, es conocer cómo penalizar o recompensar a las neuronas ocultas por su responsabilidad compartida.

Caso I: La neurona j es un nodo de salida

Cuando la neurona j está localizada en la capa de salida de la red, esta es suministrada con su respuesta deseada. Así podemos utilizar la ecuación (4.1) para calcular la señal de error $e_j(n)$ asociada con esta neurona (ver la figura 4.2). Habiendo calculado el error $e_j(n)$, es fácil calcular el gradiente local $\delta_j(n)$ usando la ecuación (4.14).

Caso II: La neurona j es un nodo oculto

Cuando la neurona j está localizada en la capa oculta de la red, no hay una respuesta específica para esta neurona. Por tanto, la señal de error para esta neurona oculta podrá ser determinada recursivamente en términos de la señal de error de todas las neuronas que están directamente conectadas con esta neurona oculta. Consideremos la situación descrita en la figura 4.3 que describe a la neurona j como un nodo oculto de la red. De acuerdo a la ecuación (4.14), nosotros podemos definir el gradiente local $\delta_j(n)$ para la

neurona oculta j como:

$$\begin{aligned}\delta_j(n) &= -\frac{\partial \xi(n)}{\partial y_j(n)} \frac{\partial y_j(n)}{\partial v_j(n)} \\ &= -\frac{\partial \xi(n)}{\partial y_j(n)} \varphi'_j(v_j(n)),\end{aligned}\quad \text{La neurona } j \text{ es oculta} \quad (4.15)$$

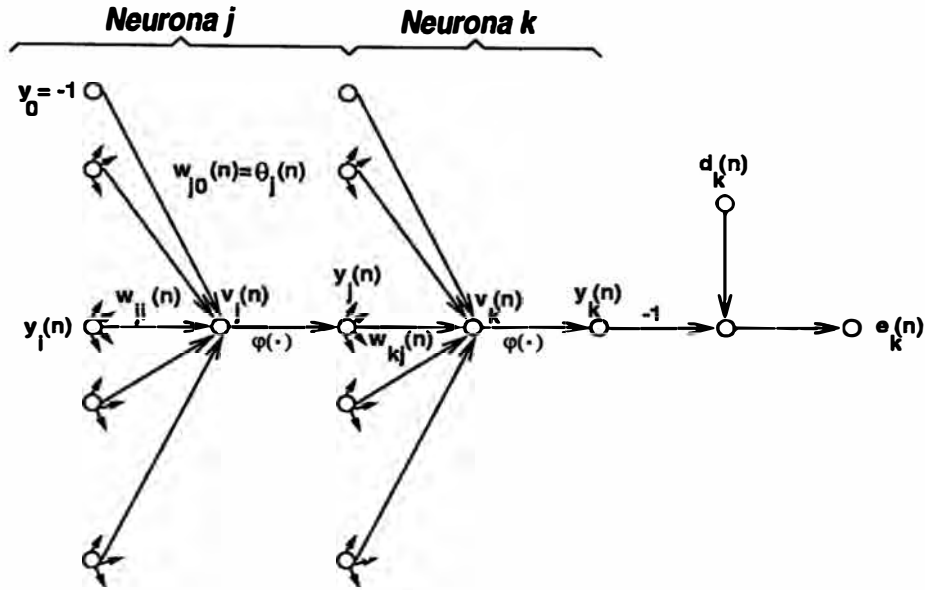


Figura 4.3: Gráfico del flujo de señales de la neurona j

donde, en la segunda línea, se ha hecho uso de la ecuación (4.9). Para calcular la derivada parcial $\partial \xi(n) / \partial y_i(n)$ se puede proceder como sigue. De la figura 4.3, se observa que la relación:

$$\xi(n) = \frac{1}{2} \sum_{k \in C} e_k^2(n), \quad \text{La neurona } k \text{ es un nodo de salida} \quad (4.16)$$

es igual a la ecuación (4.2), excepto por el uso del índice k (Esto se ha realizado para no confundirnos con el índice j que se refiere a la neurona oculta en el segundo caso) en lugar del j . Diferenciando la ecuación (4.16) con respecto a la señal función $y_j(n)$ se obtiene:

$$\frac{\partial \xi(n)}{\partial y_j(n)} = \sum_k e_k \frac{\partial e_k(n)}{\partial y_j(n)} \quad (4.17)$$

Utilizando la regla de la cadena para la derivada parcial $\partial e_k(n) / \partial y_j(n)$ y reescribiendo la ecuación (4.17) en su forma equivalente se obtiene:

$$\frac{\partial \xi(n)}{\partial y_j(n)} = \sum_k e_k(n) \frac{\partial e_k(n)}{\partial v_k(n)} \frac{\partial v_k(n)}{\partial y_j(n)} \quad (4.18)$$

Sin embargo, de la figura 4.3 se nota que

$$\begin{aligned} e_k(n) &= d_k(n) - y_k(n) \\ &= d_k(n) - \varphi_k(\nu_k(n)) \end{aligned} \quad \text{La neurona } k \text{ es un nodo de salida} \quad (4.19)$$

De esta forma

$$\frac{\partial e_k(n)}{\partial \nu_k(n)} = -\varphi'_k(\nu_k(n)) \quad (4.20)$$

De la figura 4.3 se observa que el nivel de actividad interna de la neurona k en la red es:

$$\nu_k(n) = \sum_{j=0}^q w_{kj} y_j(n) \quad (4.21)$$

donde q es el número total de entradas (excluyendo al umbral) aplicadas a la neurona k . El peso sináptico $w_{k0}(n)$ es igual al umbral $\Theta_k(n)$ aplicado a la neurona k , y su correspondiente entrada y_0 se fija al valor de -1 . Diferenciando la ecuación (4.21) con respecto a $y_i(n)$ resulta:

$$\frac{\partial \nu_k(n)}{\partial y_j(n)} = w_{kj}(n) \quad (4.22)$$

Así, usando las ecuaciones (4.19) y (4.22) en (4.18) se obtiene la derivada parcial deseada:

$$\begin{aligned} \frac{\partial \xi(n)}{\partial y_j(n)} &= - \sum_k e_k(n) \varphi'_k(\nu_k(n)) w_{kj}(n) \\ &= - \sum_k \delta_k(n) w_{kj}(n) \end{aligned} \quad (4.23)$$

donde, en la segunda línea se ha usado la definición del gradiente local $\delta_k(n)$ dada por la ecuación (4.14), con el índice k sustituido por el índice j .

Finalmente, usando la ecuación (4.23) en (4.15) y reordenando los términos en esta ecuación, se obtiene el gradiente local $\delta_j(n)$ para la neurona oculta j :

$$\delta_j(n) = \varphi'_j(\nu_j(n)) \sum_k \delta_k(n) w_{kj}(n), \quad \text{La neurona } j \text{ es oculta} \quad (4.24)$$

El factor $\varphi'_j(\nu_j(n))$ involucrado en el cálculo del gradiente local $\delta_j(n)$ en la ecuación (4.24) depende sólo de la función de activación asociada con la neurona oculta j . El factor restante involucrado en este cálculo, específicamente, la sumatoria sobre k , depende de dos términos. El primero de ellos es $\delta_k(n)$, el cual requiere conocer la señal de error $e_k(n)$, para todas las neuronas que se encuentran en el inmediato lado derecho de la neurona oculta j y que están directamente conectadas a la neurona j ; ver la figura 4.3. El segundo término w_{kj} , consiste de los pesos sinápticos asociados con tales conexiones.

Ahora se puede resumir las relaciones que se han derivado para el algoritmo de retropropagación. Primero, la corrección $\Delta w_{ji}(n)$ aplicada a los pesos sinápticos que conectan la neurona i con la neurona j se define por la siguiente regla delta:

$$\begin{bmatrix} \text{Corrección} \\ \text{peso} \\ \Delta w_{ji} \end{bmatrix} = \begin{bmatrix} \text{parámetro} \\ \text{razón aprendizaje} \\ \eta \end{bmatrix} \cdot \begin{bmatrix} \text{gradiente} \\ \text{local} \\ \delta_j(n) \end{bmatrix} \cdot \begin{bmatrix} \text{señal entrada} \\ \text{neurona } j \\ y_i(n) \end{bmatrix} \quad (4.25)$$

Segundo, para el gradiente local $\delta_j(n)$:

1. Si la neurona j es un nodo de salida, $\delta_j(n)$ es igual al producto de la derivada $\varphi'_j(\nu_j(n))$ y la señal error $e_j(n)$, las cuales están asociadas con la neurona j ; ver ecuación (4.14).
2. Si la neurona j es un nodo oculto, $\delta_j(n)$ es igual al producto de la derivada asociada $\varphi'_j(\nu_j(n))$ por la suma de los δ 's calculados para las neuronas que se encuentran en la próxima capa de salida u oculta y están conectadas a la neurona j (ver ecuación (4.24)).

4.1.3 No linealidad sigmoideal

El cálculo del δ para cada neurona en la red multicapa requiere el conocimiento de la derivada de la función de activación $\varphi(\cdot)$ asociada con esta neurona. Para que esta derivada exista se requiere que la función $\varphi(\cdot)$ sea continua y *diferenciable*. Este es el único requisito que la función de activación debe satisfacer. Un ejemplo de una función no lineal continuamente diferenciable utilizada en las redes multicapa es la *no linealidad sigmoideal*. Una forma particular de cómo es definida para la neurona j es por medio de la *función logistic*:

$$\begin{aligned} y_i(n) &= \varphi_j(\nu_j(n)) \\ &= \frac{1}{1 + e^{-\nu_j(n)}}, \quad -\infty < \nu_j(n) < \infty \end{aligned} \quad (4.26)$$

donde $\nu_j(n)$ es el nivel de actividad interna de la red para la neurona j . De acuerdo a esta no linealidad, la amplitud de la salida se encuentra entre el rango $0 \leq \nu_j(n) \leq 1$. Otro tipo de la señal sigmoideal es la *tangente hiperbólica*, la cual es antisimétrica con respecto al origen; la amplitud de salida se encuentra entre $-1 \leq \nu_j(n) \leq 1$.

Diferenciando ambos lados de la ecuación (4.26) con respecto a $\nu_j(n)$ se obtiene:

$$\begin{aligned} \frac{\partial y_i(n)}{\partial \nu_j(n)} &= \varphi'_j(\nu_j(n)) \\ &= \frac{e^{-\nu_j(n)}}{[1 + e^{-\nu_j(n)}]^2}, \quad -\infty < \nu_j(n) < \infty \end{aligned} \quad (4.27)$$

Utilizando la ecuación (4.26) para eliminar el término exponencial $e^{-\nu_j(n)}$ en la ecuación (4.27), se puede expresar la derivada $\varphi'_j(\nu_j(n))$ como:

$$\varphi'_j(\nu_j(n)) = y_j(n)[1 - y_j(n)] \quad (4.28)$$

Para una neurona j localizada en la capa de salida, $y_i(n) = o_j(n)$. De esta manera, el gradiente local para la neurona j como

$$\begin{aligned} \delta_j(n) &= e_j(n)\varphi'_j(\nu_j(n)) \\ &= [d_j(n) - o_j(n)]o_j(n)[1 - o_j(n)]. \quad \text{La neurona } j \text{ es un nodo de salida} \end{aligned} \quad (4.29)$$

donde $o_j(n)$ es la señal función a la salida de la neurona j y $d_j(n)$ es su respuesta deseada. Para una arbitraria neurona oculta el gradiente local se puede expresar como

$$\begin{aligned} \delta_j(n) &= \varphi'_j(\nu_j(n)) \sum_k \delta_k(n)w_{kj}(n) \\ &= y_j(n)[1 - y_j(n)] \sum_k \delta_k(n)w_{kj}(n). \quad \text{La neurona } j \text{ es un nodo oculto} \end{aligned} \quad (4.30)$$

4.1.4 Modos de entrenamiento Pattern y Batch

En aplicaciones prácticas del algoritmo de retropropagación, el aprendizaje se realiza presentado a la red un conjunto de ejemplos para el entrenamiento de la red. La presentación de un conjunto completo de entrenamiento de la red se denomina *época*. El proceso de aprendizaje se mantiene época por época hasta que los pesos sinápticos y los umbrales de la red se estabilizan, y el promedio cuadrático del error sobre un conjunto de entrenamiento converja a un mínimo valor. Es una buena práctica presentar en forma aleatoria los ejemplos de entrenamiento de la red, desde una época hacia la siguiente. Este entrenamiento aleatorio tiende a ser la búsqueda en el espacio de los pesos de una forma estocástica sobre los ciclos de aprendizaje, esto deja la posibilidad de ciclos límites en la evolución del vector de pesos sinápticos.

Para un conjunto de entrenamiento dado, el aprendizaje de retropropagación puede proceder de dos maneras básicas:

Modo Pattern

Cuando el aprendizaje de retropropagación se realiza en modo pattern los pesos son actualizados después de la presentación de cada ejemplo de entrenamiento; este es el mejor modo de operación para el cual el algoritmo de retropropagación aquí presentado, se aplica. Para ser explícitos, consideraremos que una época está formada por un conjunto de ejemplos de entrenamiento (patrones) ordenados como $[\mathbf{x}(1), d(1)], \dots, [\mathbf{x}(N), d(N)]$. El primer ejemplo $[\mathbf{x}(1), d(1)]$ de la época se presenta a la red y se realizan los cálculos de propagación hacia adelante y de realimentación en la red; esto da como resultado un ajuste de los pesos sinápticos y umbrales de la red.

Luego, el segundo ejemplo $[x(2),d(2)]$ de la época se presenta a la red, repitiéndose los cálculos hacia adelante, de realimentación y un nuevo ajuste de los pesos sinápticos y umbrales de la red. Este proceso se repite hasta que el último ejemplo de la época de entrenamiento se presente a la red. Denotando a Δw_{ji} como la actualización de los pesos sinápticos después de la presentación del patrón n , entonces el cambio promedio neto de los pesos de la red $\Delta \hat{w}_{ji}$ está dado por:

$$\begin{aligned}\Delta \hat{w}_{ji} &= \frac{1}{N} \sum_{n=1}^N \Delta w_{ji}(n) \\ &= -\frac{\eta}{N} \sum_{n=1}^N \frac{\partial \xi(n)}{\partial w_{ji}(n)} \\ &= -\frac{\eta}{N} \sum_{n=1}^N e_j(n) \frac{\partial e_j(n)}{\partial w_{ji}}\end{aligned}\quad (4.31)$$

donde en la segunda y tercera líneas se hace uso de las ecuaciones (4.12) y (4.2) respectivamente.

Modo Batch.

Cuando el aprendizaje de retropropagación se realiza en modo batch, la actualización de los pesos se realiza después de la presentación de todos los ejemplos de entrenamiento que constituyen una época. Para una época particular, se define la función de costo como el promedio de los errores cuadráticos de las ecuaciones (4.2) y (4.3) y aquí se reproduce en la siguiente forma compuesta:

$$\xi_{av} = \frac{1}{2N} \sum_{n=1}^N \sum_{j \in C} e_j^2(n) \quad (4.32)$$

donde la señal de error $e_j(n)$ pertenece a la salida de la neurona j para el ejemplo de entrenamiento n y es definida por la ecuación (4.1). En la ecuación (4.32) la suma interna con respecto a j se realiza sobre todas las neuronas de la capa de salida de la red, mientras la sumatoria externa con respecto a n se realiza sobre todo el conjunto de entrenamiento en la época disponible. Dada una razón de aprendizaje η , el ajuste de pesos sinápticos aplicado a w_{ji} , se define por la siguiente regla delta

$$\begin{aligned}\Delta w_{ji} &= -\eta \frac{\partial \xi_{av}}{\partial w_{ji}} \\ &= -\frac{\eta}{N} \sum_{n=1}^N e_j(n) \frac{\partial e_j(n)}{\partial w_{ji}}\end{aligned}\quad (4.33)$$

Para calcular la derivada parcial $\partial e_j(n)/\partial w_{ji}$ se procede de similar manera como en el caso anterior. De acuerdo a la ecuación (4.33), el ajuste de pesos Δw_{ji} en el modo

batch se realiza sólo después de la presentación de todo el conjunto de entrenamiento a la red neuronal.

Comparando las ecuaciones (4.31) y (4.33), se observa claramente que el cambio promedio del peso $\Delta\hat{w}_{ji}$ realizado en el modo de entrenamiento patrón es diferente al correspondiente cambio Δw_{ji} realizado por el modo batch. Presumiblemente, por alguna reducción en el error cuadrático promedio ξ_{av} que resulta de la presentación de todo el conjunto de entrenamiento a la red, $\Delta\hat{w}_{ji}$ para el modo patrón representa una estimación de Δw_{ji} para el modo batch.

Desde un punto de vista operacional “on-line”, el modo de entrenamiento patrón es preferido sobre el modo batch porque requiere menos almacenamiento local para cada conexión sináptica. Además, dado que los patrones son presentados a la red de una manera aleatoria, el uso del modo patrón para la actualización de los pesos, hace que la búsqueda en el espacio de pesos se realice de una manera estocástica; de esta manera es menos probable que el algoritmo de retropropagación quede atrapado en un mínimo local. El uso del modo de entrenamiento batch provee una estimación más precisa del vector gradiente. En el análisis final, sin embargo, la relativa efectividad de los dos modos de entrenamiento depende del problema en sí.

4.1.5 Criterio de fin de entrenamiento

No siempre puede predecirse que el algoritmo de entrenamiento vaya a converger y no hay un criterio bien definido para detener su operación. Más bien, hay algunos criterios razonables, que cuentan con méritos prácticos, los cuales se puede usar para terminar el ajuste de los pesos. Para formular tales criterios, hay que pensar en términos de la única propiedad de un mínimo local o global de la superficie de error. Dejemos que el vector de pesos w^* denote un mínimo, sea éste local o global. La condición necesaria para que w^* sea un mínimo es que el gradiente del vector $\mathbf{g}(\mathbf{w})$ (p.e. primera derivada parcial) de superficie de error con respecto al vector de pesos \mathbf{w} sea cero en $w = w^*$. Por consiguiente, se puede formular un criterio sensible de convergencia para el aprendizaje de retropropagación como sigue:

El algoritmo de retropropagación puede converger cuando la norma Euclidiana del vector gradiente alcance pequeños umbrales de gradiente.

El inconveniente de este criterio de convergencia es que para acertados ensayos, el tiempo de aprendizaje puede ser largo. También se requiere el cálculo del vector gradiente $\mathbf{g}(\mathbf{w})$.

Otra propiedad, que se puede utilizar, es el hecho que la función de costo o la medición de error $\xi_{av}(w)$ es estacionaria en el punto $w = w^*$. Por tanto, se puede sugerir diferentes criterios de convergencia:

El algoritmo de retropropagación puede converger cuando la razón absoluta del cambio del error cuadrático promedio por época es suficientemente pequeño.

Tipicamente, la razón de cambio en el error cuadrático promedio se considera a ser suficientemente pequeño si éste se encuentra en el rango de 0.1 a 1 por ciento por época; algunas veces se usa un valor pequeño como 0.01 por ciento por época.

Una variación de este segundo criterio para la convergencia del algoritmo requiere que el máximo valor del error cuadrático promedio $\xi_{av}(w)$ sea igual o menor que un pequeño umbral.

El algoritmo de retropropagación termina con un vector de pesos w_{final} cuando $\|g(w_{final})\| \leq \epsilon$, donde ϵ es un pequeño umbral de gradiente, o $\xi_{av}(w_{final}) \leq \tau$, donde τ es un pequeño umbral de error de energía.

Otro criterio útil para la convergencia es el siguiente. Después de cada iteración de aprendizaje, se comprueba la generalización de la red. El proceso de aprendizaje termina cuando la generalización es satisfactoria o cuando es aparente que la generalización ha llegado a un límite.

4.2 Resumen del algoritmo de retropropagación

La figura 2.10 presenta una red neuronal de alimentación directa multicapa. La arquitectura del aprendizaje de retroalimentación en donde se incorporan las fases de alimentación hacia adelante y de realimentación involucradas en el proceso de aprendizaje, se presentan en la figura 4.3. La red multicapa vista en la parte superior de la figura cuenta con la fase de la alimentación hacia adelante. Las notaciones usadas en esta parte de la figura son como sigue:

- $w^{(l)}$ — vector de pesos sinápticos de una neurona en la capa l
- $\Theta^{(l)}$ — umbral de una neurona en la capa l
- $v^{(l)}$ — vector de los niveles de actividad interna de las neuronas en la capa l
- $y^{(l)}$ — vector de las señales funciones de las neuronas en la capa l

El índice de la capa l se extiende desde la capa de entrada ($l = 0$) a la capa de salida ($l = L$); en la figura 4.4 se tiene que $L=3$; se refiere a L como la profundidad de la red. En la parte mas baja de la figura tenemos la fase de realimentación, la cual es llamada red de sensibilidad para el cálculo de los gradientes locales. Las notaciones utilizadas en esta parte de la figura son las siguientes:

- $\delta^{(l)}$ — vector de los gradientes locales de las neuronas en la capa l .
- e — vector de error el cual tiene como elementos a e_1, e_2, \dots, e_q .

Anteriormente mencionamos que el método de actualización de pesos patrón por patrón es preferible para una implementación "on-line" del algoritmo de retropropagación. Para este modo de operación, los ciclos algorítmicos a través del dato de entrenamiento $\{[x(n), d(n)]; n = 1, 2, \dots, N\}$ es como sigue:

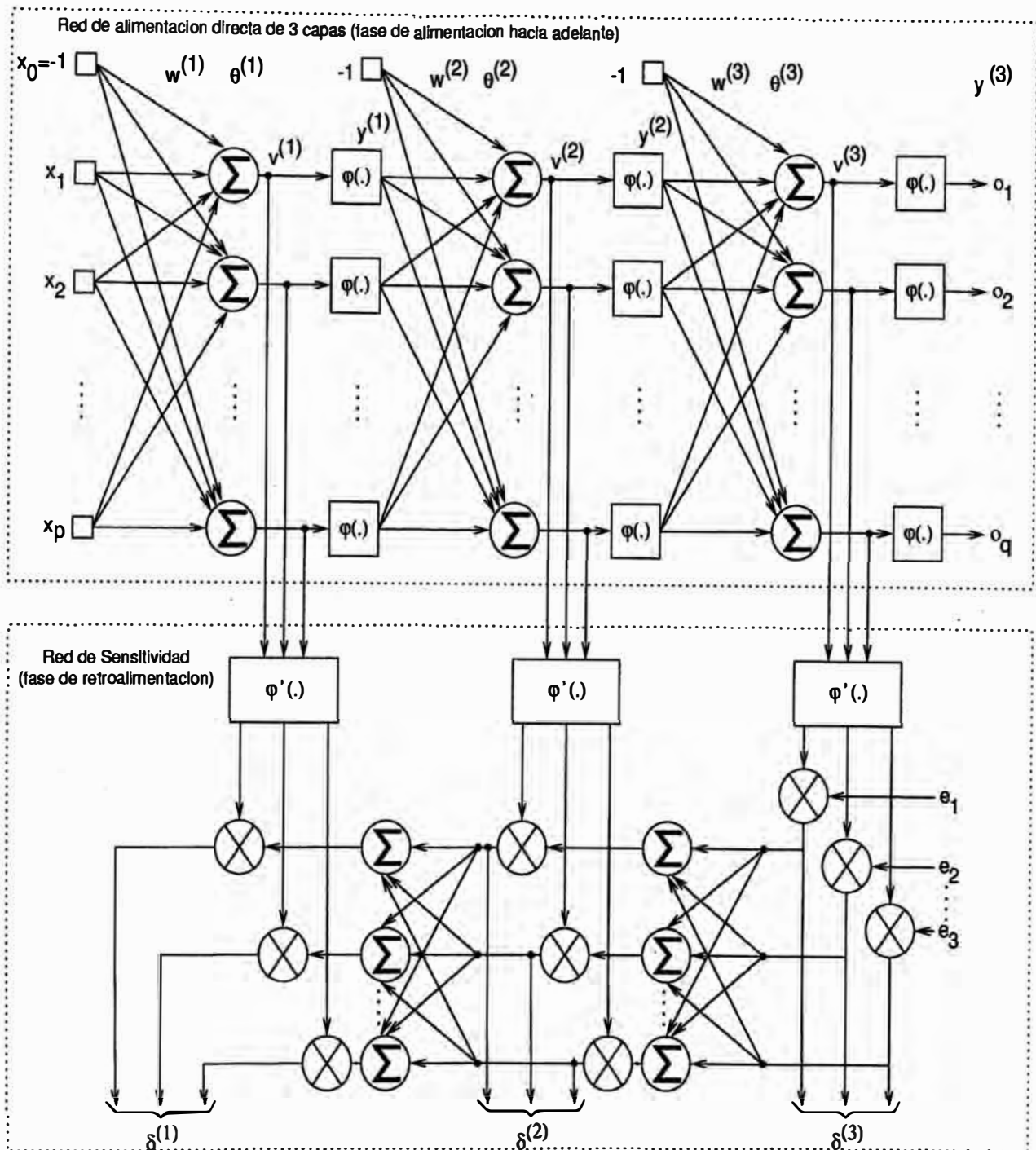


Figura 4.4: Arquitectura gráfica de una red neuronal de alimentación directa con tres capas con su asociada red de sensibilidad.

1. Inicialización.- Iniciar con una razonable configuración de la red y configurar todos los pesos sinápticos y niveles umbrales de la red en pequeños números aleatorios uniformemente distribuidos.

2. Presentación de los ejemplos de entrenamiento.- Presentar a la red una época de ejemplos de entrenamiento. Cada ejemplo en el conjunto ésta ordenado de alguna manera, realizando las secuencias de cálculo hacia adelante y de realimentación bajo los puntos 3 y 4 respectivamente.

3. Cálculo hacia adelante.- Denotemos a un ejemplo de entrenamiento de una época por $[x(n), d(n)]$. Luego, si el vector de entrada $x(n)$ se aplica a los nodos sensores de la capa de entrada, la respuesta deseada $d(n)$ se presenta en los nodos de cálculo de la capa de salida. Los cálculos de los potenciales de activación y de las funciones señales de la red se deben por computar hacia adelante, a través de la red y capa por capa. El nivel de actividad interna $v_j^{(l)}(n)$ para la neurona j en la capa l es:

$$v_j^{(l)}(n) = \sum_{i=0}^p w_{ji}^{(l)}(n) y_i^{(l-1)}(n)$$

donde $y_i^{(l-1)}(n)$ es la señal función en la iteración n de la neurona i en capas previas $l-1$ y $w_{ji}^{(l)}(n)$ es el peso sináptico de la neurona j en la capa l que es alimentada desde la neurona i en la capa $l-1$. Para $i=0$, se tiene que $y_0^{(l-1)}(n) = -1$ y $w_{j0}^{(l)}(n) = \Theta_j^{(n)}$, donde $\Theta_j^{(n)}$ es el umbral aplicado a la neurona j en la capa l . Asumiendo el uso de la función logístico para la no linealidad sigmoideal, entonces la señal función (salida) de la neurona j en la capa l es

$$y_j^{(l)}(n) = \frac{1}{1 + e^{-v_j^{(l)}(n)}}$$

Si la neurona j es la primera capa oculta (p.e., $l=1$), tendremos que

$$y_j^{(0)}(n) = x_j(n)$$

donde $x_j(n)$ es el j -ésimo elemento del vector de entrada $x(n)$. Si la neurona j es la capa de salida (p.e., $l=L$), entonces

$$y_j^{(L)}(n) = o_j(n)$$

El error de salida

$$e_j(n) = d_j(n) - o_j(n)$$

donde $d_j(n)$ es el j -ésimo elemento del vector de respuestas deseadas $d(n)$.

4. Cálculo de realimentación.- Primero debemos calcular los δ 's (p.e., los gradientes locales) de la red hacia atrás y capa por capa como sigue:

$$\begin{aligned}\delta_j^{(L)}(n) &= e_j^{(L)}(n)o_j(n)[1 - o_j(n)] && \text{neurona } j - \text{ capa salida } L \\ \delta_j^{(l)}(n) &= y_j^{(n)}(n)[1 - y_j^{(l)}(n)] \sum_k \delta_k^{(l+1)}(n)w_{kj}^{(l+1)}(n) && \text{neurona } j - \text{ capa oculta } l\end{aligned}$$

El ajuste de los pesos sinápticos de la red se realiza de acuerdo a la regla delta siguiente:

$$w_{ji}^{(l)}(n+1) = w_{ji}^{(l)}(n) + \eta \delta_j^{(l)}(n) y_i^{(l-1)}(n)$$

donde η es la razón de aprendizaje de la red.

5. Iteración.- La iteración de los cálculos se realiza para nuevas épocas de ejemplos de entrenamiento a la red hasta que los parámetros libres de dicha red se estabilicen y el error cuadrático promedio ξ_{av} se calcula sobre un conjunto de entrenamiento hasta un valor pequeño aceptable. El orden de presentación de los ejemplos de entrenamiento deben ser aleatorios por época. La razón de aprendizaje se ajusta convencionalmente (y usualmente disminuye) con el incremento del número de iteraciones.

4.3 Modelos de identificación

En [4] se presentan cuatro modelos generales para la representación de plantas SISO. Dependiendo de los subsistemas internos del modelo, estos pueden representar a sistemas lineales así como también a sistemas no lineales. De acuerdo a la complejidad del modelo, se sugiere que éstos pueden contener como subsistemas redes neuronales que han identificado una determinada no linealidad del modelo.

El problema de identificación consiste en configurar un adecuado modelo parametrizado de identificación y ajustar los parámetros del modelo para optimizar una función de costo basada en el error entre la planta y la salida del modelo de identificación. Desde que las funciones no lineales en la representación de la planta se asumen que pertenecen a una conocida clase de redes neuronales ($\Pi_{i_1, i_2, \dots, i_{N+1}}^N$) en el dominio de interés, la estructura del modelo de identificación se elige idéntico al de la planta. Las matrices de pesos de la red neuronal en el modelo de identificación existen de modo tal que, para alguna condición inicial, ambas, la planta y el modelo, tienen una salida similar para una entrada específica. Así, el procedimiento de identificación consiste en ajustar los parámetros de la red neuronal en el modelo usando el método descrito en la sección 4.2 basado en el error entre la planta y el modelo de salida. Sin embargo, hay que tomar precauciones para asegurar que los parámetros del modelo de identificación convergan a sus valores deseados.

4.3.1 Modelo de identificación paralelo

En la figura 4.5 se observa una planta representada por la ecuación (4.34). Para identificar tal planta se puede asumir que el modelo la estructura de identificación visto

en la figura 4.5 será descrito por la ecuación (4.35).

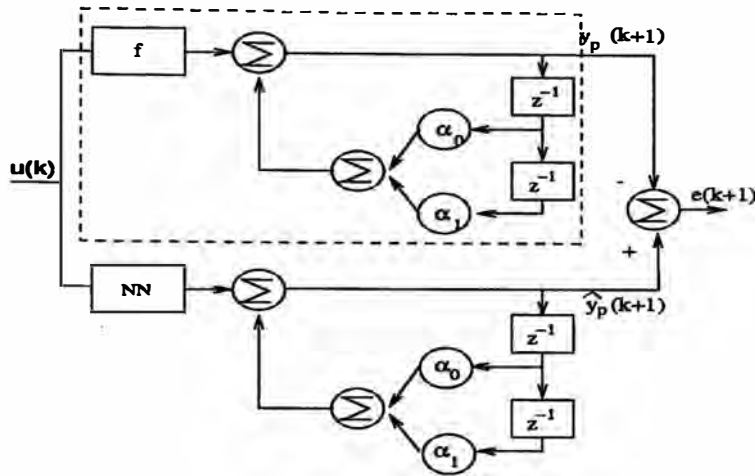


Figura 4.5: Modelo de identificación Paralelo

$$y_p(k+1) = \alpha_0 y_p(k) + \alpha_1 y_p(k-1) + G(u[k]) \quad (4.34)$$

$$\hat{y}_p(k+1) = \hat{\alpha}_0 \hat{y}_p(k) + \hat{\alpha}_1 \hat{y}_p(k-1) + G(u[k]) \quad (4.35)$$

La identificación en este modelo involucra la estimación de los parámetros $\hat{\alpha}_i$ así como también los pesos de la red neuronal usando una retropropagación dinámica [4], [5] basado en el error $e(k)$ entre la salida del modelo $\hat{y}_p(k)$ y la salida actual $y_p(k)$.

Se asume que la planta tiene una entrada y salida acotada (BIBO). De esta manera todas las señales en la planta estarán uniformemente acotadas. En contraste, la estabilidad del modelo de identificación con una red neuronal no puede ser asegurada. Así, si se usa el modelo paralelo, no hay garantía que los parámetros convergan o que la salida de error tienda a cero. Según la bibliografía [4], a pesar de dos décadas de trabajo, las condiciones bajo la cual el modelo paralelo tienda a converger en el caso lineal, están siendo estudiadas actualmente. El modelo serie-paralelo muestra otra alternativa en la realización de la identificación, esta se describe en la sección siguiente.

4.3.2 Modelo serie-paralelo

En contraste al modelo paralelo descrito anteriormente, en el modelo serie-paralelo la salida de la planta (y no la del modelo de identificación) se realimenta dentro del modelo de identificación tal como se observa en la figura 4.6. Esto implica que en este caso, el modelo de identificación tiene la forma

$$\hat{y}_p(k+1) = \hat{\alpha}_0 y_p(k) + \hat{\alpha}_p(k-1) + N[u(k)]. \quad (4.36)$$

El modelo serie-paralelo tiene varias ventajas sobre el modelo paralelo. Desde que se asume que la planta es un sistema estable con salida y entrada acotadas, todas

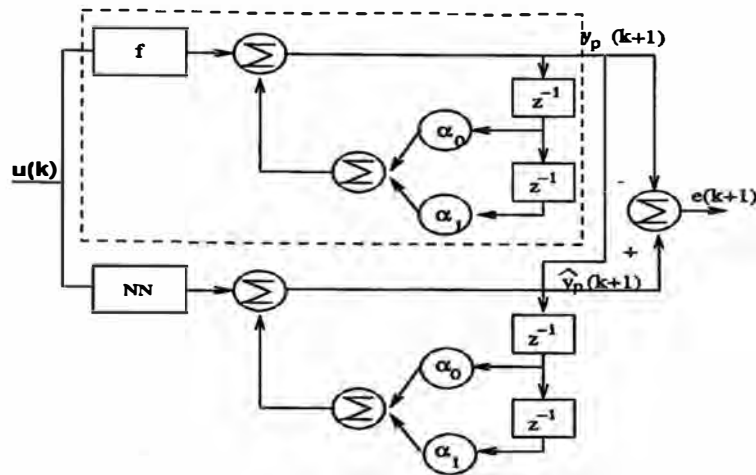


Figura 4.6: Modelo de identificación Serie-Paralelo

las señales usadas en el procedimiento de identificación (por ejemplo, entradas a la red neuronal) son acotadas. Además, desde que no existe un lazo de realimentación en el modelo, el algoritmo de entrenamiento de retropropagación estática puede ser usado para ajustar los parámetros de la red, reduciendo sustancialmente los cálculos computacionales. Si se asume que el valor de salida tiende asintóticamente a pequeños valores tal que $y_p(k) \approx \hat{y}_p(k)$, entonces el modelo serie-paralelo puede ser reemplazado por el paralelo sin serias consecuencias. Solo habría una implicación práctica si el modelo de identificación fuera usado off line, el cual es nuestro caso.

CAPITULO V

IDENTIFICACION Y CONTROL USANDO REDES NEURONALES

En este capítulo se realizan las simulaciones de identificación de los parámetros lineales y no lineales del BR1L utilizando redes neuronales; se a partido del caso donde solo se identifican los parámetros no linealidades del motor hasta el caso de identificación simultánea de los parámetros lineales y no lineales.

En la parte del control, el sistema BR1L se linealiza por una realimentación de compensación de los parámetros no lineales utilizando redes neuronales; el sistema BR1L compensado (lineal) es controlado por un controlador PID, el cual es comparado con controles PID simple y realimentación de estado sin utilizar la realimentación de compensación.

Para las simulaciones de identificación de los parámetros lineales y no lineales se a considerado un modelo simplificado y discreto del sistema; para el control se utiliza el modelo BR1L continuo. Los algoritmos de identificación se desarrollaron en “Matlab” y en “C”. En este último caso, para aumentar la velocidad de simulación.

5.1 Identificación de los parámetros del motor utilizando redes neuronales

Para la identificación de los parámetros lineales y no lineales del motor, se ha considerado un modelo simplificado, en el cual se desprecia su inductancia. El modelo discreto Motor-Actuador se deduce a partir de las ecuaciones (3.12) y (3.13) haciendo $L=0$, obteniéndose la siguiente ecuación diferencial:

$$J_{eff} \dot{\omega} = \frac{V_b K K_{act}}{R} - (b_{eff} + \frac{nEK}{R}) \omega - f^*(n\omega) \quad (5.1)$$

Las señales en tiempo discreto surgen si el sistema involucra la operación de muestro de señales continuas. Nosotros definiremos ω_k como la velocidad angular muestreada en el k - ésimo instante de tiempo. Si $\dot{\omega} = \frac{\omega_{k+1} - \omega_k}{T_s}$, siendo T_s el periodo de muestreo, entonces de (5.1) tenemos:

$$\omega_{k+1} = \left[1 - \frac{T_s}{J_{eff}} \left(b_{eff} + \frac{nEK}{R} \right) \right] \omega_k + \frac{KK_{act}T_s}{RJ_{eff}} \left[V_{bk} - \frac{Rf^*(n\omega_k)}{KK_{act}} \right] \quad (5.2)$$

Si $f^*(n\omega_k) = C \text{Sgn}(n\omega_k)$, entonces:

$$\omega_{k+1} = \left[1 - \frac{T_s}{J_{eff}} \left(b_{eff} + \frac{nEK}{R} \right) \right] \omega_k + \frac{KK_{act}T_s}{RJ_{eff}} \left[V_{bk} - \frac{RC \text{Sgn}(n\omega_k)}{KK_{act}} \right] \quad (5.3)$$

Para las simulaciones, la no linealidad $f(n\omega_k)$ es modelada como $f(n\omega_k) = \frac{RC}{KK_{act}} \text{Sgn}(n\omega_k)$. Esto se deduce en la sección 6.2.2. El modelo continuo del motor, ecuación (5.1), se implementa en Simulink¹ tal como se muestra en la figura 5.2. La respuesta de este sistema para una entrada senoidal de amplitud 0.5 Vol y una frecuencia angular de 3 Rad s^{-1} se presenta en la figura 5.1, donde se muestra el efecto de la no linealidad cuando hay un cruce por cero. Es claro observar en la figura 5.2 que en el modelo simplificado del sistema tomado, el torque de fricción no lineal $f^*(n\omega_k)$ puede ser trasladado junto a la tensión de entrada del sistema. Entonces, la eliminación de esta no linealidad, puede ser realizada por compensación. Es decir, generando una tensión de igual magnitud que $f(n\omega_k)$ pero de signo contrario, de tal manera que al adicionarla a la tensión de entrada, elimine el torque de fricción no lineal.

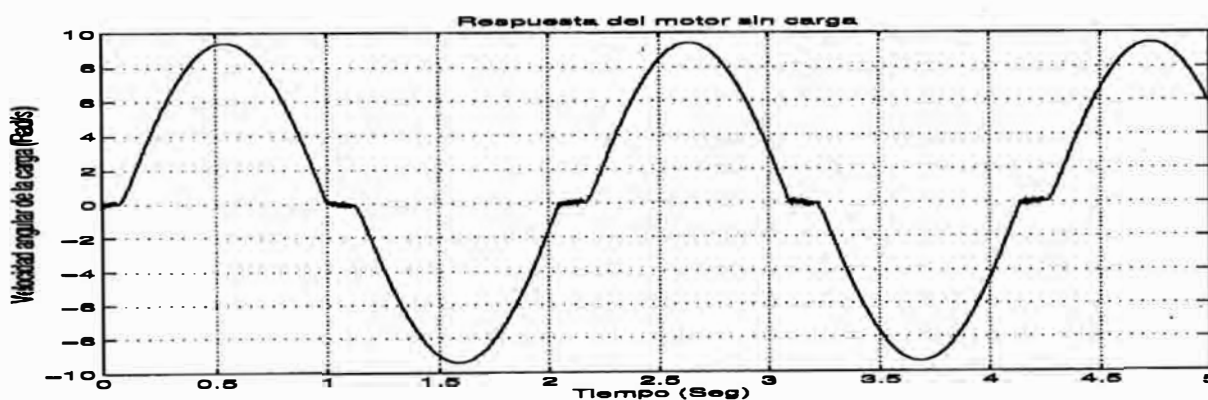


Figura 5.1: Respuesta del sistema para una entrada $0.5 \sin(3t)$

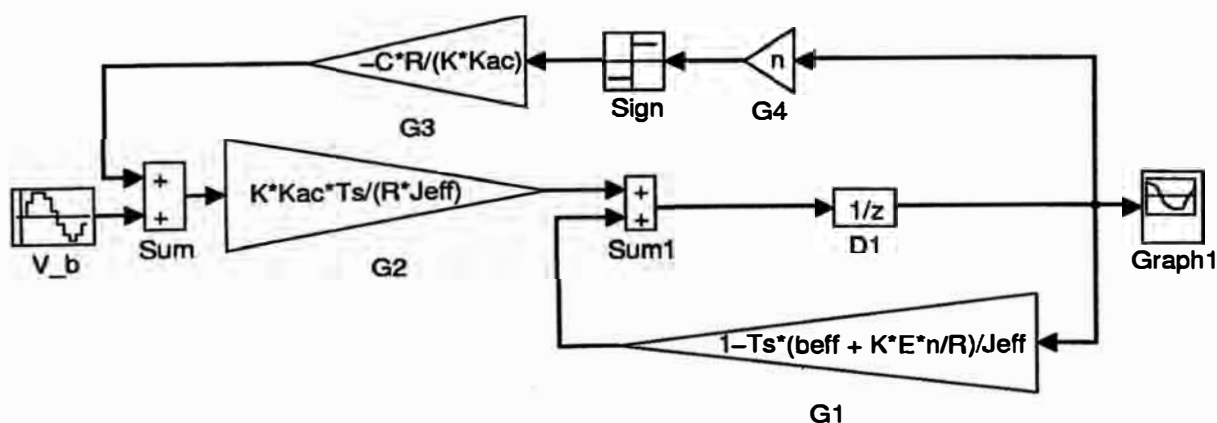


Figura 5.2: Modelo discreto del motor

¹Herramienta del programa matemático Matlab.

de entrada y salida del sistema, respectivamente. Para nuestro sistema la tensión de entrada se encuentra entre $[-1.53, 1.37]$ Volt. Se realizaron experiencias con tres tipos de señales: (1) Escalones con amplitudes variando dentro del rango de entrada y tiempo de duración variable para cada escaln; (2) Escalones con amplitudes variando dentro del rango de entrada y tiempo de duración fijo; (3) Sinusoides con amplitudes variables y periodo fijo. Para el entrenamiento se eligió el 2^{do} tipo de señal, el criterio de elección fue la velocidad de aprendizaje de la red ³, el periodo de las señales de entrenamiento se eligió por el método de prueba y error, teniendo como valor inicial la frecuencia de resonancia del BR1L linealizado alrededor de $q = 0$.

Se eligió 30 valores fijos de tensiones uniformemente distribuidos para el conjunto \mathbb{I} , los cuales produjeron un conjunto de velocidades $\mathbb{S} \in [-37, 37]$ Rad s⁻¹ que es el rango de salida. Por consiguiente, la señal de excitación consiste de un tren de escalones cuyas amplitudes se encuentran aleatoriamente distribuidas en cada conjunto \mathbb{I} . Cada escalón tiene una duración de 0.4 seg, produciendo una velocidad que incluye un estado transitorio y estacionario como se muestra en la figura 5.4.

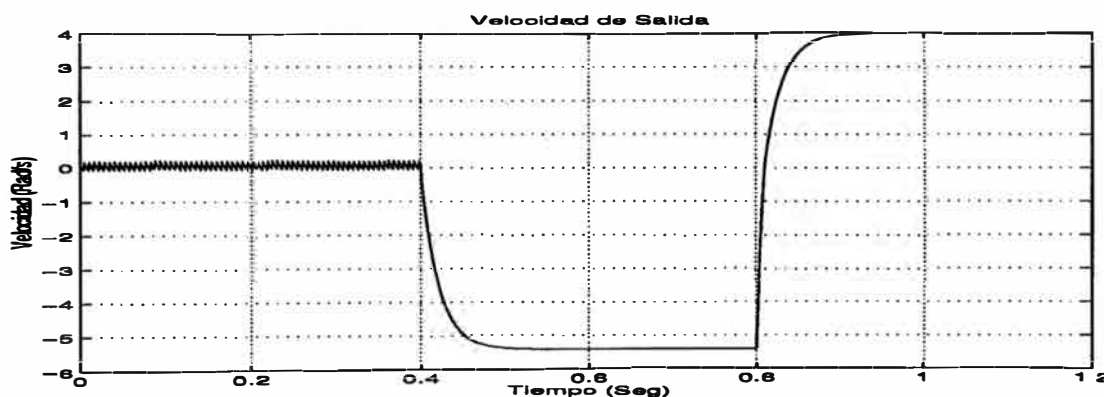


Figura 5.4: Respuesta del sistema para un tren de escalones

El sistema se discretizó con un periodo de muestro $T_s = 0.001$ seg y se tomaron los datos de la tabla 6.2.3 para calcular los valores nominales del sistema a simular los cuales se muestran en la tabla 5.1.1. El entrenamiento se ha dividido en fases. En cada fase, la tensión de entrada del sistema esta constituida por un mismo tren de escalones cuyas amplitudes están distribuidas en dos conjuntos de entrada \mathbb{I} , que a partir del periodo de muestreo tomado, suman 24000 muestras por fase. El conjunto de pesos \mathbb{W} de la red neuronal se actualizó usando un parámetro de velocidad de aprendizaje $Lr = 0.15$. La figura 5.5 presenta la evolución del error por muestra.

Para comprobar el aprendizaje de la red, se excita al sistema con un tren de escalones, donde sus amplitudes se distribuyen en un conjunto de entrada \mathbb{I} distinto al entrenamien-

³Número de entrenamientos.

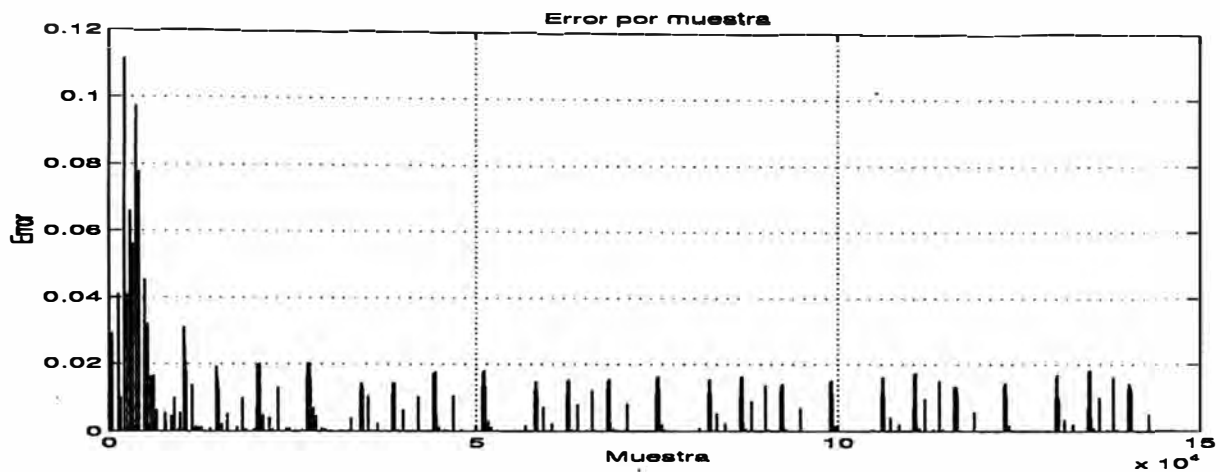


Figura 5.5: Error cuadrático por muestra durante el entrenamiento, para un parámetro de velocidad de aprendizaje $Lr = 0.15$.

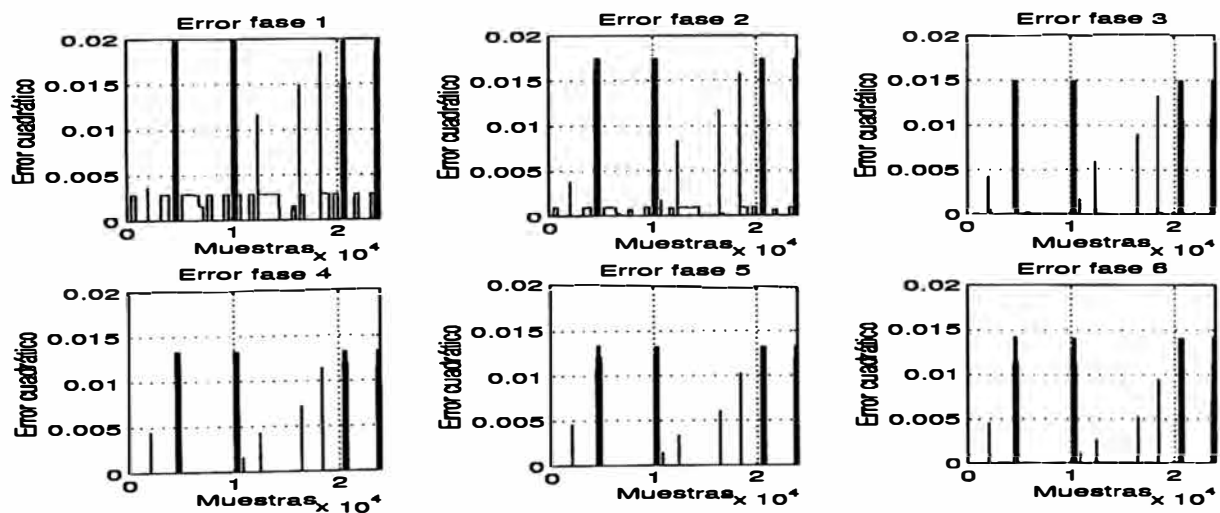


Figura 5.6: Evolución del error cuadrático en cada fase de entrenamiento, sin actualizar los pesos de la red

to. En la figura 5.6 se muestra la evolución de la red sin actualizar sus pesos en cada fase.

Si definimos el error cuadrático por fase como:

$$E_f = \frac{\sum_{k=1}^N e_k^2}{N} \quad (5.4)$$

donde N es el número de muestras por fase y e_k es el error por muestra sin actualizar los pesos de la red. Entonces se obtiene un error cuadrático promedio que se muestra en la figura 5.7.

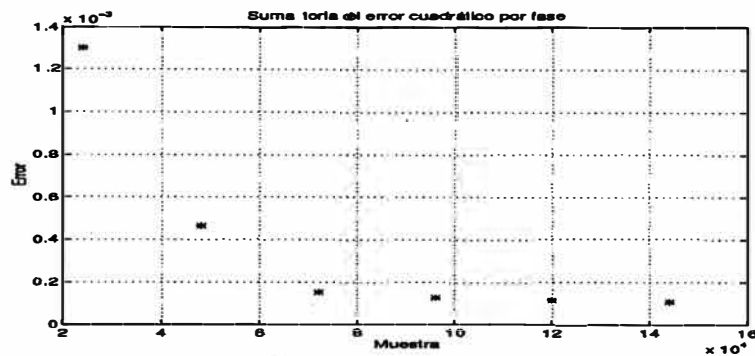


Figura 5.7: Error cuadrático promedio por fase de entrenamiento

En la figura 5.8 se muestra la identificación de la no linealidad $f(n\omega_k) = \frac{RC \operatorname{Sgn}(n\omega_k)}{KK_{act}}$ realizada por la red neuronal $\Pi_{1,20,10,1}^3$.

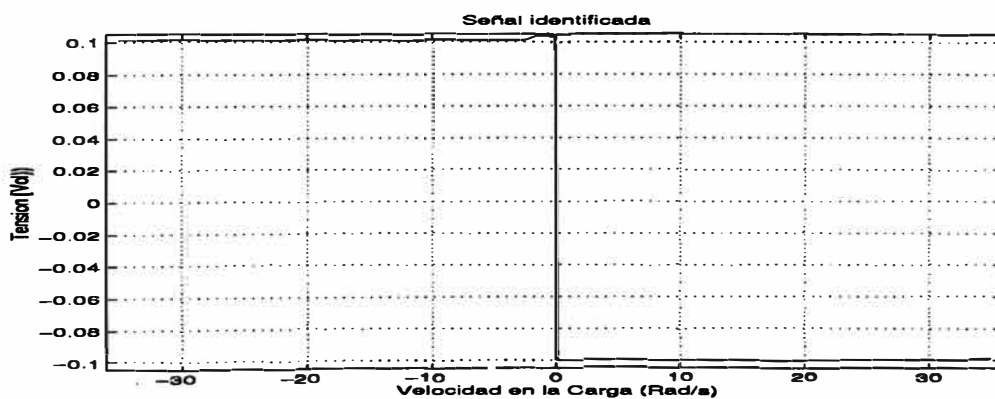


Figura 5.8: Identificación de la no linealidad realizada por la red Neuronal

5.1.2 Identificación de $f(n\omega_k)$ y los parámetros lineales del motor sin carga

Hasta este momento se ha mostrado que una red neuronal es capaz de identificar con un error aceptable a $f(n\omega_k) = \frac{RC \operatorname{Sgn}(n\omega_k)}{KK_{act}}$. Esto se realiza a partir de la ecuación

(5.5) que es una versión simplificada de (5.3):

$$\omega_{k+1} = G_1 \omega_k + G_2 \{V_{bk} + G_3 \text{Sgn}(n\omega_k)\} \quad (5.5)$$

Ahora asumiremos que no se conocen los parámetros lineales G_1 y G_2 del modelo matemático del motor ⁴. Entonces definiremos \widehat{G}_{1k} , \widehat{G}_{2k} y $\widehat{\omega}_k$ como los parámetros lineales y la velocidad angular respectivamente, estimados en el k -ésimo instante de tiempo, los cuales se encuentran relacionados por la siguiente ecuación:

$$\widehat{\omega}_{k+1} = \widehat{G}_{1k} \omega_k + \widehat{G}_{2k} \{V_{bk} + NNT(n\omega_k)\} \quad (5.6)$$

donde $NNT(n\omega_k)$ es la salida de la red neuronal para una entrada discreta $n\omega_k$. Si el error en el instante k se define como la diferencia entre la velocidad angular real y la estimada:

$$e_k = \widehat{\omega}_k - \omega_k. \quad (5.7)$$

Entonces el objetivo de identificación se reduce a minimizar el error e_k utilizando algún método de optimización. El método seleccionado es el *algoritmo del gradiente* por su simplicidad. Este algoritmo minimiza la función de costo $J = \frac{e_k^2}{2}$ en función de los parámetros estimados:

$$\Theta_{k+1} = \begin{bmatrix} \widehat{G}_{1k+1} \\ \widehat{G}_{2k+1} \end{bmatrix} \quad (5.8)$$

donde $\Theta_k = [\widehat{G}_{1k}, \widehat{G}_{2k}]^T$ es el vector de los parámetros identificados. La ley de identificación del vector Θ_k es de forma tal que la función de costo se minimiza en cada k -ésimo instante de tiempo, de modo que

$$\begin{bmatrix} \widehat{G}_{1k} \\ \widehat{G}_{2k} \end{bmatrix} = \begin{bmatrix} \widehat{G}_{1k-1} \\ \widehat{G}_{2k-1} \end{bmatrix} - \begin{bmatrix} g1 & 0 \\ 0 & g2 \end{bmatrix} \frac{\partial J}{\partial \Theta_{k-1}} \quad (5.9)$$

donde $\begin{bmatrix} g1 & 0 \\ 0 & g2 \end{bmatrix}$ es la matriz de velocidad de identificación y $g1, g2 > 0$ son los parámetros de velocidad de identificación que permiten variar la velocidad de convergencia. Si:

$$\begin{aligned} \frac{\partial J}{\partial \Theta_{k-1}} &= \frac{\partial J}{\partial e_k} \frac{\partial e_k}{\partial \Theta_{k-1}} \\ &= e_k \frac{\partial e_k}{\partial \Theta_{k-1}} \\ &= -e_k \begin{bmatrix} \omega_{k-1} \\ V_{bk-1} + NNT(n\omega_{k-1}) \end{bmatrix} \end{aligned} \quad (5.10)$$

⁴Este es el caso real.

se obtiene entonces la siguiente ley de identificación discreta:

$$\begin{bmatrix} \widehat{G}_{1k} \\ \widehat{G}_{2k} \end{bmatrix} = \begin{bmatrix} \widehat{G}_{1k-1} \\ \widehat{G}_{2k-1} \end{bmatrix} + \begin{bmatrix} g_1 & 0 \\ 0 & g_2 \end{bmatrix} \begin{bmatrix} \omega_{k-1} \\ V_{bk-1} + NNT(n\omega_{k-1}) \end{bmatrix} e_k. \quad (5.11)$$

Después de haber obtenido la ley de identificación discreta para la identificación de los parámetros lineales del modelo matemático del motor, el siguiente paso es la identificación conjunta de los parámetros lineales y elementos no lineales del motor. El esquema de identificación a implementar es el Serie-Paralelo, el cual se muestra en la figura 5.3. La señal de error e_k en este esquema sirve como dato de error para la red neuronal y la ley de identificación discreta. A continuación se muestra la simulación del entrenamiento de una red neuronal $\Pi_{1,20,10,1}^3$ que aproxima a la función no lineal $G_3 \text{Sgn}(n\omega_k)$ y a su vez realiza la identificación de los parámetros lineales del motor usando la ley de identificación (5.11).

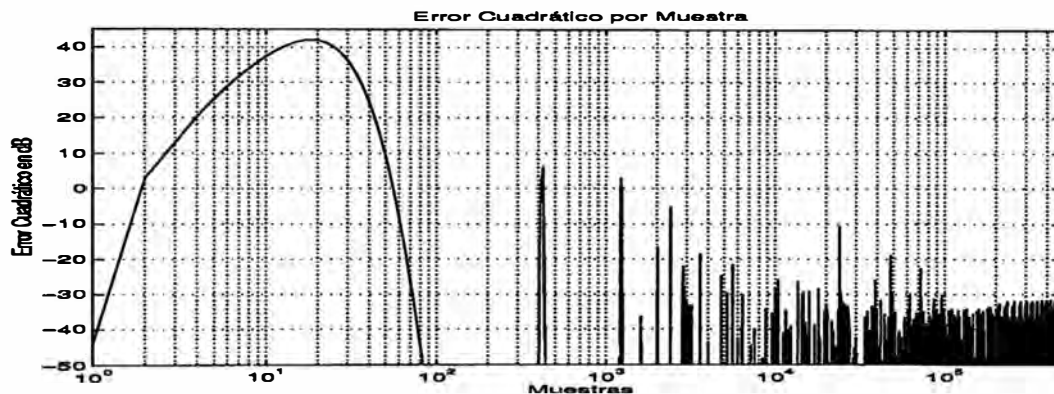


Figura 5.9: Error cuadrático por muestra durante el entrenamiento

El modelo del sistema a identificar se muestra en la figura 5.2 y los parámetros de este sistema en la tabla 5.1.1. La señal de excitación para aprendizaje y entrenamiento es idéntica al del 1^{er} experimento. El entrenamiento y comprobación del aprendizaje de la red se dividió también en fases de 24000 muestras cada una. A continuación se muestra en la figura 5.9 la evolución del error cuadrático por muestra (ECM), esta se muestra en dB para una mejor visualización de la misma. En esta figura se observa que el ECM llega hasta $130 \text{ rad}^2 \text{ s}^{-2}$. Este valor depende de los valores iniciales de los parámetros lineales (G_{1k} y G_{2k}). Existe también un ECM residual de $0.0125 \text{ rad}^2 \text{ s}^{-2}$.

En la figura 5.10 se presenta el error cuadrático por fase de entrenamiento (ECFE). De igual forma que la anterior experiencia, el ECFE se obtiene después de cada fase de entrenamiento sin actualizar los pesos de la red, con el fin de comprobar el aprendizaje de la misma. En esta figura se muestran los tres primeros y los tres últimos ECFE y se observa que la red aprendió mejor para entradas de tensión $V_b \in \{[-1.53, -0.13] \wedge [0.17, 1.37]\}$. Notar que los errores cuadráticos tienden a cero a diferencia de los errores

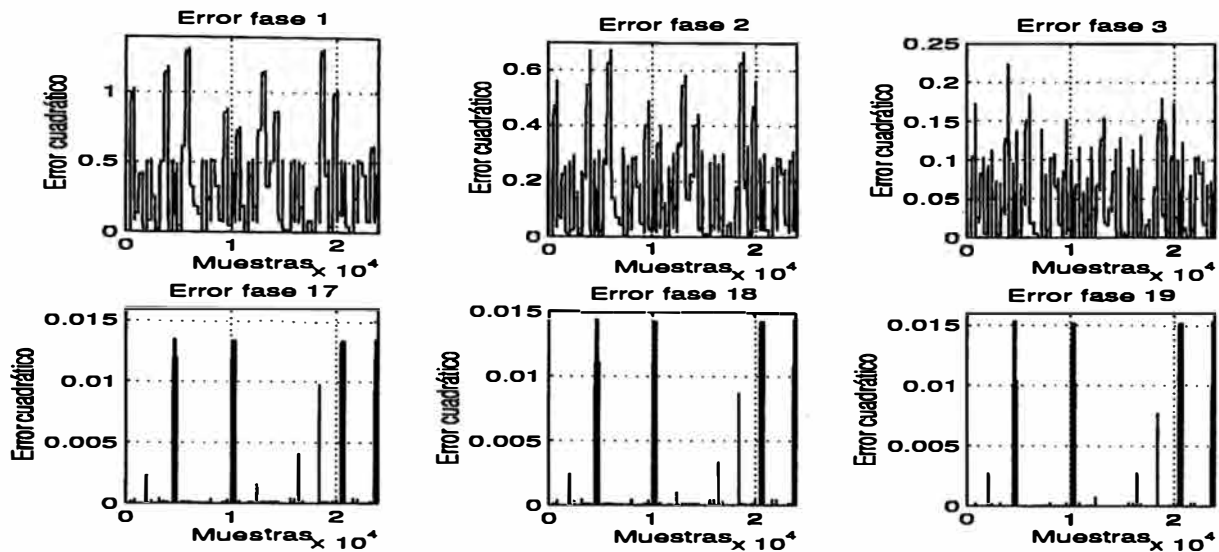


Figura 5.10: Evolución del error cuadrático en cada fase de entrenamiento, sin actualizar los pesos de la red.

producidos por tensiones de entrada $V_b \in [-0.13, 0.17]$ que son del orden de 0.015. Esta diferencia se debe a que existe una discontinuidad cuando $|\omega| \approx 0$ debido al modelo no lineal de la fricción de Coulomb.

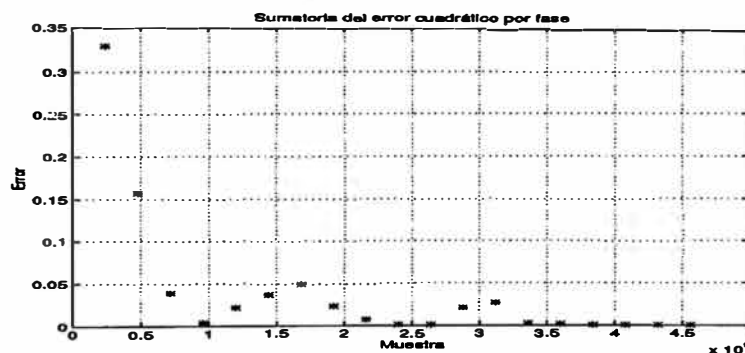


Figura 5.11: Error cuadrático promedio por fase de entrenamiento

En la figura 5.11 se muestra el error cuadrático promedio por fase de entrenamiento (ECPF) que se calcula a partir de la ecuación (5.4). Esta figura se presenta también la evolución del aprendizaje de la red. En este experimento se han procesado 456000 muestras en 19 fases; los pesos W de la red neuronal en las primeras 14 fases fueron actualizados con un $Lr = 0.15$. Con este Lr se observa que pueden ocurrir dos cosas: (1) Lr es grande lo cual hace que la red rebote, alejándose demasiado del valor mínimo verdadero; (2) Lr es grande haciendo que la red salte de un mínimo local a otro⁵. Todo esto ocurre en una superficie de espacio de pesos, observándose que el error cuadrático

⁵Este puede ser un mínimo local o global.

comienza a disminuir y a aumentar para luego disminuir otra vez. Después de la fase $N^{\text{ro}} 14$ se utilizó un $Lr = 0.04$ lo que hace que la red converja a un mínimo local o global. Lo ideal es que la red converja a un mínimo global, pero si la solución es admisible desde el punto de vista del error, no importa si se trata de un mínimo local o global, o si el entrenamiento se ha detenido en algún momento previo a alcanzar un mínimo real.

El número de muestras a procesar se pudo reducir cambiando Lr al final de la 4^{ta} fase por ejemplo y comprobar si el error cuadrático es aceptable. En la experiencia realizada, al final del entrenamiento se encontró un error cuadrático promedio de 8.4483×10^{-5} .

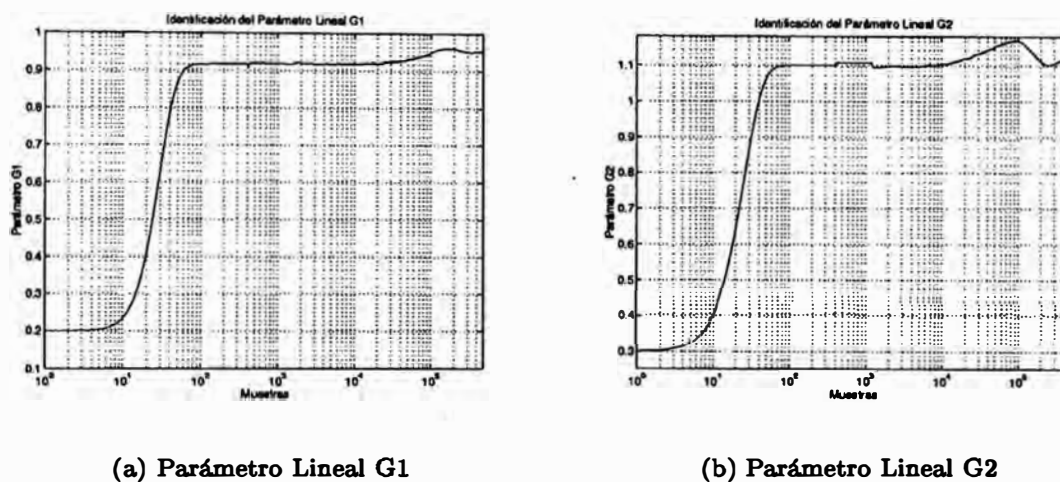


Figura 5.12: Identificación de los Parámetros Lineales G1 y G2

En la figura 5.12 se muestra la evolución de la identificación de los parámetros lineales $G1$ y $G2$ del modelo matemático del motor. Para la identificación se usó la ecuación (5.11); los parámetros de velocidad de identificación fueron $g1 = 0.001$ y $g2 = 0.009$. Estos parámetros juegan un papel importante en la identificación total del sistema, ya que valores de $g1, g2 > 0.01$ hacen que la identificación de la no linealidad del motor $C Sgn(n\omega_k)$ sea inestable y difícil de realizar por la red neuronal. Los valores de $g1, g2$ tienen que ser pequeños para tener una identificación estable. Se probaron valores iniciales de G_{1k}, G_{2k} menores que los nominales y mayores que cero al inicio de la identificación, con los cuales se logró identificar a los parámetros G_{1k}, G_{2k} en forma estable.

En la figura 5.13 se muestra la identificación de la no linealidad $f(n\omega_k) = \frac{RC Sgn(n\omega_k)}{KK_{act}}$ realizada por la red neuronal $\Pi_{1,20,10,1}^3$. Como se mencionó anteriormente, la identificación de $f(n\omega_k)$ para $|\omega| \approx 0$ no es buena ⁶ comparada con $|\omega| > 0$. En el gráfico 5.13 se pueden observar picos alrededor de $\omega = 0$; su efecto es tolerable si se toma como referencia el error cuadrático promedio del sistema identificado.

⁶En $\omega = 0$ existe una discontinuidad.

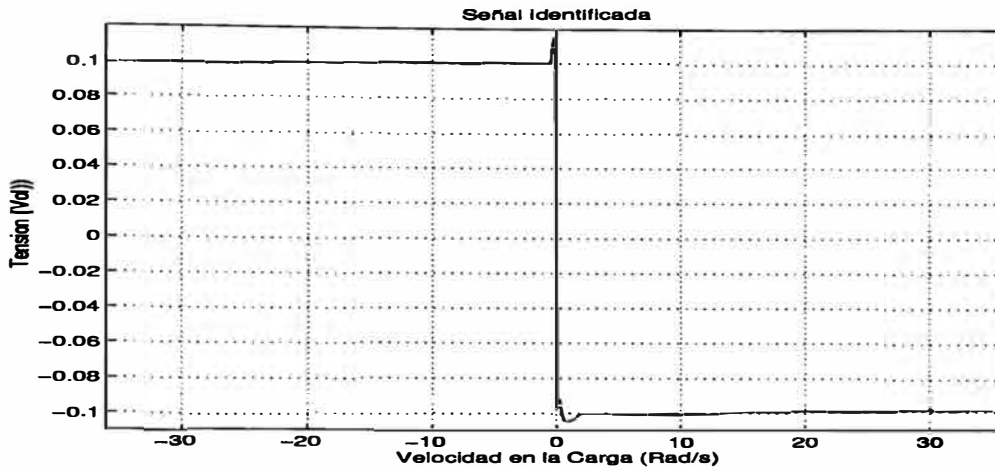


Figura 5.13: Identificación de la no linealidad realizada por la red Neuronal

5.1.3 Identificación los parámetros del motor con carga

Hasta el momento se ha realizado la identificación de los parámetros del motor sin carga. El siguiente paso a realizar es el análisis e identificación de los parámetros de un motor con una varilla en el eje. El modelo matemático de este sistema se muestra en la ecuación (3.7). De igual forma que en las secciones anteriores, se ha considerado la inductancia del motor despreciable ($L = 0$). A partir de (3.7) se obtiene la siguiente ecuación:

$$\left[J_{eff} + \frac{mL_o^2}{3n} \right] \dot{\omega} = \frac{V_b K K_{act}}{R} - \left[b_{eff} + \frac{nEK}{R} \right] \omega - \frac{gL_o m}{2n} \sin(q) - f^*(n\omega) \quad (5.12)$$

En la ecuación (5.12) se tiene dos no linealidades : $\sin(q)$ y $f^*(n\omega)$; reescribiendo (5.12) en una forma mas conveniente se obtiene:

$$\dot{\omega} = -\frac{b_{eff} + \frac{nEK}{R}}{J_{eff} + \frac{mL_o^2}{3n}} \omega + \frac{K K_{act}}{R \left[J_{eff} + \frac{mL_o^2}{3n} \right]} \left[V_b - \frac{R g L_o m}{2n K K_{act}} \sin(q) - \frac{R}{K K_{act}} f^*(n\omega) \right] \quad (5.13)$$

En la ecuación (5.13) se muestra que las no linealidades han sido trasladadas hacia la tensión de entrada; esto es útil porque se tiene acceso directamente a V_b . Por lo tanto, se puede operar sobre esta variable para eliminarlo mediante una realimentación de compensación de las no linealidades del motor. Discretizando la ecuación (5.13) se

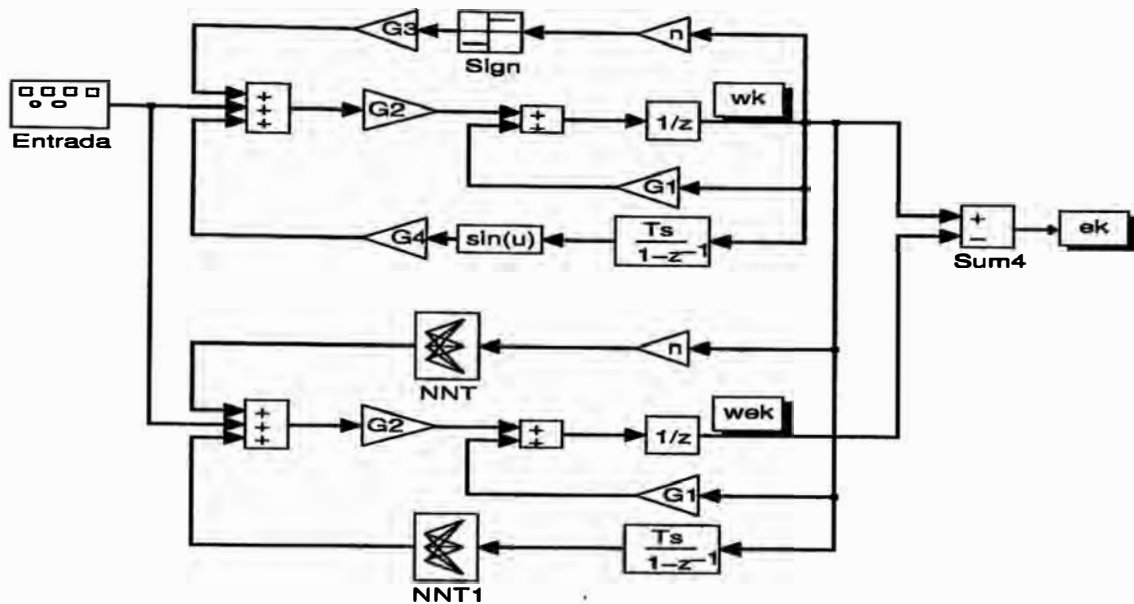


Figura 5.15: Esquema de identificación Serie-Paralelo implementado

ecuación (5.14) como:

$$\omega_{k+1} = G_1 \omega_k + G_2 \{V_{bk} + G_4 \sin(q_k) + NNT(n\omega_k)\} \quad (5.15)$$

entonces $\widehat{\omega}_{k+1}$ se puede escribir como sigue:

$$\widehat{\omega}_{k+1} = \widehat{G}_{1k} \omega_k + \widehat{G}_{2k} \{V_{bk} + NNT1(q_k) + NNT(n\omega_k)\} \quad (5.16)$$

Por tanto, si comparamos las ecuaciones (5.15), (5.16) con (5.5) y (5.6) respectivamente, se deduce fácilmente la siguiente ley de identificación:

$$\begin{bmatrix} \widehat{G}_{1k} \\ \widehat{G}_{2k} \end{bmatrix} = \begin{bmatrix} \widehat{G}_{1k-1} \\ \widehat{G}_{2k-1} \end{bmatrix} + \begin{bmatrix} g_1 & 0 \\ 0 & g_2 \end{bmatrix} \begin{bmatrix} \omega_{k-1} \\ V_{bk-1} + NNT1(q_{k-1}) + NNT(n\omega_{k-1}) \end{bmatrix} e_k \quad (5.17)$$

El sistema a simular se discretizó con un periodo de muestreo $T_s = 0.001 \text{ seg}$, los parámetros de este sistema se muestran en la tabla 5.1.3. Se probaron 3 posibles señales de entrada para entrenar a la red NNT1 de la figura 5.15. Primero, una señal escalón de periodo 0.7 seg, cuyas amplitudes se encuentran distribuidas en dos conjuntos de entrada II mencionados en la sección 5.1.1. Segundo, una señal sinusoidal, cuyas frecuencias, se encuentran distribuidas en un ancho de banda alrededor de la frecuencia de resonancia del sistema linealizado en $\theta = 0$. En este ancho de banda se tomo el criterio de -3dB para el cálculo de las frecuencias iniciales y finales, las amplitudes están distribuidas como en el primer conjunto de señales de entrada. Tercero, se combinaron las señales anteriores, para generar una nueva señal de entrada, esto se realiza intercalando el primer con el segundo conjunto de señales de entrada. Mejores resultados con respecto a la velocidad

Parámetro	Descripción	Valor
G_1	$1 - \frac{T_s(b_{eff} + \frac{nEK}{R})}{J_{eff} + \frac{mL_o^2}{3n}}$	0.99624
G_2	$\frac{KK_{act}}{R(J_{eff} + \frac{mL_o^2}{3n})}$	0.089013
G_4	$-\frac{RgL_o m}{2nKK_{act}}$	-0.19581
n	n	19.741
T_s	Período de Muestreo	0.001

Tabla 5.2: Tabla de Parámetros del Sistema Motor-Péndulo a Simular

de convergencia en la identificación de los parámetros lineales y no lineales se obtuvo con el primer conjunto de entrenamiento. El conjunto de pesos W de la red neuronal fueron actualizados con un $Lr = 0.05$ y los parámetros lineales G_1 , G_2 con una velocidad de identificación $nn1 = 0.0001$, $nn2 = 0.001$ respectivamente.

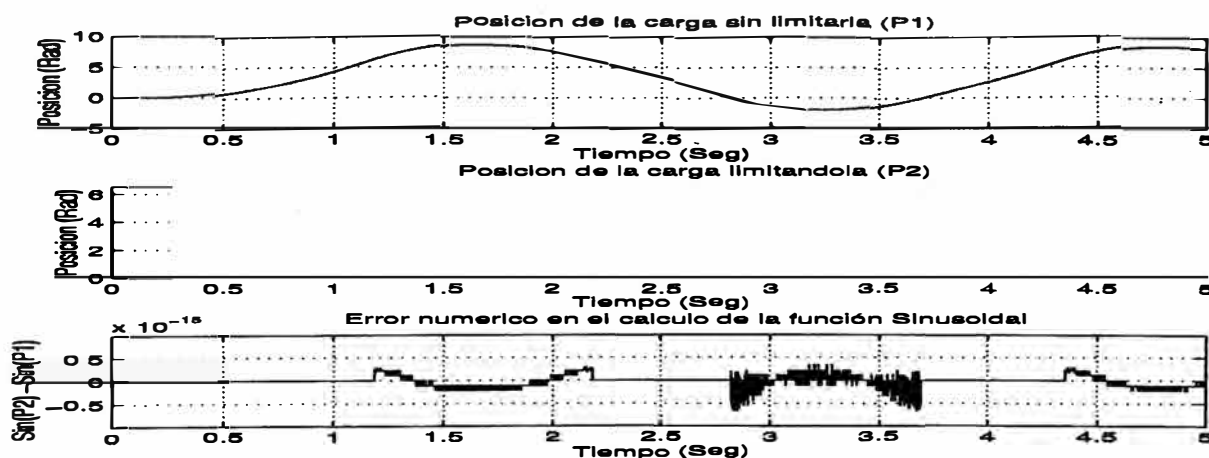


Figura 5.16: Evolución de la posición del eje del motor para una entrada sinusoidal, P1 es la posición sin utilizar limitación y P2 es la posición utilizandola.

Otro punto importante que hay que tomar en cuenta es la señal de entrada a NNT1. El método de actualización de pesos empleado ⁷ sirve solo para sistemas estáticos ⁸; estos sistemas se caracterizan por tener un conjunto de salida acotado para un único conjunto acotado de entrada. En nuestro caso, a partir del modelo matemático del sistema, se deduce fácilmente que la no linealidad a identificar ($\frac{RgL_o m}{2nKK_{act}} \sin(q_k)$) es

⁷Retropropagación.

⁸Sistemas sin dinámica.

una función estática ⁹ en el rango de $[0, 2\pi]$. En nuestras simulaciones, la posición se derivada a partir de la velocidad del eje de la carga. Por tanto se puede aprovechar la periodicidad de la función sinusoidal para acotar la posición del eje la carga en el rango de $[0, 2\pi]$, es decir limitarla. En la figura 5.16 se muestra como varia la posición de la carga para una entrada sinusoidal. en la primera figura se muestra la posición de la carga sin usar limitación, la cual llamaremos P1 mientras que en la segunda figura se muestra la posición usando limitación, la cual llamaremos P2, siendo esta señal la que se utiliza como entrada para la red neuronal NNT1. La limitación se ha realizado con la instrucción “ $u[1] - \text{floor}(u[1]/(2\pi)) * 2\pi$ ” de Matlab, la cual también a sido implementada en C donde $u[1]$ viene a ser la posición sin limitar en cada k – ésimo instante. Existe un error numérico ($\sin(P1) - \sin(P2)$) producto de la limitación, que es presentado en la tercera figura y que se puede despreciarse.

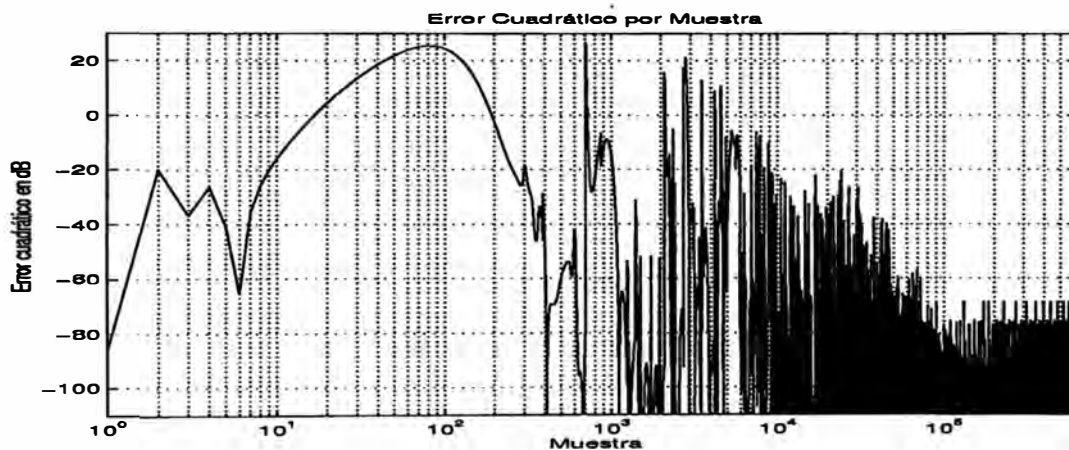


Figura 5.17: Error cuadrático por muestra durante el entrenamiento

El entrenamiento de NNT1 que consiste en una red $\Pi_{1,20,10,1}^3$, se realizó simultáneamente con identificación de los parámetros lineales de motor-péndulo G_1 y G_2 , a partir de la ley de identificación de la ecuación (5.17). La señal de error e_k de la figura 5.15 sirve como dato para los algoritmos de identificación utilizados. La red NNT de la sección 5.1.3, se usó para identificar la fricción de Coulomb. En la figura 5.17 se muestra el error cuadrático por muestra en dB para un mejor análisis, en este gráfico podemos observar que el ECM llega hasta $22.5 \text{ rad}^2 \text{ s}^{-2}$ que depende de los valores iniciales de G_1 y G_2 , y un error residual de $3.75 \times 10^{-4} \text{ Rad}^2 \text{ s}^{-2}$.

En la figura 5.18 se muestra el ECFE, este, como en las secciones anteriores nos muestra la evolución del aprendizaje de la red. Este error se determina sin actualizar los pesos de la red, la excitación del sistema consiste de señales escalones de duración 0.4 seg y con una distribución de amplitudes distinta a la señal de entrenamiento.

⁹Esta es una de las ventajas de tener información del modelo matemático del sistema.

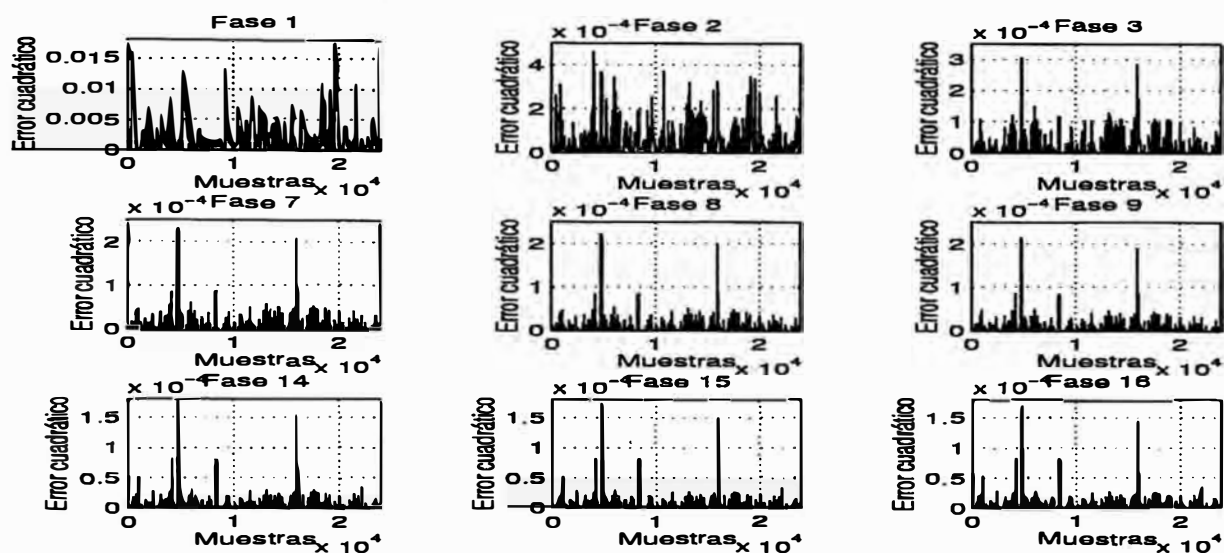


Figura 5.18: Error cuadrático en cada fase de entrenamiento sin actualizar los pesos de la red.

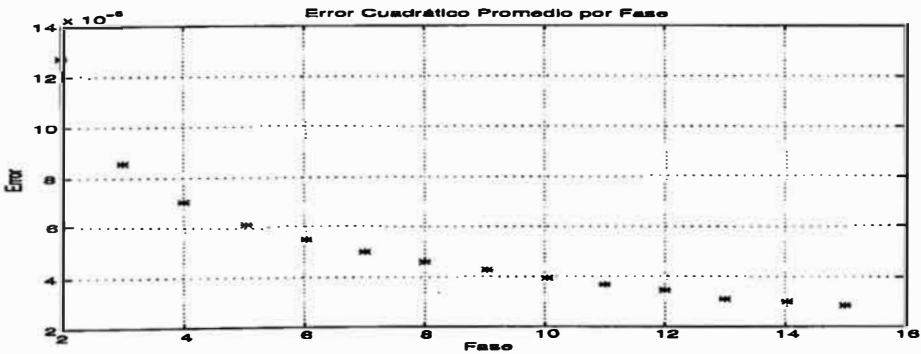


Figura 5.19: Error cuadrático promedio por fase de entrenamiento.

En la figura 5.19 se muestra el ECPF, producto de un experimento en el que se han procesado 641992 muestras en 16 fases. En este gráfico se presenta el ECPF a partir de la tercera fase para una mejor presentación de la evolución del error, los ECPF de las dos primeras fases son 0.002 y 3.9863×10^{-5} respectivamente.

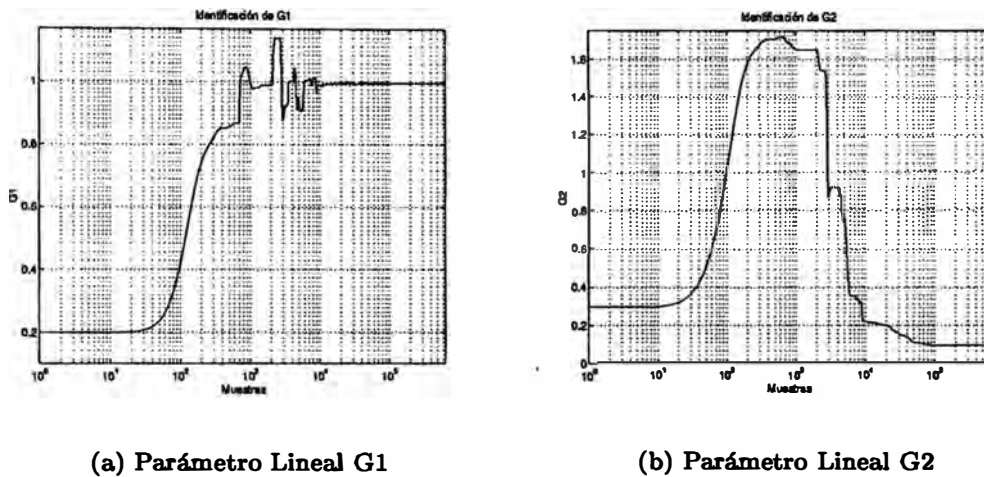


Figura 5.20: Identificación de los Parámetros Lineales G_1 y G_2

En las figuras 5.20(a) y 5.20(b) se muestra la evolución de la identificación de los parámetros lineales G_1 y G_2 . Los valores iniciales de estos parámetros al inicio del experimento fueron 0.2 y 0.3 respectivamente.

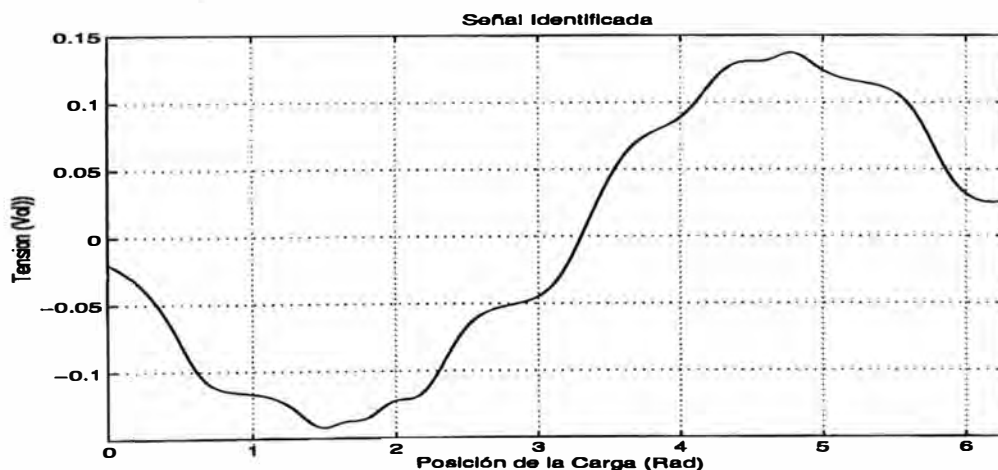


Figura 5.21: Señal identificada por la red neuronal NNT1.

En la figura 5.21 se muestra la señal identificada por la red neuronal NNT1, la señal de entrada a esta red es la posición del eje de la carga limitada entre $[0, 2\pi]$. Esta red a identificado a la no linealidad producida por la varilla en el eje de la carga. En este gráfico se observa el efecto de NNT (red que a identificado a la fricción de Coulomb) sobre la identificación de NNT1; en la siguiente sección se muestra el efecto conjunto

de estas dos redes. Aquí se observa que tan sensible es la identificación de NNT1 ante disturbios externos ¹⁰.

5.2 Compensación del sistema no lineal

En esta sección se muestra la compensación de los sistemas no lineales utilizando una realimentación negativa con las redes neuronales NNT y NNT1, las cuales fueron determinadas en secciones anteriores, de forma tal que cancelen los elementos que dan la característica no lineal del sistema a controlar.

5.2.1 Compensación del motor sin carga

La compensación del motor sin carga consiste en eliminar la fricción de Coulomb utilizando la red neuronal NNT. En la figura 5.22 se muestra el esquema de compensación de la fricción de Coulomb, en donde además se ha considerado el modelo continuo de motor ¹¹, ecuación (5.1), de forma tal que nos permita obtener la respuesta del modelo analógico del motor.

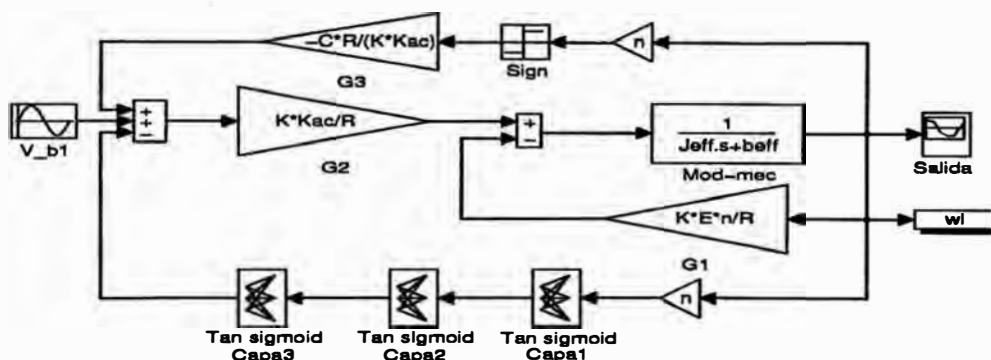


Figura 5.22: Esquema de compensación de la fricción de Coulomb.

La figura 5.23 muestra la respuesta del sistema utilizando la compensación de la fricción de Coulomb. El primer gráfico presenta la respuesta sin utilizar la compensación y el segundo cuando lo emplea. Aquí se observa la eliminación de la fricción de Coulomb por la acción de NNT.

5.2.2 Compensación del motor con carga

En esta sección se muestra la compensación de la fricción de Coulomb y el efecto de la componente sinusoidal del peso de la varilla por acción de la gravedad; aquí se utilizan dos redes neuronales NNT y NNT1.

¹⁰Cabe hacer la observación que NNT identificó a la fricción de Coulomb con un error, cuya presencia está afectando a NNT1. Experimentos de identificación sin considerar a NNT dieron como resultado una señal senoidal sin distorsiones.

¹¹En la figura 5.22 Mod-mec, es el modelamiento del sistema mecánico del motor.

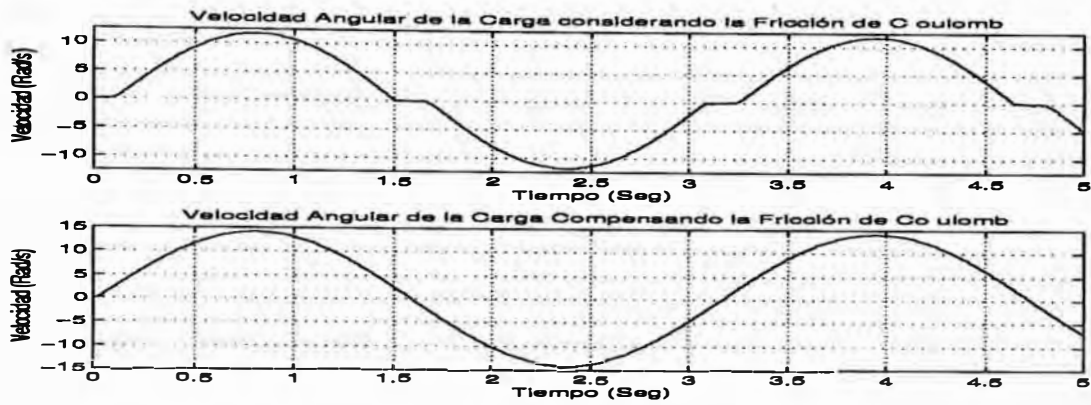


Figura 5.23: Respuesta del sistema utilizando compensación.

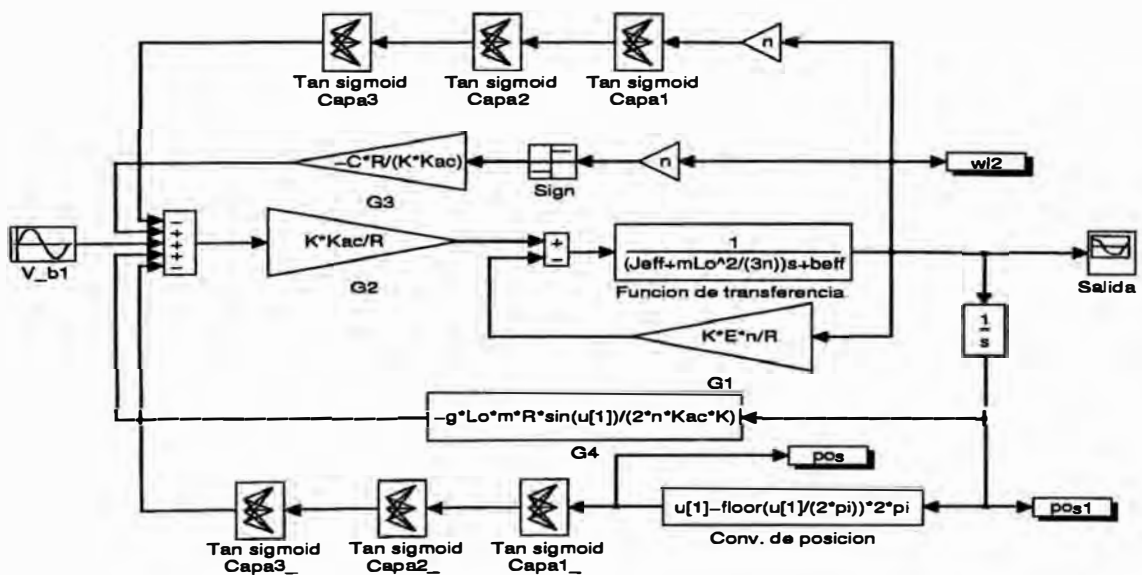


Figura 5.24: Esquema de compensación del sistema motor péndulo.

La figura 5.24 muestra el esquema de compensación utilizado, en donde al igual que en la sección anterior se ha considerado el modelo continuo del sistema (ecuación (5.12)).

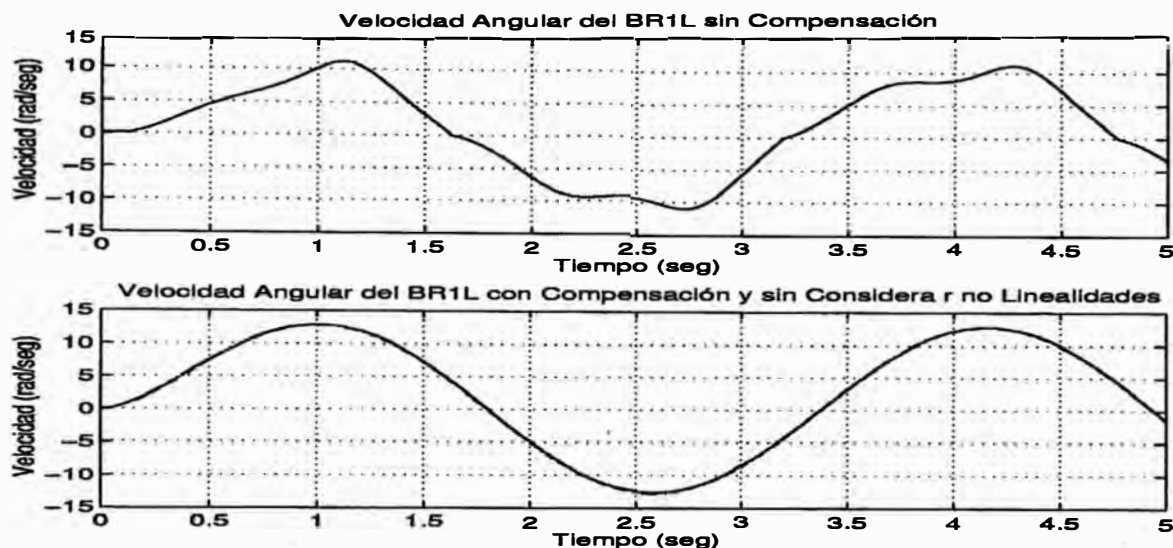


Figura 5.25: Respuesta del sistema utilizando compensación.

La figura 5.25 muestra la respuesta del sistema utilizando la compensación de las redes NNT y NNT1. El primer gráfico presenta la respuesta sin utilizar la compensación y el segundo superpone las respuestas con y sin compensación. Aquí se observa que la red NNT1 compensa el error residual de la red NNT.

5.3 Control no lineal del BR1L

En esta sección se analiza y diseña el control del sistema no lineal utilizando una realimentación de compensación con fines de linealización. La ecuación (5.18) muestra el sistema linealizado luego de la compensación realizado por las redes neuronales NNT y NNT1 las cuales compensan la fricción estática y de Coulomb y el efecto de la componente sinusoidal del peso de la varilla respectivamente. La ecuación (5.18) se deduce a partir de la ecuación (5.12).

$$\left(J_{eff} + \frac{mL_o^2}{3n}\right)\dot{\omega} = \frac{V_b K K_{act}}{R} - \left(b_{eff} + \frac{nEK}{R}\right)\omega \quad (5.18)$$

5.3.1 Análisis del sistema linealizado

Para realizar este análisis se puede considerar despreciable el error residual debido a la compensación. La ecuación (5.18) representada en variables de estado se muestra

en la ecuación (5.19):

$$\mathbf{x} = \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} q \\ \omega \end{bmatrix}$$

$$\dot{\mathbf{x}} = \begin{bmatrix} 0 & 1 \\ 0 & -\frac{(b_{eff} + \frac{nEK}{R})}{(J_{eff} + \frac{mL_o^2}{3n})} \end{bmatrix} \mathbf{x} + \begin{bmatrix} 0 \\ \frac{KK_{act}}{R(J_{eff} + \frac{mL_o^2}{3n})} \end{bmatrix} V_b = \mathbf{Ax} + \mathbf{BV}_b \quad (5.19)$$

$$\mathbf{y} = \begin{bmatrix} 1 & 0 \end{bmatrix} \mathbf{x} = \mathbf{Cx}$$

Los criterios de controlabilidad, observabilidad y estabilidad son definidos en [1] los cuales son aplicados a continuación:

Controlabilidad.-

La controlabilidad esta establecida si y sólo si la matriz $\mathbf{C} = [\mathbf{B} | \mathbf{AB}]$ es de rango 2. Entonces de (5.19) tenemos que:

$$|\mathbf{C}| = \left| \begin{bmatrix} 0 & \frac{KK_{act}}{R(J_{eff} + \frac{mL_o^2}{3n})} \\ \frac{KK_{act}}{R(J_{eff} + \frac{mL_o^2}{3n})} & -\frac{(b_{eff} + \frac{nEK}{R})KK_{act}}{R(J_{eff} + \frac{mL_o^2}{3n})^2} \end{bmatrix} \right| = |[\mathbf{B} | \mathbf{AB}]| \neq 0 \quad (5.20)$$

Por lo tanto el rango de la matriz de controlabilidad \mathbf{C} es igual a 2 y el sistema (5.19) es completamente controlable.

Observabilidad.-

La observabilidad esta establecida si y sólo si la matriz $\mathbf{O} = \begin{bmatrix} \mathbf{C} \\ \mathbf{CA} \end{bmatrix}$ es de rango 2. Entonces de (5.19) tenemos que :

$$|\mathbf{O}| = \left| \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \right| = \left| \begin{bmatrix} \mathbf{C} \\ \mathbf{CA} \end{bmatrix} \right| \neq 0 \quad (5.21)$$

Por lo tanto el rango de la matriz de observabilidad \mathbf{O} es igual a 2 y el sistema (5.19) es completamente observable.

Estabilidad.-

La estabilidad del sistema lineal (5.19) se encuentra garantizado si los eigenvalores λ de la matriz \mathbf{A} tienen una parte real negativa; entonces si:

$$|\lambda \mathbf{I} - \mathbf{A}| = \left| \begin{bmatrix} \lambda & -1 \\ 0 & \lambda + \frac{(b_{eff} + \frac{nEK}{R})}{(J_{eff} + \frac{mL_o^2}{3n})} \end{bmatrix} \right| = 0; \quad (5.22)$$

resultando:

$$\begin{aligned} \lambda_1 &= 0 \\ \lambda_2 &= -\frac{(b_{eff} + \frac{nEK}{R})}{(J_{eff} + \frac{mL_o^2}{3n})} \end{aligned} \quad (5.23)$$

Por consiguiente el sistema es inestable; entonces se debe diseñar un sistema de control que traslade los polos del sistema a un lugar conveniente, con la finalidad de obtener un sistema estable.

5.3.2 Diseño del servo controlador

Después de linealizar el BR1L, el control se puede realizar utilizando algún método clásico. En esta sección se diseña el servo controlador del BR1L linealizado. El método de control elegido es el PID, por su simplicidad y fácil diseño (sintonización).

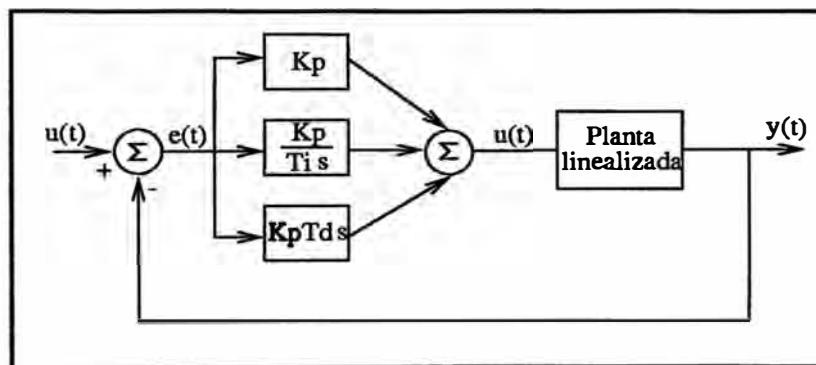


Figura 5.26: Esquema básico de un sistema de control utilizando un controlador PID

Un contrador PID clásico se muestra en la figura 5.26. La ecuación que describe tal controlador es la siguiente:

$$u(t) = K_p \left[e(t) + \frac{1}{T_i} \int^t e(\tau) d(\tau) + T_d \frac{de(t)}{dt} \right] \quad (5.24)$$

donde el error $e(t)$ es la diferencia entre la señal de comando $r(t)$ (trayectoria a seguir) y el proceso de salida $y(t)$ (la variable medida). En el dominio de la frecuencia, utilizando Laplace, la función de transferencia del controlador se puede escribir como:

$$H_c(s) = \frac{U(s)}{E(s)} = K_p \left[1 + \frac{1}{sT_i} + T_d s \right] \quad (5.25)$$

La función de transferencia $H_p(s)$ del sistema linealizado (planta), ecuación (5.18),

se muestra en la siguiente ecuación:

$$H_p(s) = \frac{Y(s)}{U(s)} = \frac{\frac{K_{act}K}{R(J_{eff} + \frac{mL_0^2}{3n})}}{s^2 + \frac{b_{eff} + \frac{KE\eta}{R}}{J_{eff} + \frac{mL_0^2}{3n}}} = \frac{a}{s^2 + bs} \quad (5.26)$$

La ecuación de transferencia de lazo cerrado $H_{LC}(s)$ se obtiene a partir de las ecuaciones (5.25) y (5.26), como sigue:

$$H_{LC}(s) = \frac{Y(s)}{R(s)} = \frac{H_p H_c}{1 + H_p H_c} = \frac{\frac{aK_p}{T_i}(T_i T_d s^2 + T_i s + 1)}{s^3 + (b + aK_p T_d)s^2 + aK_p s + \frac{aK_p}{T_i}} \quad (5.27)$$

La ecuación característica del sistema de lazo cerrado se obtiene a partir de la ecuación (5.28):

$$s^3 + (b + aK_p T_d)s^2 + aK_p s + \frac{aK_p}{T_i} = 0 \quad (5.28)$$

Si los polos del sistema de lazo cerrado son $[p_1, p_2, p_3]$, la ecuación característica del sistema BR1L viene dada por:

$$s^3 - (p_1 + p_2 + p_3)s^2 + (p_1 p_2 + p_2 p_3 + p_1 p_3)s - p_1 p_2 p_3 = 0 ; \quad (5.29)$$

de donde, comparando las ecuaciones (5.29) y (5.28), se obtiene el siguiente sistemas de ecuaciones para el cálculo de K_p , T_i , T_d :

$$\begin{aligned} b + aK_p T_d &= -(p_1 + p_2 + p_3) \\ aK_p &= p_1 p_2 + p_2 p_3 + p_1 p_3 \\ \frac{aK_p}{T_i} &= -p_1 p_2 p_3 \end{aligned} \quad (5.30)$$

Si se eligen las siguientes raíces, que caracterizan la respuesta del sistema de lazo cerrado:

$$\begin{aligned} p_1 &= -20.0 + 13.643i \\ p_2 &= -20.0 - 13.643i \\ p_3 &= -50. \end{aligned} \quad (5.31)$$

se obtienen los siguientes parámetros del controlador:

$$\begin{aligned} K_p &= 29.053 \\ T_i &= 0.0982 \\ T_d &= 0.010 \end{aligned} \quad (5.32)$$

La ecuación (5.24) es una versión simplificada del controlador PID. Originalmente los controladores PID fueron implementados usando una tecnología analógica. Virtualmente hoy todos los controladores PID son implementados digitalmente, incorporándose en ellas las características más relevantes de los diseños analógicos. A continuación se muestra una aproximación discreta de un controlador PID analógico con algunas modificaciones.

Modificación de la respuesta lineal

Al implementar la parte derivativa pura del controlador, se produce una gran amplificación de la medición del ruido. Por tanto, se tiene que limitar la ganancia derivativa del controlador; esto se realiza aproximando la función de transferencia sT_d , como se muestra a continuación:

$$sT_d \approx \frac{sT_d}{1 + \frac{sT_d}{N}} \quad (5.33)$$

La función de transferencia de la derecha aproxima bien a la derivada a bajas frecuencias, pero esta ganancia es limitada por N a altas frecuencias. N típicamente se encuentra en el rango de 3 a 20.

En trabajos realizados con controladores analógicos se encontró ventajoso no dejar que la parte derivativa actúe sobre la señal de comando y se dejó que solo una fracción b de la señal de comando actúe sobre la parte proporcional. De esta manera el algoritmo del controlador PID llega a ser:

$$U(s) = K_p \left[b R(s) - Y(s) + \frac{1}{sT_i} (R_c(s) - Y(s)) - \frac{sT_d}{1 + \frac{sT_d}{N}} Y(s) \right] \quad (5.34)$$

Se observa que se proveen distintos caminos entre la señal de salida del proceso y la señal de comando, lo cual puede interpretarse como una manera de posicionar los ceros de lazo cerrado.

Aproximación discreta

El controlador PID de la ecuación (5.34), se puede aproximar discretamente como sigue. La parte proporcional.

$$P(t) = K_p (br_c(t) - y(t)), \quad (5.35)$$

no requiere aproximación. Para la parte integral

$$I(t) = \frac{K}{T_i} \int^t e(\tau) d\tau, \quad (5.36)$$

se utiliza una aproximación rectangular de la forma:

$$I_{k+1} = I_k + \frac{K_p T_s}{T_i} e_k \quad (5.37)$$

y la parte derivativa tiene la forma:

$$\frac{T_d}{N} \frac{dD}{dt} + D = -K_p T_d \frac{dy}{dt} \quad (5.38)$$

cuando se toma una diferenciación atrazada, la cual dá:

$$D_k = \frac{T_d}{T_d + NT_s} D_{k-1} - \frac{K_p T_d N}{T_d + NT_s} (y_k - y_{k-1}). \quad (5.39)$$

Esta aproximación tiene la ventaja de ser siempre estable y los polos de sistema discretizado tienden a cero cuando T_d tiende a cero. Por tanto, la señal de control viene dada por:

$$u_k = P_k + I_k + D_k \quad (5.40)$$

Utilizando la transformada Z en las ecuaciones (5.40), (5.35), (5.37) y (5.39), la ecuación (5.34) puede entonces escribirse como:

$$U(z) = K_p \left[bR_c(z) - Y(z) + \frac{z^{-1}T_s}{(1-z^{-1})T_i} E(z) - \frac{NT_d}{T_d + NT_s} \frac{(1-z^{-1})}{1 - \frac{T_d}{T_d + NT_s} z^{-1}} Y(z) \right] \quad (5.41)$$

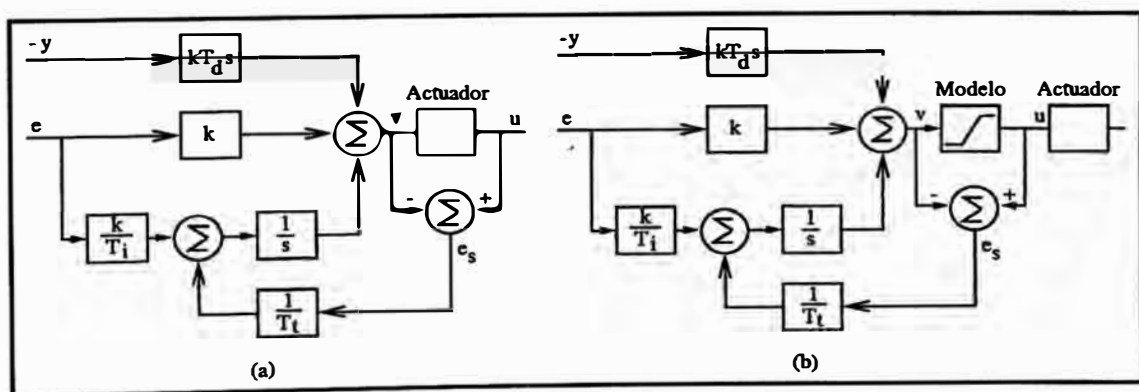


Figura 5.27: Sistema de control con Antiwindup. Un sistema donde la salida del actuador es medible se muestra en (a) y un sistema donde la salida del actuador es estimada por un modelo matemático se muestra en (b)

Integrador Windup

Si la acción integral del controlador se combina con un actuador que tiene características de saturación entonces se pueden dar efectos indeseables. Por ejemplo, si el error del control es tan grande de tal forma que el integrador satura al actuador, el camino de realimentación se rompería dado que el actuador permanecería saturado cuando el proceso de salida cambia. El integrador es un sistema inestable, porque puede llegar a integrar hasta un valor muy grande. Cuando el error es finalmente reducido, la integral puede ser tan grande que requiere tomar un considerable tiempo, hasta que la integral llega a su valor normal otra vez. Este efecto es llamado *integrador windup*. Hay varias maneras de anular la integración windup. Una posibilidad es parar la actualización de la integral cuando el actuador está saturado. Otro método es mostrado en el diagrama de bloques de la figura 5.27. En este sistema se adiciona un camino extra de realimentación para medir la salida del actuador y formar la señal de error (e_s) que es la diferencia entre el actuador de salida (u_c) y el controlador de salida (v). Este error se realimenta al integrador a través de la ganancia $1/T_t$. La señal de error e_s es cero cuando el actuador no está saturado. Cuando el actuador se satura, el camino extra de realimentación trata de hacer que la señal de error e sea igual a cero. Esto significa que el integrador en esta situación se pone a un valor cuando la salida del controlador está al límite de la saturación. El integrador se puede reconfigurar a un valor apropiado usando la constante de tiempo T_t , la cual es llamada *constante de tiempo de seguimiento*. La ventaja con este esquema antiwindup es que puede ser aplicado a cualquier actuador; por ejemplo, no solo para un actuador con saturación, sino también para uno con características arbitrarias tales como zona muerta o histéresis, siempre que la salida del actuador sea medible. Si la salida del actuador no es medible como en nuestro caso, se puede utilizar un modelo del actuador para generar una señal equivalente al actuador, tal como se observa en la figura 5.27.

Sintonización

En un controlador PID los parámetros a ser elegidos son K_p , T_i , T_d , T_t , b , N , u_{low} , u_{high} . Los parámetros primarios son K_p , T_i y T_d . El parámetro N , puede fijarse en $N = 10$. La constante de tiempo de seguimiento T_t está frecuentemente relacionado con el tiempo de integración T_i . En algunos casos, esta constante de tiempo puede ser elegida entre 0.1 a 0.5 veces el valor de T_i . Los parámetros u_{low} y u_{high} son elegidos dentro de los límites de saturación del actuador. Los parámetros primarios pueden ser calculados a partir del método descrito en la sección 5.3.2. En la tabla 5.3.2 se muestran los valores tomados para el servocontrolador. La ecuación (5.34) y el Integrador Windup se implementaron en Simulink, tal como se muestra en la figura 5.28. El subsistema

Parámetro	Valor	Parámetro	Valor
K_p	29.053	b	0.9
T_i	0.098	N	10
T_d	0.010	u_{low}	-1.53
T_i	0.082420	u_{high}	1.37

Tabla 5.3: Tabla de Parámetros del Controlador

Motor-Pendulo contiene el modelo matemático del motor, driver y su realimentación de compensación. El modelo del driver esta dado por la ecuación 5.42. Si a la tensión de entrada del modelo (V_b) mostrado en la figura 5.24 le adicionamos un saturador (modelo del driver) se obtendría el subsistema Motor-Pendulo.

$$f(V_b) = \begin{cases} u_{max} & \text{Si } V_b > u_{max} \\ V_b & \text{Si } u_{min} \leq V_b \leq u_{max} \\ u_{min} & \text{Si } V_b < u_{min} \end{cases} \quad (5.42)$$

La señal **Sat** se obtiene al realizar la diferencia: $Sat = V_b - f(V_b)$ El servo sistema

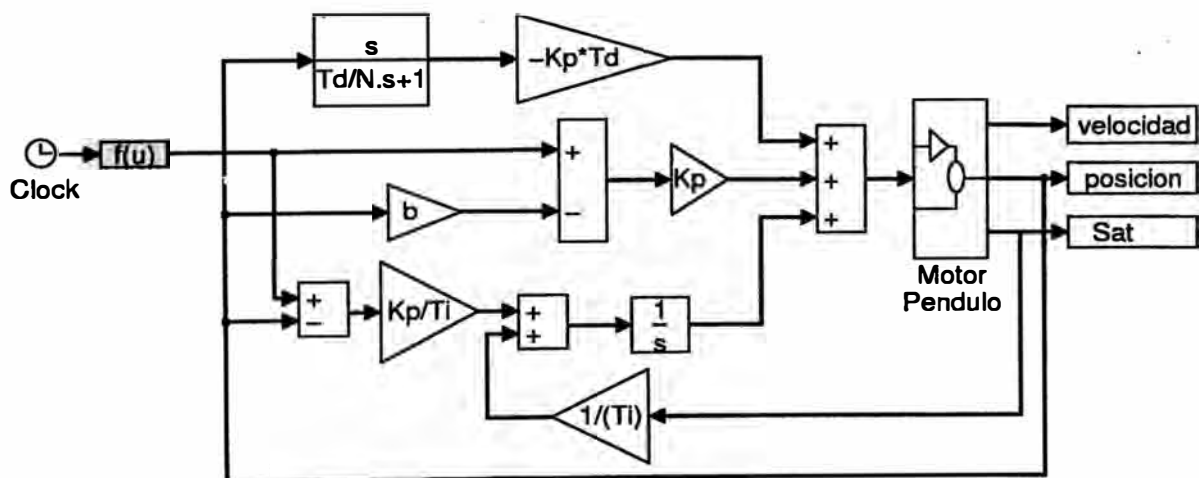


Figura 5.28: Esquema de simulación del sistema de control usando redes neuronales.

diseñado se probó para seguir la trayectoria: $r(t) = -\sin(\pi t) + \sin(\frac{\pi t}{4})$. Los parámetros K_p , T_i , T_d se sintonizaron a partir de la ecuación (5.32). La posición del BR1L con respecto al tiempo se muestra en la figura 5.29.

La señal de control $u(t)$ transmitida a la planta se muestra en la figura 5.30.

5.4 Control PID del BR1L

Ahora deseamos controlar el sistema no lineal BR1L empleando un controlador PID y sin utilizar redes neuronales. Un requisito importante para el diseño de este controlador

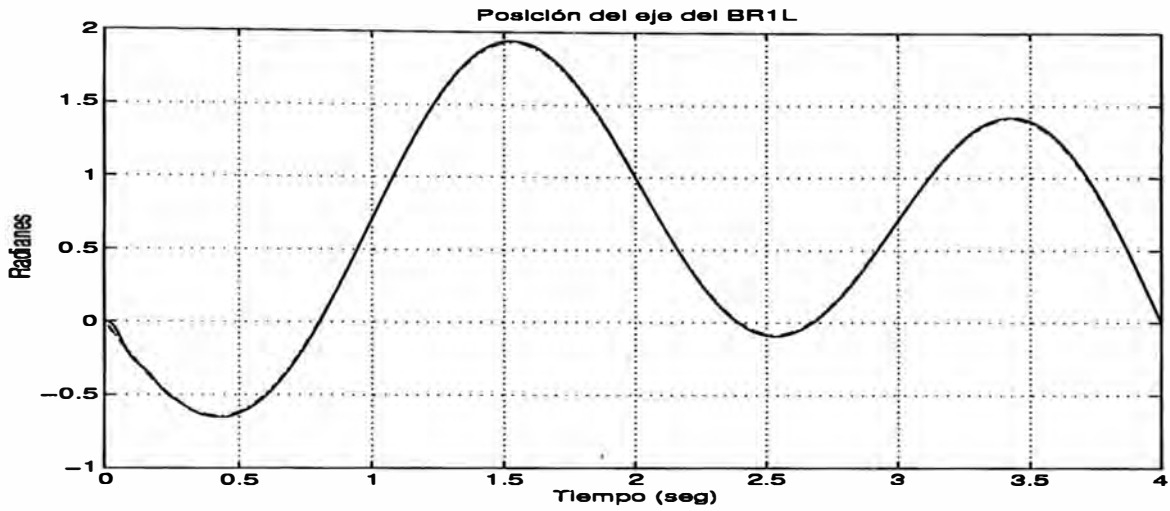


Figura 5.29: Señal de salida del sistema de control en donde se muestra el ángulo del eje del BR1L linealizado. La línea continua es la trayectoria seguida por el BR1L linealizado y la punteada es la trayectoria de referencia.

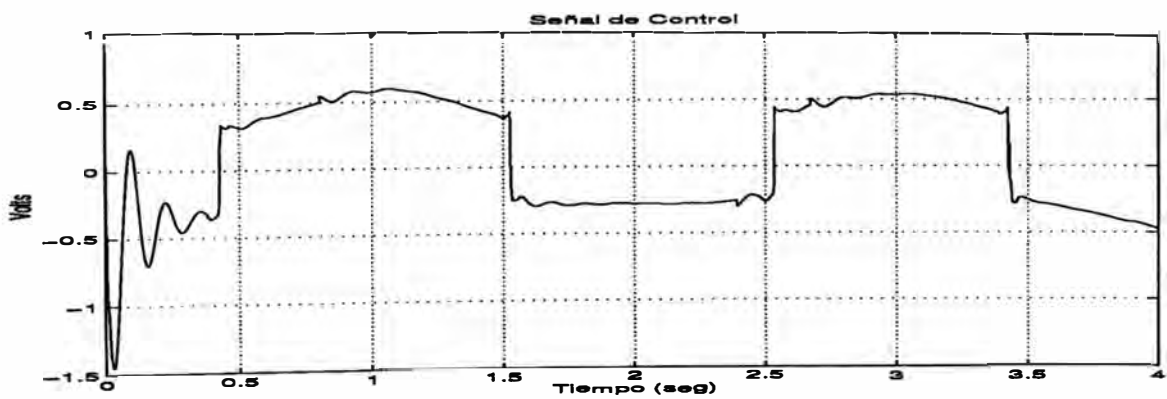


Figura 5.30: Señal de control proporcionada a la planta en presencia de las redes neuronales.

es trabajar con un sistema lineal. El sistema no lineal BR1L se puede linealizar alrededor de $\theta = 0$ y se puede considerar despreciable el efecto de la fricción estática y de Coulomb. Este sistema linealizado, utilizando realimentación de estado se muestra en la ecuación (5.43).

$$\begin{aligned} \mathbf{x} &= \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} q \\ \omega \end{bmatrix} \\ \dot{\mathbf{x}} &= \begin{bmatrix} 0 & 1 \\ -\frac{gL_0m}{(J_{eff} + \frac{mL_0^2}{3n})} & -\frac{(b_{eff} + \frac{nEK}{R})}{(J_{eff} + \frac{mL_0^2}{3n})} \end{bmatrix} \mathbf{x} + \begin{bmatrix} 0 \\ \frac{KK_{act}}{R(J_{eff} + \frac{mL_0^2}{3n})} \end{bmatrix} V_b = \mathbf{Ax} + \mathbf{BV}_b \\ \mathbf{y} &= \begin{bmatrix} 1 & 0 \end{bmatrix} \mathbf{x} = \mathbf{Cx} \end{aligned} \quad (5.43)$$

Es fácil demostrar que el sistema linealizado de la ecuación (5.43) es completamente controlable, completamente observable y asintóticamente estable. Esto se puede verificar fácilmente empleando los criterios descritos en la sección 5.3.1.

El diseño del controlador es similar al de la sección 5.3.2. La función de transferencia de la planta (H_p) viene dada por:

$$H_p(s) = \frac{Y(s)}{U(s)} = \frac{\frac{K_{act}K}{R(J_{eff} + \frac{mL_0^2}{3n})}}{s^2 + \frac{b_{eff} + \frac{KEN}{R}}{J_{eff} + \frac{mL_0^2}{3n}}s + \frac{gL_0m}{2n(J_{eff} + \frac{mL_0^2}{3n})}} = \frac{a}{s^2 + bs + c} \quad (5.44)$$

La función de transferencia de lazo cerrado $H_{LC}(s)$ es de la forma:

$$H_{LC}(s) = \frac{Y(s)}{R(s)} = \frac{H_p H_c}{1 + H_p H_c} = \frac{\frac{aK_p}{T_i}(T_i T_d s^2 + T_i s + 1)}{s^3 + (b + aK_p T_d)s^2 + (aK_p + c)s + \frac{aK_p}{T_i}} \quad (5.45)$$

y la ecuación característica del sistema se obtiene haciendo el denominador de (5.45) igual a cero. Es decir $s^3 + (b + aK_p T_d)s^2 + (aK_p + c)s + \frac{aK_p}{T_i} = 0$. Si los polos de lazo cerrado son p_1, p_2 y p_3 , la ecuación característica del sistema BR1L viene dado por la ecuación (5.29). Entonces los parámetros K_p, K_i, K_d del controlador se obtienen a partir del siguiente conjunto de ecuaciones:

$$\begin{aligned} b + aK_p T_d &= -(p_1 + p_2 + p_3) \\ aK_p + c &= p_1 p_2 + p_2 p_3 + p_1 p_3 \\ \frac{aK_p}{T_i} &= -p_1 p_2 p_3 \end{aligned} \quad (5.46)$$

Si se eligen las siguientes raíces, que caracterizan la respuesta del sistema de lazo cerrado:

$$\begin{aligned} p1 &= -6.66 + 17.395i \\ p2 &= -6.66 - 17.395i \\ p3 &= -10. \end{aligned} \quad (5.47)$$

se obtienen los siguientes parámetros del controlador:

$$\begin{aligned} K_p &= 5.204 \\ T_i &= 0.1334 \\ T_d &= 0.0423. \end{aligned} \quad (5.48)$$

El servosistema diseñado para BR1L linealizado alrededor de $\theta = 0$ se probó para seguir la trayectoria: $r(t) = -\sin(\pi t) + \sin(\frac{\pi t}{4})$, donde $0 \leq t \leq 4$. La posición del BR1L con respecto al tiempo se muestra en la figura 5.32. Se están considerando despreciables los efectos de la fricción estática de Coulomb. El esquema de este sistema de control realizado en simulink se muestra en la figura 5.31. El subsistema Motor-Pendulo de la figura 5.31 es igual al de la figura 5.24 pero sin las realimentaciones de compensación.

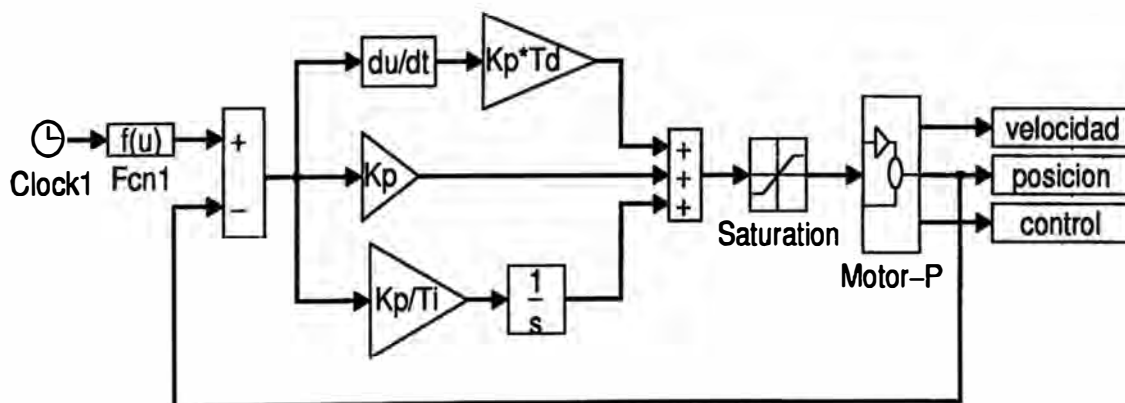


Figura 5.31: Esquema de simulación del sistema de control lineal, utilizando un controlador PID.

La señal de control $u(t)$ transmitida a la planta se muestra en la figura 5.33.

La compensación utilizando redes neuronales disminuye el error de seguimiento de la trayectoria con respecto al del PID simple. En el gráfico 5.34 se muestran estos errores de seguimiento.

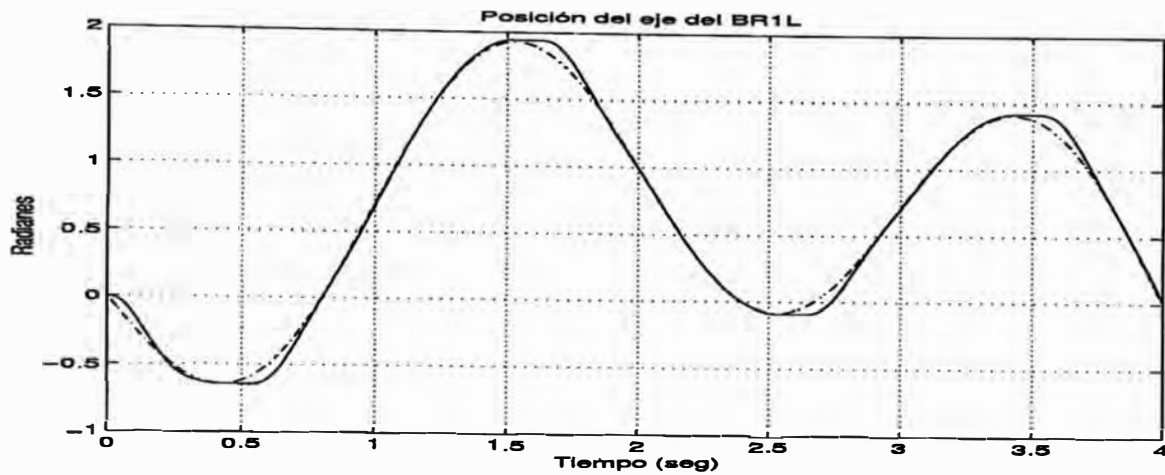


Figura 5.32: Señal de salida del sistema de control donde se muestra el ángulo del eje del BR1L. La línea continua es la trayectoria seguida por el BR1L y la punteada es la trayectoria de referencia.

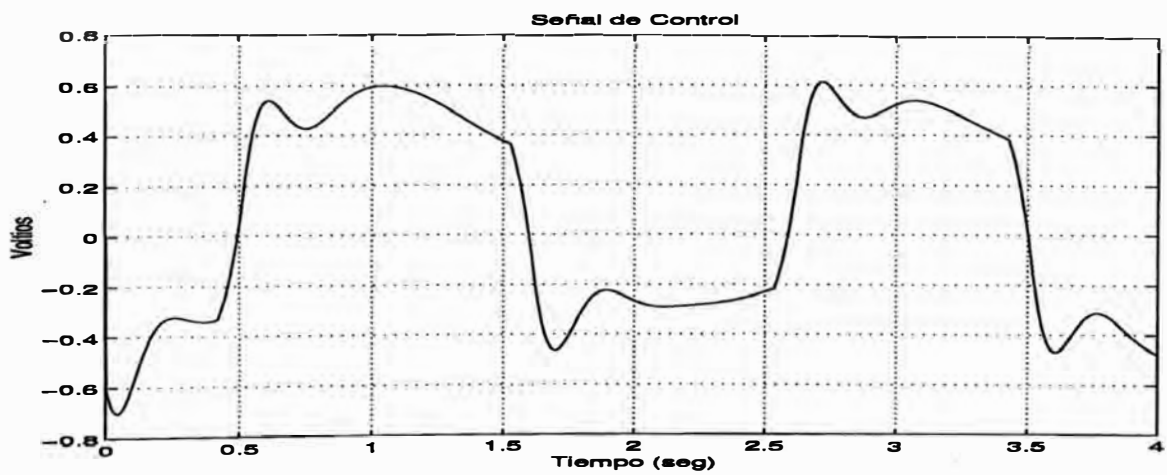


Figura 5.33: Señal de control proporcionada a la planta

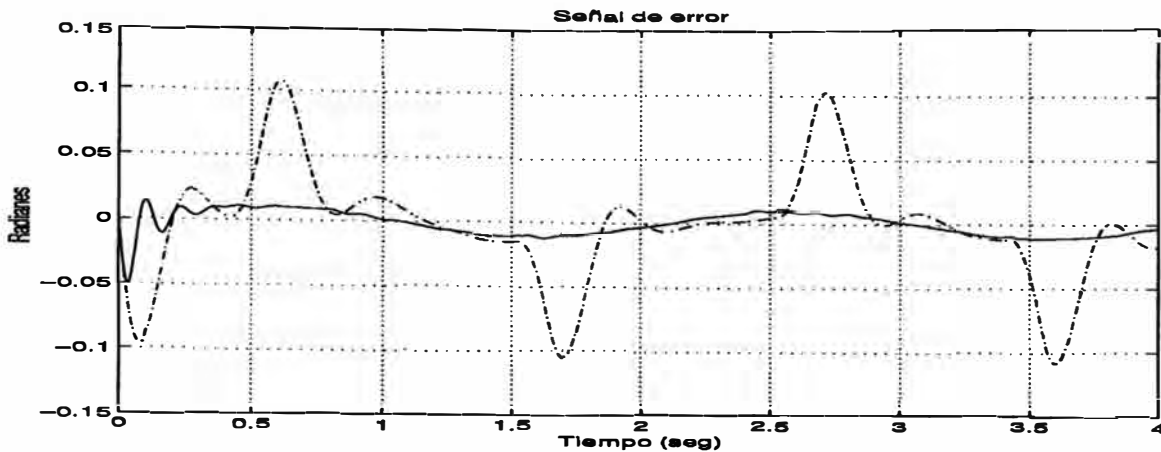


Figura 5.34: Error en el seguimiento de la trayectoria para un motor con carga, la línea continua muestra el error utilizando compensación y la punteada sin compensación

5.5 Control del BR1L utilizando realimentación de estado

De igual manera que en la sección anterior se ha linealizado la planta a controlar alrededor de $\theta = 0$. Entonces, a partir de la ecuación (5.44) se observa que la planta a controlar no tiene un integrador, entonces el principio básico en el diseño del servosistema es insertar un integrador en la trayectoria directa entre el comparador de error y la planta, como se puede observar en la figura 5.35. Del diagrama se obtiene:

$$\begin{aligned} \dot{\mathbf{x}} &= \mathbf{A}\mathbf{x} + \mathbf{B}u \\ y &= \mathbf{C}\mathbf{x} \\ u &= -\mathbf{K}\mathbf{x} + k_1 \xi \\ \dot{\xi} &= r - y = r - \mathbf{C}\mathbf{x} \end{aligned} \quad (5.49)$$

Aplicando los criterios de observabilidad y controlabilidad de la sección 5.3.1 se puede mostrar fácilmente que la ecuación (5.49) tiene un estado completamente controlable y observable. Además el sistema no tiene un cero en el origen para que cancele el integrador insertado.

Se supone que la entrada de referencia (función escalón) se aplica al tiempo $t = 0$, Entonces para $t > 0$, se puede escribir la dinámica del sistema por medio de una ecuación que es una combinación de las ecuaciones (5.49)

$$\begin{bmatrix} \dot{\mathbf{x}} \\ \dot{\xi} \end{bmatrix} = \begin{bmatrix} \mathbf{A} & \mathbf{0} \\ -\mathbf{C} & \mathbf{0} \end{bmatrix} \begin{bmatrix} \mathbf{x}(t) \\ \xi(t) \end{bmatrix} + \begin{bmatrix} \mathbf{B} \\ \mathbf{0} \end{bmatrix} u(t) + \begin{bmatrix} \mathbf{0} \\ 1 \end{bmatrix} r(t) \quad (5.50)$$

Si se diseña un sistema asintóticamente estable tal que $\mathbf{x}(\infty)$, $\xi(\infty)$, $u(\infty)$ tienden a valores constantes. Entonces, en estado estacionario $\dot{\xi}(t) = 0$, y se tiene que $y(\infty) = r$.

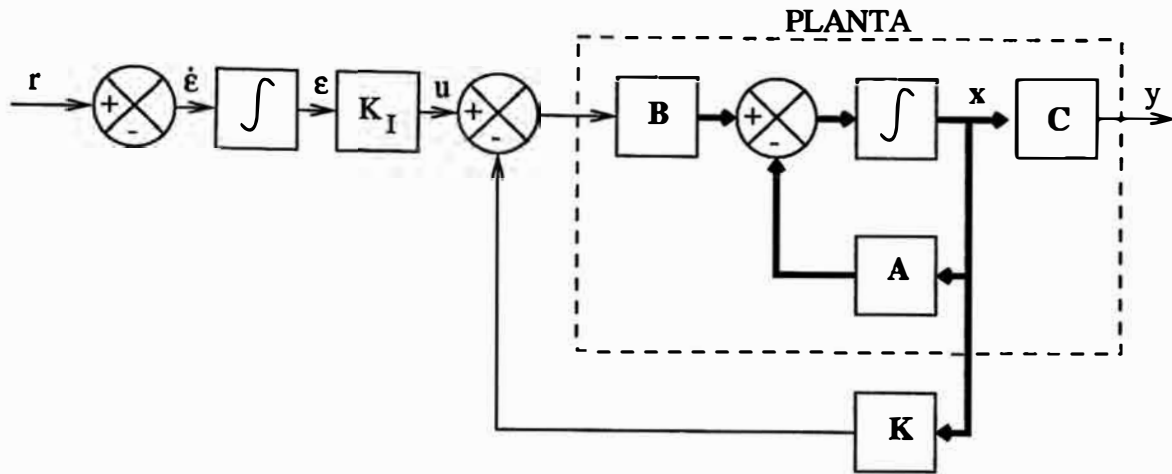


Figura 5.35: Esquema de un sistema de control usando realimentación de estado, la planta no tiene un integrador

Notar que en estado estacionario,

$$\begin{bmatrix} \dot{\mathbf{x}}(\infty) \\ \dot{\xi}(\infty) \end{bmatrix} = \begin{bmatrix} \mathbf{A} & \mathbf{0} \\ -\mathbf{C} & 0 \end{bmatrix} \begin{bmatrix} \mathbf{x}(\infty) \\ \xi(\infty) \end{bmatrix} + \begin{bmatrix} \mathbf{B} \\ 0 \end{bmatrix} u(\infty) + \begin{bmatrix} 0 \\ 1 \end{bmatrix} r(\infty) \quad (5.51)$$

Puesto que $r(t)$ es una entrada escalón, se tiene que $r(\infty) = r(t) = r(\text{constante})$.

Restando la ecuación (5.51) de la (5.50), se obtiene:

$$\begin{bmatrix} \dot{\mathbf{x}}(t) - \dot{\mathbf{x}}(\infty) \\ \dot{\xi}(t) - \dot{\xi}(\infty) \end{bmatrix} = \begin{bmatrix} \mathbf{A} & \mathbf{0} \\ -\mathbf{C} & 0 \end{bmatrix} \begin{bmatrix} \mathbf{x}(t) - \mathbf{x}(\infty) \\ \xi(t) - \xi(\infty) \end{bmatrix} + \begin{bmatrix} \mathbf{B} \\ 0 \end{bmatrix} (u(t) - u(\infty)) \quad (5.52)$$

Se define

$$\begin{aligned} \mathbf{x}(t) - \mathbf{x}(\infty) &= \mathbf{x}_e(t) \\ \xi(t) - \xi(\infty) &= \xi_e(t) \\ u(t) - u(\infty) &= u_e(t) \end{aligned} \quad (5.53)$$

donde $u_e(t) = -\mathbf{K}\mathbf{x}_e(t) + k_I\xi_e(t)$. Si definimos el vector de error $\mathbf{e}(t) = \begin{bmatrix} \mathbf{x}_e(t) \\ \xi_e(t) \end{bmatrix} = (n+1) - \text{vector}$. Entonces la ecuación (5.52) es

$$\dot{\mathbf{e}} = \hat{\mathbf{A}}\mathbf{e} + \hat{\mathbf{B}}u_e \quad (5.54)$$

donde

$$\hat{\mathbf{A}} = \begin{bmatrix} \mathbf{A} & \mathbf{0} \\ -\mathbf{C} & 0 \end{bmatrix}, \quad \hat{\mathbf{B}} = \begin{bmatrix} \mathbf{B} \\ 0 \end{bmatrix} \quad (5.55)$$

También $u_e(t)$ se puede reescribir como

$$u_e = -\hat{\mathbf{K}}\mathbf{e} \quad (5.56)$$

donde $\hat{\mathbf{K}} = [\mathbf{K} \mid -k_I]$.

La idea básica de diseñar este servosistema es diseñar un sistema regulador estable de dimensión $(n+1)$, que lleve al nuevo vector de error $\mathbf{e}(t)$ a cero, para cualquier condición inicial $\mathbf{e}(0)$.

Las ecuaciones (5.54) y (5.56) describen la dinámica del sistema regulador de orden $(n+1)$. Si el sistema definido por la ecuación (5.54), tiene estado completo controlable, entonces al especificar la ecuación característica deseada para el sistema, la matriz $\hat{\mathbf{K}}$ se puede ubicar por medio de la técnica de ubicación de polos, describiéndose sus pasos a continuación.

1. Verificar la matriz de controlabilidad, el rango de dicha matriz debe de ser orden n .
2. determinar los valores $a_i, i = 1, \dots, n$ a partir de la ecuación característica de la matriz \mathbf{A} , $|\mathbf{S}\mathbf{I} - \mathbf{A}| = s^n + a_1 s^{n-1} + \dots + a_{n-1} s + a_n$.
3. Determinar la matriz de transformación $\mathbf{T} = \mathbf{M}\mathbf{W}$, la cual transforma la ecuación de estado de la planta en su forma canónica controlable, la matriz \mathbf{W} tiene la siguiente forma:

$$\mathbf{W} = \begin{bmatrix} a_{n-1} & a_{n-2} & \dots & a_1 & 1 \\ a_{n-2} & a_{n-3} & \dots & 1 & 0 \\ \vdots & \vdots & & \vdots & \vdots \\ a_1 & 1 & \dots & 0 & 0 \\ 1 & 0 & \dots & 0 & 0 \end{bmatrix} \quad (5.57)$$

4. Utilizando los valores propios deseados, es decir los polos deseados de la función de transferencia del sistema, escribir la ecuación característica polinomial $(s - u_1)(s - u_2) \dots (s - u_n) = s^n + \alpha_1 s^{n-1} + \dots + \alpha_{n-1} s + \alpha_n$; donde $u_i, i = 1, \dots, n$ son dichas raíces deseadas.
5. La matriz de ganancia de realimentación de estado se determina de la formula siguiente: $\mathbf{K} = [\alpha_n - a_n \mid \alpha_{n-1} - a_{n-1} \mid \dots \mid \alpha_1 - a_1] \mathbf{T}^{-1}$.

El servo sistema se diseña para seguir la trayectoria: $r(t) = -\sin(\pi t) + \sin(\frac{\pi t}{4})$. Los polos deseados del sistema son los siguientes: $u_1 = -40, u_2 = -15\sqrt{2} - 15i, u_3 = -15\sqrt{2} + 15i$, el esquema de simulación utilizado en simulink se muestra en la figura 5.36. El subsistema Motor-Pendulo de la figura 5.36 es igual al de la figura 5.24 pero sin las realimentaciones de compensación.

La posición del BR1L con respecto al tiempo se muestra en la figura 5.38. Se están considerando despreciables los efectos de la fricción estática y de Coulomb.

La señal de control $u(t)$ transmitida a la planta se muestra en la figura 5.38.

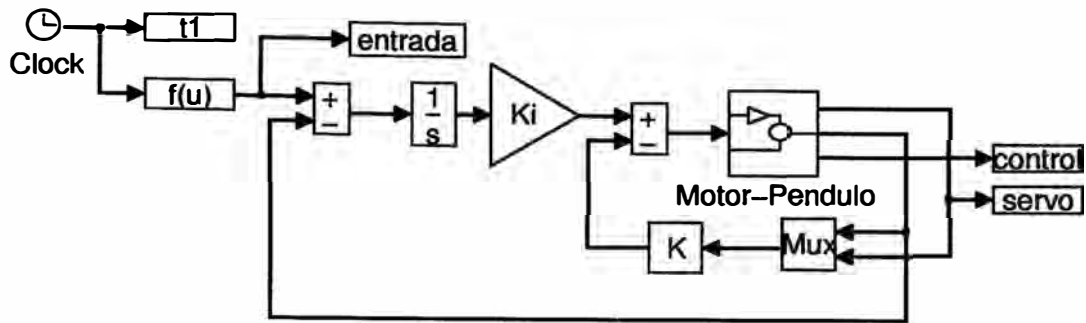


Figura 5.36: Esquema de simulación del sistema de control usando realimentación de estado.

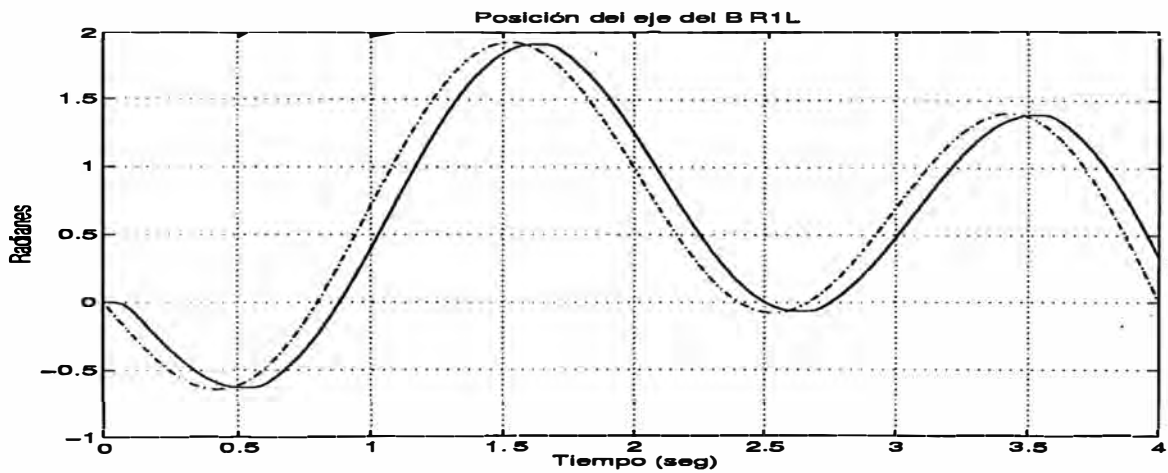


Figura 5.37: Señal de salida del sistema de control donde se muestra el ángulo del eje del BR1L. La línea continua es la trayectoria seguida por el BR1L y la punteada es la trayectoria de referencia.

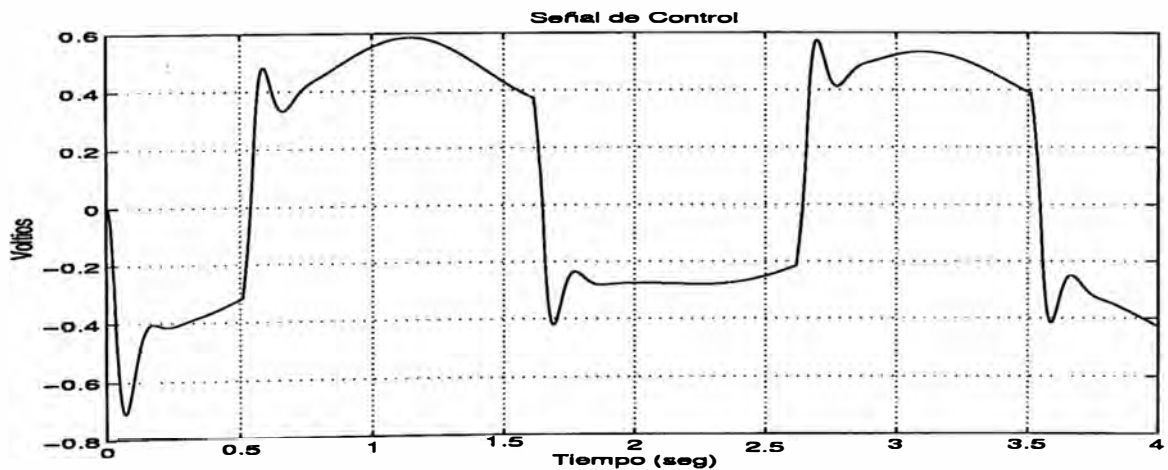


Figura 5.38: Señal de control proporcionada a la planta

CAPITULO VI DESARROLLO DEL PROYECTO

En este capítulo se presenta la implementación del sistema de identificación y control utilizando redes neuronales; realizar esto, implica también el diseño y desarrollo de la circuitería necesaria para el manejo del motor, así como también para la identificación de los parámetros del sistema ¹ utilizando métodos convencionales como la respuesta en frecuencia. Estos parámetros identificados fueron utilizados en las simulaciones de identificación y control del sistema realizadas en el capítulo 5.

6.1 Diseño e implementación del driver del motor DC.

El driver del motor esta formado por una etapa digital que genera la lógica de disparo para la etapa analógica. Estas etapas son descritas a continuación.

6.1.1 Circuito generador de la lógica de disparo.

En la figura A.2 del apéndice A se muestra el circuito generador de las señales PWM y \overline{PWM} con zona muerta. El LM3524 es un generador de PWM. El pin 9 (comparador) recibe la señal de control de la Tarjeta de adquisición de Datos Lab-PC+. El rango de tensión en el pin 9 puede variar entre 0.8 a 3.7 voltios; un valor superior o inferior satura al LM3524. La base de tiempo se controla por R_{27} y C_5 trabajándose aproximadamente a una frecuencia de 15.4Khz. En C_A y C_B se forman dos señales pulso desfasadas 180 grados con un duty cycle limitado a variar de 0 a 50%. La suma de estas señales da a lugar una señal PWM con un duty cycle variable de 0 a 100%. El efecto de R_{28} es sumar las señales de C_A y C_B . Los condensadores C_3 y C_4 se utilizan para filtrar el ruido.

En la figura A.1 se observa el diagrama de tiempos de la generación de PWM y \overline{PWM} con zona muerta, en donde se están considerando los retardos de las compuertas lógicas utilizadas. La señal PWM se muestra en (a). \overline{PWM} (b) se deriva de U6F con un retardo T_a . U5A se utiliza como seguidor, aumentando el FAN OUT de OSC (c). U7B y U7A son disparados con los flancos de subida de PWM y OSC respectivamente. T_d es el retardo de U7A. El ancho del pulso de U7A-13 (d) está controlado por R_{30} y C_7 . Similarmente, T_b es el retardo de U7B. el ancho del pulso de U7B-5 (g) es controlado

¹El cual fue modelado en el capítulo 3

por R29 y C6. El tiempo muerto entre el flanco de bajada de PWM y el flanco de subida de PWM está dado por U7A-13 (d), el cual se adiciona a PWM generando la nueva señal de control $PWM_{OSN} = PWM + U7A - 13$ (f). El tiempo muerto para el siguiente flanco está dado por U7B-5 que se adiciona a PWM generado la nueva señal de control $PWM_{ON} = PWM + U7B - 5$ (g). En esta señal se observa un retardo T_c que se puede ignorar porque no será visible al conmutador, el cual tiene un retardo superior a T_c que es aproximadamente el de una compuerta lógica.

6.1.2 Circuito amplificador PWM.

En la figura A.3 se muestra la etapa amplificadora de PWM_{ON} u PWM_{OSN} . Su funcionamiento y un esquema general del mismo se muestra en la figura 3.2.

Un conmutador amplificador puede ser hecho cambiando la frecuencia de conmutación o el duty cycle o cambiando ambas. El efecto neto buscado es que el promedio de voltaje de salida sea proporcional a la tensión de comando. El método más común ² de un conmutador amplificador es la modulación por ancho de pulso (PWM). Aquí los dispositivos conmutadores ³ conmutan a una frecuencia constante, dando como resultado la variación tensión de salida entre dos valores extremos. Por variación del ancho de pulso o del duty cycle, el valor promedio de voltaje de salida puede ser cambiado en forma proporcional a la tensión de comando.

El conmutador amplificador implementado es de tipo "H"; su diagrama se muestra en la figura 3.2. La principal ventaja del tipo "H" es que sólo es necesario una fuente cuando se requiera alta tensión; los dispositivos conmutadores son conmutados en pares generando una tensión bipolar a la salida. El efecto de tiempo muerto, el cual se explica en la etapa disparo, es dejar al motor en el "aire"; como el motor tiene una inductancia, éste trata de mantener la corriente aumentando la tensión. Por consiguiente, hay que tener cuidado en la elección de la duración del tiempo muerto para minimizar este efecto. En el circuito implementado, un buen valor se encuentra entre 1 y $3\mu\text{seg}$.

El amplificador tipo "H" se implementa con dispositivos conmutadores Mosfet's Buz 80 y una fuente de 24 voltios; para disparar los Mosfet's es necesario una circuiteria adicional para que genere la tensión de disparo en cada conmutador. En la figura A.3 el circuito de disparo de Q14 consiste de dos transistores Q1 y Q3 los cuales forman un amplificador que trabaja en clase B. Este amplificador se maneja por un una compuerta NOT de colector abierto que dependiendo de la señal PWM_{OSN} llega a saturar o cortar a Q1. El propósito de Q3 es absorber las corrientes de fuga de Q14. El circuito de disparo de Q5 consiste también de un amplificador que trabaja en clase B, el cual esta formado

²El cual se a implementado

³Mosfet's en el driver implementado

por $Q2$ y $Q4$; como la tensión de disparo esta dada por V_{GS} ⁴ y V_S no es tierra, entonces $Q2$ se alimenta por el condensador $C1$, el cual tiene como referencia a V_S ; es decir, $C1$ juega el papel de fuente; El diodo $D1$ con la fuente de $12V_{ol}$ dan la tensión y corriente necesaria a $C1$ y al circuito amplificador. $Q12$ sirve para llevar a corte o saturación a $Q2$; $Q12$ es a su vez se lleva a corte o saturación por la señal PWM_{ON} que pasa por dos compuertas NOT de colector abierto; $Q4$ sirve para absorber las corrientes de fuga de $Q5$. El circuito de disparo de $Q6$ y $Q7$ son idénticos al de $Q5$ y $Q14$, respectivamente.

6.2 Cálculo de los parámetros del sistema

En la presente sección se identifican los parámetros del modelo del BR1L encontrados en el Capítulo 3. Los parámetros identificados serán comparados con los encontrados usando la identificación con redes neuronales. A continuación se analiza un modelo simplificado del sistema, que se justifica cuando se realiza la identificación en el dominio de la frecuencia del motor previamente linealizado; es decir, después de compensar la fricción estática y de Coulomb. Para la identificación total del sistema, se tomaron también los parámetros encontrados en la documentación del motor.

6.2.1 Amplificador de potencia del PWM

En el análisis efectuado en el Capítulo 3 se asumió que el driver actúa como un amplificador lineal de ganancia K_{act} ; para verificar lo anterior se calculó el ancho de pulso en la salida del driver para una tensión de entrada. La relación entre el ancho del pulso y la tensión de salida se muestra en la figura 6.1, donde $V = 24$ es la tensión pico de salida del driver, $T = 65\mu\text{seg}$ es el periodo de la señal PWM y V_{DC} es la tensión de salida del driver. En la figura 6.2 se muestra la tensión de salida versus la tensión de entrada en el driver. Se tomaron 59 puntos para realizar esta gráfica, de la cual se deduce que el $K_{act} = 14.9$.

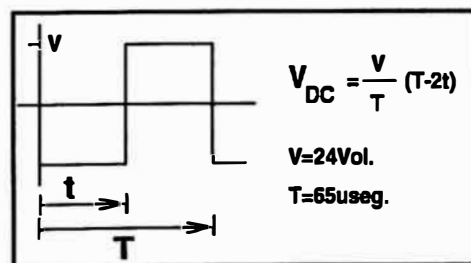


Figura 6.1: Relación ancho de pulso y tensión de salida

⁴ V_{GS} , diferencia de tensión entre los pines Gate y Source del Mosfet.

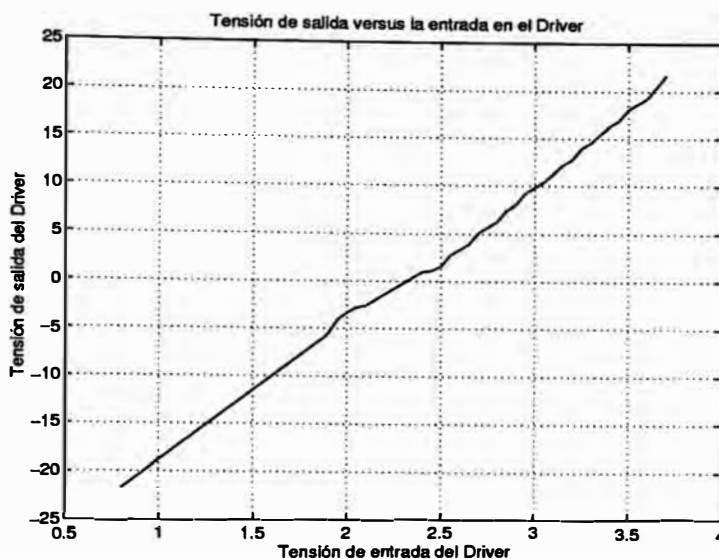


Figura 6.2: Gráfica tensión de salida versus tensión de entrada en el driver

6.2.2 Compensación de la fricción estática y de Coulomb

En esta sección se asume un modelo simplificado del sistema Driver-Motor, para el cual se hizo la inductancia del motor igual cero. Esto se justifica en la sección 6.2.3. Por tanto, la ecuación (3.12) se puede reescribir como:

$$J_{eff}\dot{\omega} = \frac{V_b K K_{act}}{R} - (b_{eff} + \frac{nEK}{R})\omega - f^*(n\omega) \quad (6.1)$$

En la ecuación (6.1), $f^*(n\omega)$ modela el torque de fricción estática y de Coulomb; a partir de la figura 6.3, se puede deducir una primera aproximación de $f^*(n\omega)$, el cual está dado por $f^*(n\omega) = C Sgn(n\omega)$. Cuando el motor está en eminente movimiento, el torque de fricción toma distintos valores ⁵; si la velocidad es distinta de cero, el torque toma dos valores constantes dependiendo del sentido de giro del motor ⁶.

La figura 6.3 presenta la gráfica de la velocidad angular del motor versus la tensión de entrada en el driver; la zona muerta se observa cuando la velocidad es cero, Esto es debido a la fricción estática en el motor. Una pequeña tensión de realimentación v_{rs} se adiciona a la entrada para compensar los distintos valores que toma el torque de fricción cuando la velocidad es cero. Esta tensión de realimentación está dada en la ecuación (6.2). Como la fricción estática ocurre a bajas velocidades cuando el motor está a punto de detenerse, la realimentación también tiene que ser aplicada cuando la velocidad del motor se encuentre dentro de un umbral ω_{th} alrededor de cero. El valor inicial de ω_{th} tomado, es el promedio de la velocidad anterior a cero del motor para

⁵Fricción Estática

⁶Fricción de Coulomb

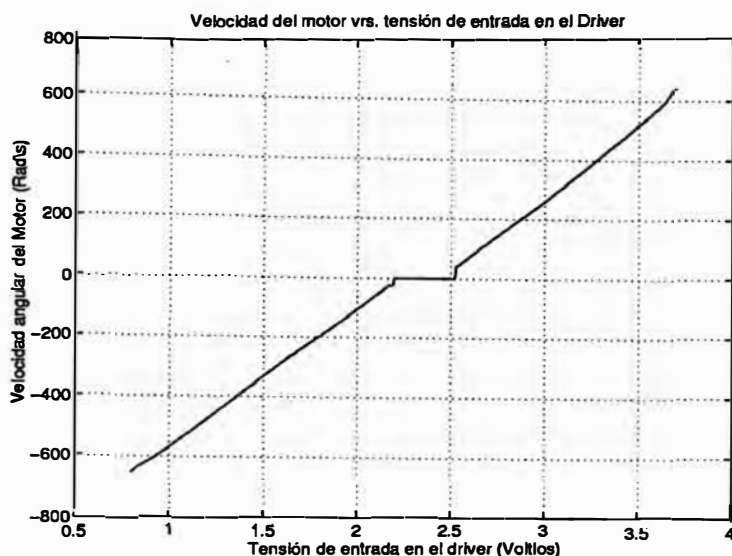


Figura 6.3: Gráfica de la velocidad en el motor para una entrada del driver

distintas experiencias similares a la figura 6.3.

$$v_{rs} = v_{zm} \operatorname{sgn}(V_b - \text{offset}) \quad \text{para} \quad |\omega_M| < \omega_{th} \quad (6.2)$$

En (6.2), v_{zm} es la mitad del ancho de la zona muerta medida de la figura 6.3. Para linealizar la ecuación (6.1) se debe adicionar una tensión a la entrada del driver tal que compense la fricción de Coulomb cuando $|\omega_M| \geq \omega_{th}$. Para realizar esto se debe de identificar el valor de C y K_{act} para adicionar la tensión $v_{rc} = \frac{CR}{KK_{act}} \operatorname{sgn}(V_b)$ cuando $|\omega_M| \geq \omega_{th}$. Sin embargo, C es pequeña en motores DC, siendo difícil su identificación en presencia de ruido. Por tanto, se usando el método de prueba y error para el cálculo de v_{rc} , el valor inicial tomado fue v_{zm} y se fue disminuyendo hasta encontrar el valor apropiado. La ecuación (6.3) resume el esquema de realimentación implementado:

$$v_{bias} = \begin{cases} .15 \operatorname{sgn}(V_b - \text{offset}) & \text{para } \omega_M < 22.6 \operatorname{radsec}^{-1}. \\ .13 \operatorname{sgn}(V_b - \text{offset}) & \text{en caso contrario.} \end{cases} \quad \text{Volts} \quad (6.3)$$

La justificación del esquema de realimentación (6.3) está basada en la dependencia del modelo de fricción con la velocidad descrita en la figura 6.4. La fricción estática es la responsable de mantener en reposo al motor, hasta que se excede un umbral de movimiento que equilibra el torque resultante de una corriente aplicada. El propósito de v_{rs} es mantener el punto de operación del motor en la zona de eminente movimiento, dependiendo de la dirección de la corriente aplicada.

Cuando se inicia el movimiento, la fricción de Coulomb y la fricción de viscosidad actúan juntos. Oponiéndose al movimiento del motor. El efecto no lineal de la fricción de Coulomb se reduce al aplicar v_{rc} dependiendo de la dirección del movimiento. El

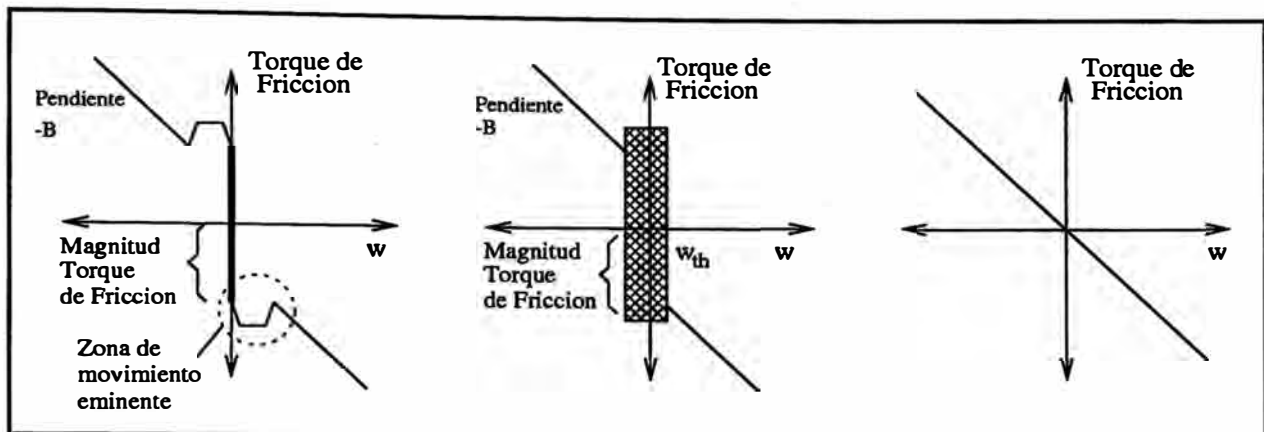


Figura 6.4: Gráfica de la velocidad en el motor para una entrada del driver

resultado obtenido es una dependencia aproximadamente lineal de la velocidad con la fricción. El experimento de la gráfica 6.3 es repetido ahora utilizando el esquema de compensación (6.3). Los resultados se muestran en la figura 6.5

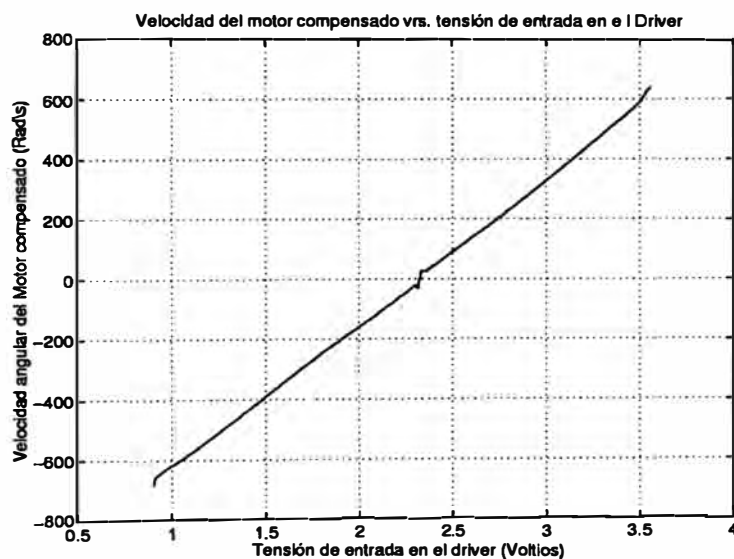


Figura 6.5: Gráfica de la velocidad en el motor para una entrada del drive; se utiliza compensación para reducir las no linealidades

6.2.3 Identificación en el dominio de la frecuencia del sistema linealizado

Para la respuesta en frecuencia experimental del sistema (Motor+Driver) se tomaron 105 frecuencias desde $\omega = 2Hz$ hasta $\omega = 106Hz$; la no linealidad se redujo utilizando el esquema de realimentación (6.3). Se exitó el sistema con una onda senoidal de amplitud 1V y fase 0 grados. la magnitud y fase de la salida se calculó usando la ecuación (6.4)

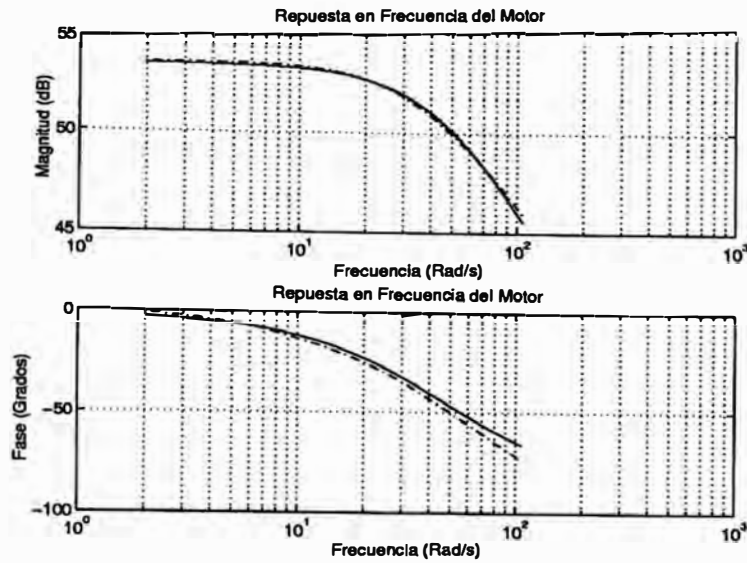


Figura 6.6: Respuesta en frecuencia del motor; se compensa la fase debida al cálculo de la velocidad y al filtro eliminador de ruido de la velocidad

$$\begin{aligned}
 A &= \frac{2}{N} \sum_{k=1}^N \omega(k) \sin(2\pi f k T_s) \\
 B &= \frac{2}{N} \sum_{k=1}^N \omega(k) \cos(2\pi f k T_s) \\
 M &= \sqrt{A^2 + B^2} \\
 \phi &= \arctan\left(\frac{B}{A}\right) - \phi_1 - \phi_2
 \end{aligned} \tag{6.4}$$

Donde f es la frecuencia de la prueba; T_s es el periodo de muestreo; $\omega(k)$ es la velocidad medida en la K_{esima} muestra; N número total de muestras; A y B son los coeficientes de Fourier de la salida a la frecuencia f . Como se conoce la magnitud M_i y ϕ_i de la entrada, la ganancia del sistema esta dada por $\frac{M}{M_i}$ y la fase por $(\phi - \phi_i)$. Cuando el proceso descrito se repite para varias frecuencias, se genera el gráfico de Bode del sistema muestreado. En la figura 6.6 se observa este gráfico para el motor usado en el sistema.

Se esta adicionando a la velocidad un retardo (ϕ_1) de $\frac{2\pi f T}{2}$ debido que la velocidad se deriva a partir de la posición. El ruido generado en la velocidad se reduce con un filtro Butterword pasa bajo con una frecuencia de corte de 100Hz, para filtrar el ruido en la posición; la gráfica de la respuesta en frecuencia del filtro se muestra en la figura 6.7, en donde se desprecia la ganancia del filtro en la ganancia total del sistema. La fase (ϕ_2) del filtro se compensa en la fase total del sistema.

Se utilizó un frecuencia de muestreo de 2KHz. La magnitud de la respuesta en

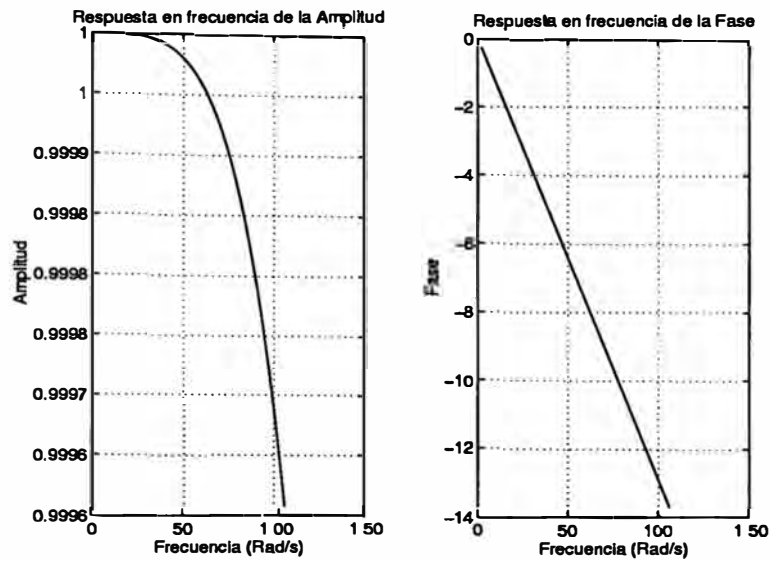


Figura 6.7: Respuesta en frecuencia del filtro, con ruido en la velocidad eliminado

frecuencia del motor se aproxima a un sistema de primer orden, por lo tanto se puede despreciar la inductancia del mismo para simplificar el sistema.

La función de transferencia linealizada del sistema se deriva de la ecuación (6.1).

$$J_{eff}\dot{\omega} = \frac{V_b K K_{act}}{R} - (b_{eff} + \frac{nEK}{R})\omega \quad (6.5)$$

donde, utilizando la Transformada Laplace:

$$\omega_M(s) = \frac{\frac{nKK_{act}}{RJ_{eff}}}{s + (\frac{b_{eff}}{J_{eff}} + \frac{nEK}{RJ_{eff}})} V_b(s) \quad (6.6)$$

La ganancia DC se encuentra con $s=0$:

$$\frac{\omega_M(0)}{V_b(s)} = \frac{nKK_{act}}{b_{eff}R + nEK} = 53.392dB = 467.32 \text{ rad seg}^{-1} V^{-1} \quad (6.7)$$

Como el polo del sistema aparece en $\omega = 47.0428 \text{ rad seg}^{-1}$ y se estima $K_{act} = 14.9$, a partir de los valores de la gráfica 6.2 se deriva lo siguiente:

$$\frac{b_{eff}}{J_{eff}} + \frac{nEK}{RJ_{eff}} = \omega_{3dB} = 47.0428 \quad (6.8)$$

De 6.7 y 6.8 se obtiene:

$$\begin{aligned} b_{eff} &= 7.0566 \times 10^{-5} \text{ Nm/rads}^{-1} \\ J_{eff} &= 5.6331 \times 10^{-5} \text{ Kg m}^2 \end{aligned} \quad (6.9)$$

A continuación se listan los parámetros del sistema.

Símbolo	Parámetro	Mét. de Identificación	Valor
R	Resistencia	Esp. del fabricante	7.38Ω
L	Inductancia	Esp. del fabricante	$4.64mH$
n	Reducción del Motor	Esp. del fabricante	19.741
K	Par Motriz constante	Esp. del fabricante	$31.071 \times 10^{-3} Nm/A$
E	Fuerza Contraelectromotriz	Esp. del fabricante	$31.0352 \times 10^{-3} Vs/Rad$
J_M	Momento de Inercia del Motor	Esp. del fabricante	$1.9066 \times 10^{-6} Kg m^2$
b_M	Coefficiente de Fricción Viscosa del Motor	Esp. del fabricante	$1.83 \times 10^{-6} Nm/Rad s^{-1}$
K_{act}	Ganancia del Driver del Motor	Regresión Lineal	14.9
J_{eff}	Momento de Inercia referido a la carga	Resp. en Frecuencia	$5.63 \times 10^{-5} Kg m^2$
b_{eff}	Coefficiente de Fricción Viscosa ref. a la carga	Resp. en Frecuencia	$7.05 \times 10^{-5} Nm/Rad s^{-1}$
m	Masa de la varilla	Medición	0.06377 Kg
L_o	Longitud de la varilla	Medición	0.776 m

Tabla 6.1: Tabla de Parámetros del Sistema

6.3 Identificación neuronal

En esta sección se describe mediante diagramas de flujo los pasos realizados para la adquisición de datos del motor (velocidad, posición) con una base de tiempo adecuada. También se describe la implementación del algoritmo de entrenamiento de la red.

6.3.1 Adquisición de datos

La tarjeta de adquisición de datos (TAD) cuenta con 3 contadores-temporizadores, los cuales se usan para generar ondas cuadradas con un ciclo de trabajo de 50%, estas se utilizan como base de tiempo en el algoritmo de adquisición de datos. Mediante software se tiene la capacidad de programar el periodo de los pulsos de los contadores, así como también de reconocer un flanco de subida o de bajada. En nuestros programas se ha tomado como referencia, un flanco de subida para el inicio de un periodo de muestreo. La TAD tiene la capacidad de enviar 8 bits de datos por el bus ISA de la PC; entonces, para la lectura de la posición se tienen que multiplexar los dos bytes generados por el sensor de posición en el PLD, esto se realiza utilizando una señal de control adecuada en la adquisición del byte menos y más significativo respectivamente. Como se puede observar esta lectura de datos utiliza un método asíncrono, al no conocer el estado de los contadores en el instante de lectura. En lecturas de datos experimentales se observan ruidos aleatorios en el tiempo de adquisición; para contrarrestar este problema se tienen que validar los datos leídos haciendo dos lecturas consecutivas, luego estas se comparan y si son iguales se valida la lectura.

A continuación se presenta un diagrama de tiempo para la adquisición de datos.

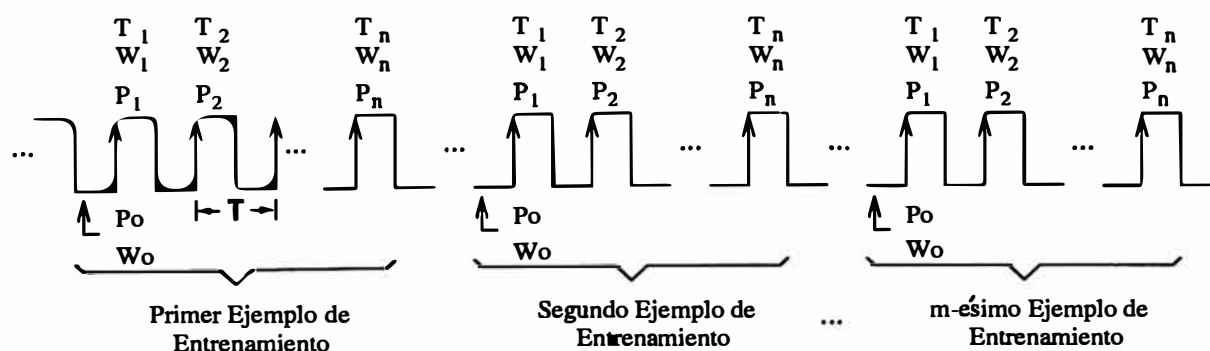


Figura 6.8: Diagrama de tiempo de la adquisición de datos

En la figura 6.8 se observa que un ejemplo de entrenamiento está formado por n periodos (T seg.) de tiempo y el conjunto de entrenamiento de la red está formado por m ejemplos de entrenamiento. P_i, W_i, T_i $i = 1, \dots, n$; son la posición, velocidad angular y tensión de entrada en el i -ésimo instante, respectivamente. El diagrama de flujo básico del algoritmo de adquisición de datos es el siguiente (figura 6.8):

Como la salida de los contadores varía entre 0 y 65536 se encontrarían discon-

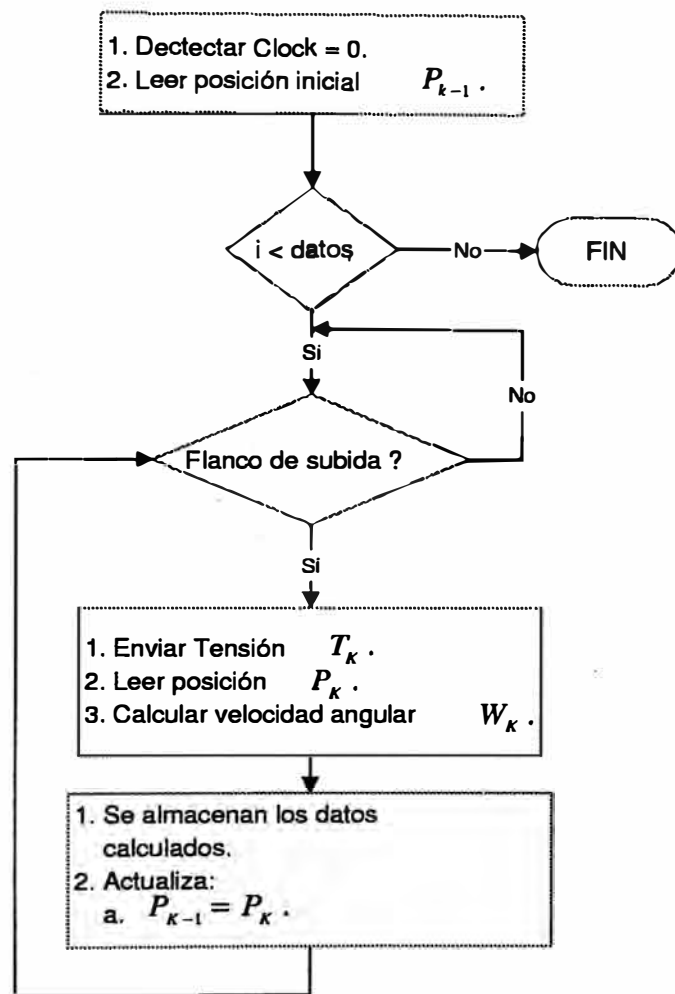


Figura 6.9: Diagrama de flujo de la adquisición de datos

tinuidades en el cálculo de la velocidad y posición; este problema se resolvió trabajando con los incrementos de la posición en cada instante de adquisición de datos. Cuando se encuentren incrementos excesivos (esto solo sucede cuando se pasa por los límites de los contadores), se adiciona o sustrae 65536 a los incrementos, compensando de esta forma las discontinuidades.

En el diagrama de flujo de la figura 6.9, se realiza primero la inicialización de la TAD, después en el instante que $\text{clock} = 0$ se inicializa la posición y velocidad angular. Luego, se detecta un flanco de subida, considerándose para este primer flanco $t = 0$; también se lee la posición para calcular la posición total y la velocidad angular (como se explicó anteriormente se está trabajando con incrementos de posición). Finalmente, se almacenan los datos calculados y se repiten los pasos anteriores para un nuevo flanco de subida hasta que $i = \text{datos}$, donde $\text{datos} = nm$, que es el instante de tiempo final de duración de un ejemplo de entrenamiento.

6.3.2 Entrenamiento de la red neuronal

En esta sección se muestra mediante un diagrama de flujo la implementación del algoritmo de retropropagación para el entrenamiento de la red. En el lenguaje de programación "C", una red neuronal se representa por medio de una matriz. Por tanto, las operaciones que se realizan son matriciales. En [6] se encontró una serie de rutinas para la realización de operaciones entre matrices, vectores y la combinación de los mismos.

La identificación de los parámetros de la planta se pudieron realizar "on-line", pero se realizó "off-line" para no reducir el tiempo de vida del motor por excesivos entrenamientos ((1) Sin la varilla, (2) Con la varilla). Primero, se realiza la inicialización de las variables a utilizar que son: (1) Las velocidades de aprendizaje g_1 y g_2 las cuales tienen que ser menores que 0.01 para evitar inestabilidades en la identificación. (2) Los parámetros lineales G_{1K}, G_{2K} se inicializaron con valores entre $0 < G_{1K}, G_{2K} < 1$. (3) Para los pesos de la red neuronal w_{ij} se inicializaron en un principio con valores aleatorios entre $0 < w_{ij} < 1$; en este caso el algoritmo de entrenamiento convergía lentamente, luego, los pesos fueron inicializados con *Matlab*, por medio de la función *INITFF* del toolbox de redes neuronales, aquí se observa que la red converge más rápidamente. (4) Se inicializa $\text{Error} > \text{umbral}$ para poder entrar al lazo; umbral es un valor para el cual se considera que el entrenamiento de la red ha terminado; en el programa se ha considerado $\text{umbral} = 0.1$. Segundo, se leen del disco los datos iniciales de la tensión de entrada T_0 , posición angular P_0 y velocidad angular w_0 ; luego se calcula la salida inicial de la red S_{K-1} , para una matriz de pesos inicial W_0 y una entrada w_0 . Tercero, se entra, al lazo de entrenamiento en sí; leyéndose los datos T_K, w_K, P_K para el instante K . Cuarto, se realiza la estimación de la velocidad angular ew_K usando en un principio el modelo de identificación "serie-paralelo", para finalmente utilizar el modelo "paralelo"

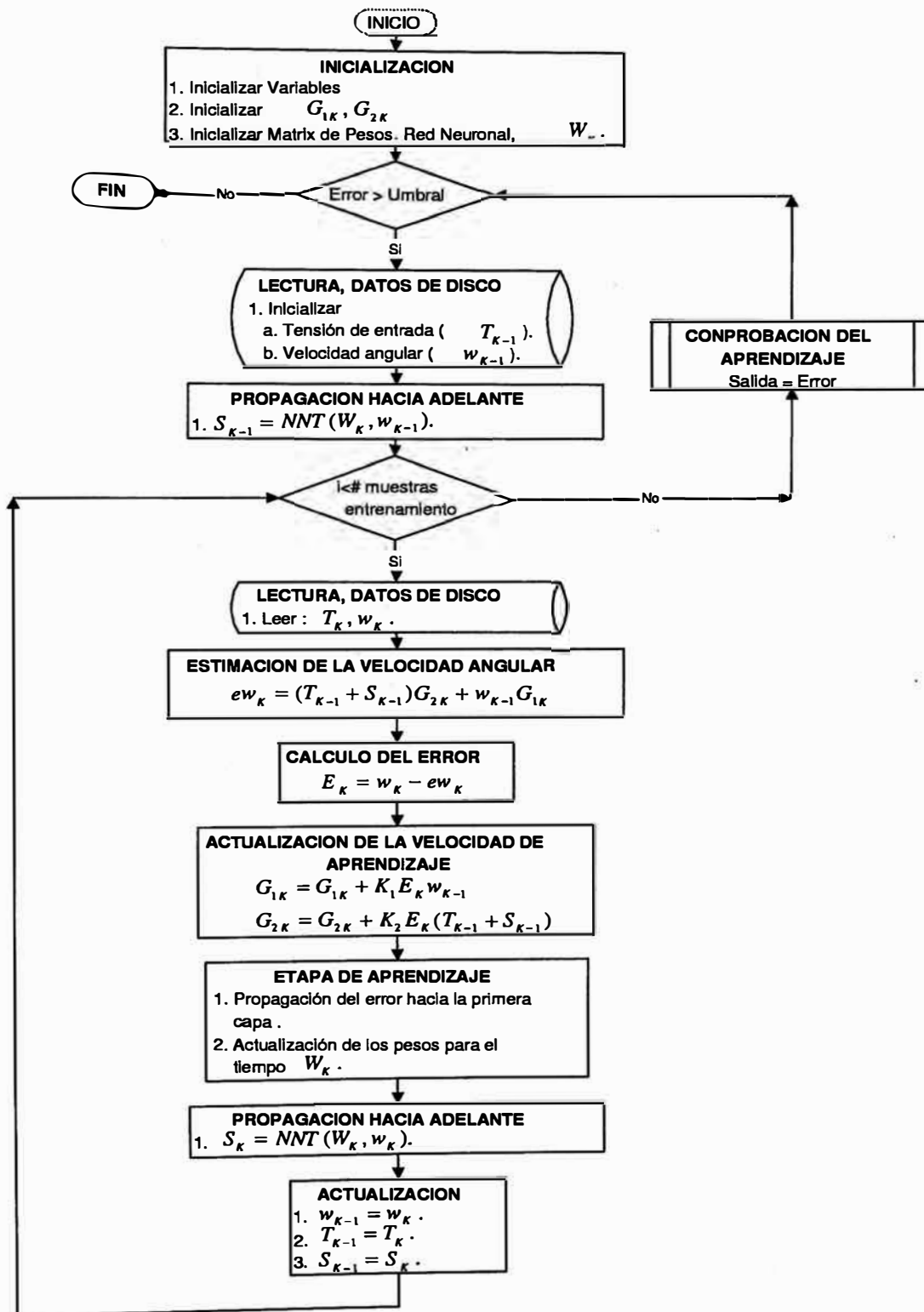


Figura 6.10: Diagrama de flujo del algoritmo de aprendizaje

cuando se observe que los valores lineales a identificar tienden a valores estables, o si la variable *Error*, comienza a disminuir lentamente. Quinto, se calcula el error entre la velocidad angular real y estimada, la que se utiliza en la actualización de las velocidades de aprendizaje. Sexto, se realiza la etapa de aprendizaje, que consiste de funciones que realizan la retropropagación del error y actualizan los pesos de la red neuronal para el siguiente lazo de entrenamiento. Estas funciones se realizaron en base a las rutinas de entrenamiento utilizando el algoritmo de backpropagation del toolbox de redes neuronales de Matlab. Octavo, se realiza la propagación hacia adelante con los nuevos pesos de la red calculados y se actualizan los valores w_{K-1} , T_{K-1} , P_{K-1} para ejecutar otra vez el lazo de entrenamiento de la red, hasta que el valor de $i > N^{\text{ro de muestras}}$. Noveno, una vez terminada la ejecución del lazo de entrenamiento, se realiza la comprobación del aprendizaje de la red, resultando un *Error* que es comparado con el *umbral*, una vez que $Error < umbral$, el entrenamiento ha terminado.

En la figura 6.11 se muestra el diagrama de flujo de la comprobación del aprendizaje de la red, a diferencia del esquema de entrenamiento, figura 6.10, para el cálculo de la velocidad angular estimada se utiliza el modelo de identificación paralelo. En la propagación hacia adelante, la salida de la red es una función de la matriz de pesos encontrada en el entrenamiento de la red y de la velocidad angular estimada.

6.4 Software de control

El software del sistema de control se desarrolló en "C". El diagrama de flujo se muestra en la figura 6.12 en donde se realizan los siguientes pasos: (1) Inicialización de la tarjeta de adquisición de datos y de las variables utilizadas en el programa. (2) Detectar $clock=0$ y leer el primer dato de la posición, aquí se da inicio al algoritmo de control. (3) Se comprueba que la variable **k** sea menor a **Num. Puntos**. La variable **Num. Puntos** junto con la frecuencia de muestreo determinan el tiempo de duración del experimento. (4) Se detecta un flanco de subida. De igual forma que la adquisición de datos, un flanco de subida dá inicio a un periodo de control, en donde se realiza el cálculo de la señal de control para ese instante de tiempo. (5) Se estima la velocidad a partir de la posición. La posición se calcula como se explicó en la sección 6.3.1 y luego se calcula la señal de control. (6) se envía la señal de control y se repite el bucle hasta que **k** alcance a **Num. Puntos**.

Las redes neuronales entrenadas con los algoritmos de aprendizaje son utilizadas para linealizar el BR1L. Este sistema linealizado se controla con un controlador PID para seguimiento de trayectorias. El controlador PID es diseñada analógicamente y para la implementación del control se discretiza utilizando una aproximación discreta; esto se mostró en la sección 5.3.2.

La parte principal de la implementación en "C" del algoritmo se muestra a contin-

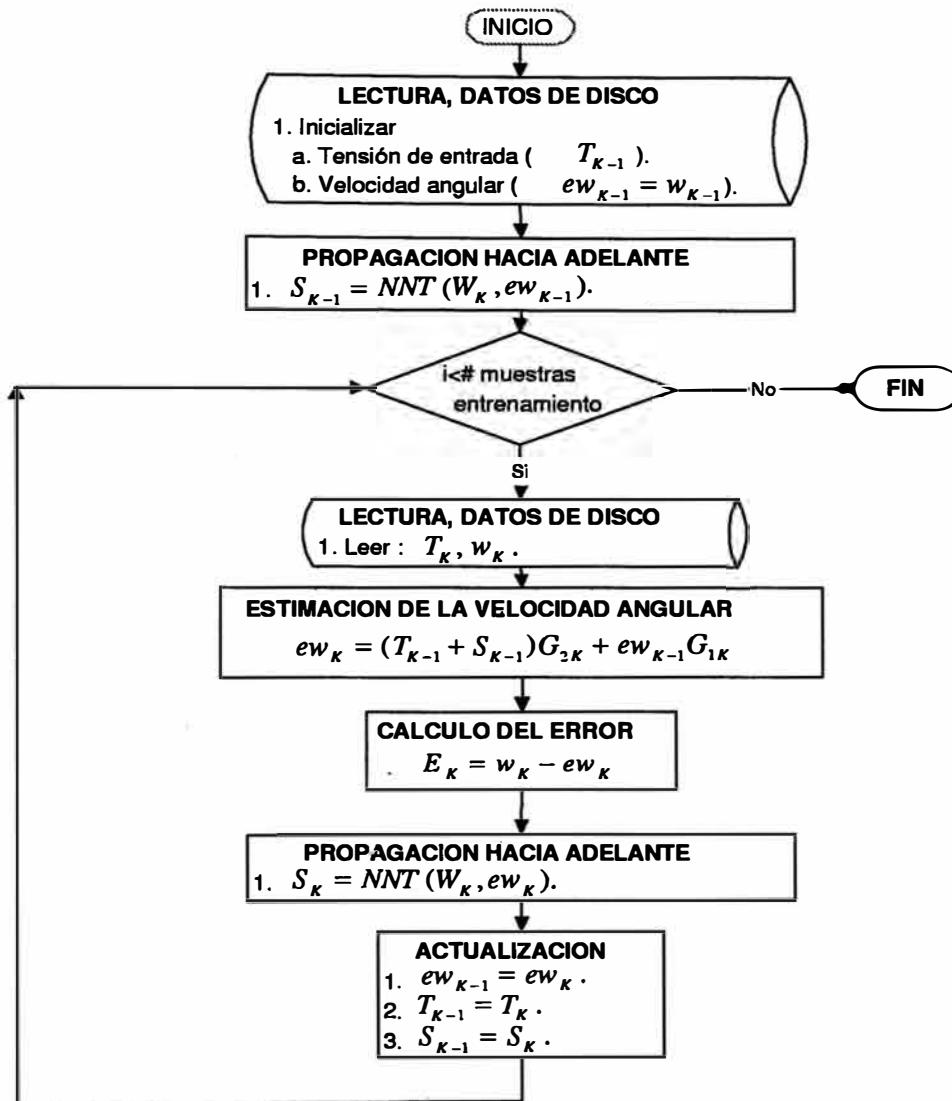


Figura 6.11: Diagrama de flujo de la comprobación del aprendizaje

uación:

```

while (k<=N_puntos)
{ /*while - 1*/
    act = NivelClock();
    if ((ant==0)&&(act==1)) // Detecta inicio de muestreo
    { /*if - 1*/

/*-----Generacion de la se~nal de control-----*/
Setpoint =.8*t-1.5*sin(3*t);
t +=0.001;

```



```

ek   = Setpoint - qk;           //Error
Pk   = Kp*(b*Setpoint - qk);    //Parte Proporcional
Ik   = Ik_1 + Kp*Ts*ek_1/Ti;    //Parte Integral
IW   = IW_1 + Ts*Sk_1/Tt;      //Integrador Antiwindup
Dk   = Td*Dk_1/(Td+N*Ts)-Kp*Td*N*(qk-qk_1)/(Td+N*Ts); //Parte Derivativa
/*----- Red neuronal que compensa la Friccion de Coulomb -----*/
dtansig_in(S1,W1,vel[k]*n,B1,rowB1);
dtansig(S2,S1,W2,B2,rowW2,colW2);
dtansig(compensacion,S2,W3,B3,rowW3,colW3);
/*- Red neuronal que compensa la componente sinusoidal del peso de la varilla-*/
dtansig_in(S1_,W1_,pos[k],B1_,rowB1);
dtansig(S2_,S1_,W2_,B2_,rowW2,colW2);
dtansig(compensacion_,S2_,W3_,B3_,rowW3,colW3);
/*-----Almacenamiento de la se~nal de Control-----*/
control[k]=Pk+Ik+Dk+IW+offset-*compensacion-*compensacion_;
/*-----Modelo del accionador del motor-----*/
if((control[k]<=3.7)&&(control[k]>=0.8)) Sk=0.0;
else if(control[k]<0.8){ Sk=.8-control[k];control[k]=.8;}
else if(control[k]>3.7){Sk=3.7-control[k];control[k]=3.7;}
/*-----Envio de la se~nal de control-----*/
SendVoltage(control[k]);

    /* if -1 */
    ant=act;
}/*while - 1*/

```

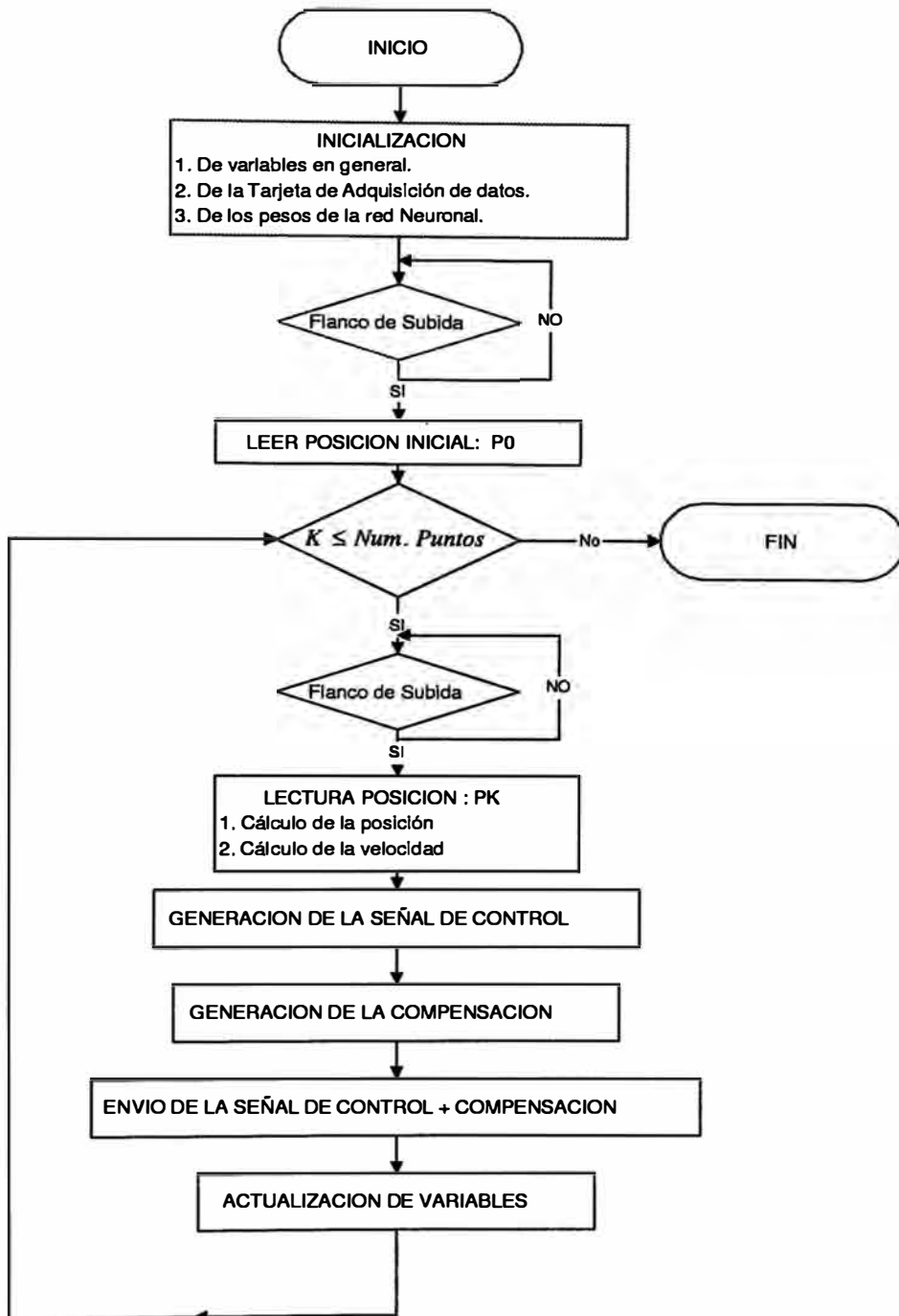


Figura 6.12: Diagrama de flujo del algoritmo de control

CAPITULO VII

RESULTADOS

En este capítulo se muestran los resultados de la implementación de los algoritmos de identificación y de control realizados en capítulos anteriores.

7.1 Adquisición de datos

Para la identificación de $f(n\omega)$ y $\frac{gL_0m}{2n} \sin(q)$ (ver ecuación (5.12)) los patrones de entrenamiento consisten de velocidades con una duración de 0.5 *seg* y 2 *seg*, respectivamente. Las tensiones de entrada que producen estas velocidades son escalones de 0.8 y 1.1 Volt., respectivamente. Se observa en la figura 7.1 que las velocidades incluyen una respuesta transitoria y una estacionaria, así como también un ruido debido a la cuantificación de los contadores.

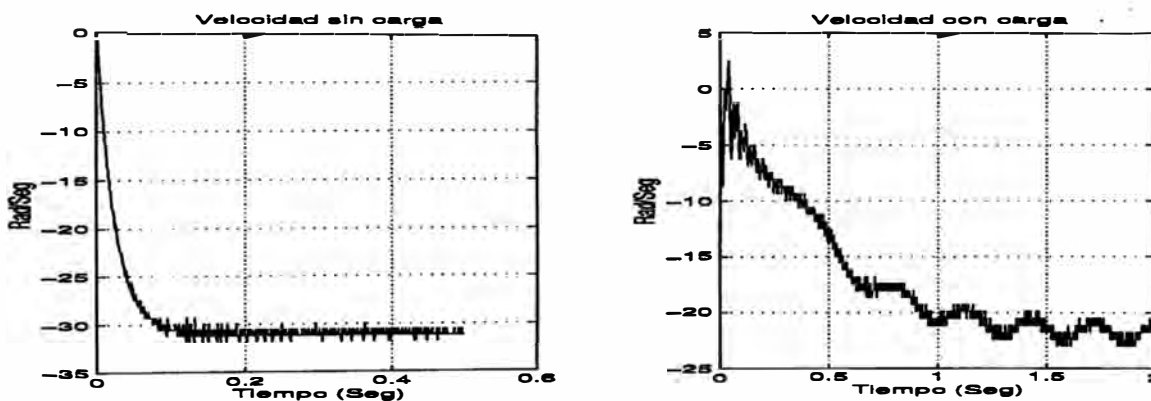


Figura 7.1: La primera figura muestra la velocidad de salida sin carga en el motor y la segunda, con carga en el motor.

7.2 Identificación de las no linealidades del BR1L

A continuación se muestra la identificación de la fricción estática y de Coulomb y la componente sinusoidal del peso de la varilla debido al efecto del movimiento de ésta en un plano vertical. La ecuación (5.13) muestra el modelo continuo del sistema BR1L y la ecuación (5.14) presenta el modelo discreto del BR1L utilizado en la identificación.

7.2.1 Identificación de la fricción estática y de Coulomb

De la misma forma que en las simulaciones, para realizar esta identificación no se ha considerado a la varilla. El modelo de identificación empleado se muestra en la ecuación (5.5) el cual se deduce de la ecuación (5.14) haciendo $m = 0$. La fricción estática y de Coulomb son identificadas por la red neuronal NNT $\Pi_{1,20,10,1}^3$ utilizando la ley (5.11), en donde se identifican también los parámetros lineales del modelo. Primero se utilizó el modelo serie-paralelo hasta que los parámetros lineales y el error de identificación converjan a un pequeño valor, y luego el modelo paralelo para disminuir aún más este valor. En la figura 7.2 se muestra la evolución del error cuadrático promedio por época y el error residual al final del entrenamiento para una época de entrenamiento.

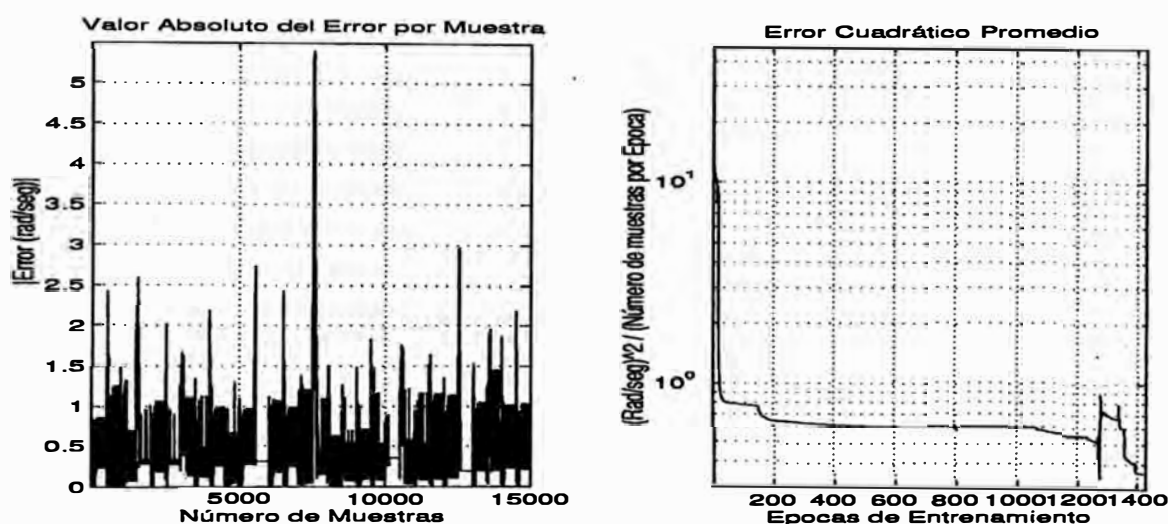


Figura 7.2: La primera figura muestra el error residual al final del entrenamiento y la segunda, el error cuadrático promedio.

En la figura 7.3 se compara la velocidad del sistema real con la velocidad estimada del modelo utilizado, la señal de entrada al sistema es un escalón de 0.8 Volt.

La figura 7.3 se muestra que el modelo empleado promedia el ruido de cuantificación de la velocidad real del sistema; también el error residual al final del entrenamiento mostrado en la figura 7.2 es del orden del ruido de cuantificación de la velocidad real. A continuación, en la figura 7.4 se muestra la identificación de la fricción estática y de Coulomb realizada por la red neuronal.

Es claro observar en la figura 7.4 que la fricción tiene distintos valores en cada sentido de giro del eje del motor (no es una función impar) y que para un sentido de giro, la fricción toma distintos valores dependiendo de la velocidad de giro; esto marca la diferencia del modelo $G_3 Sgn(n\omega)$ utilizado en las simulaciones. Inclusive se observa que la red ha estimado un offset de tensión adicional al empleado al realizar

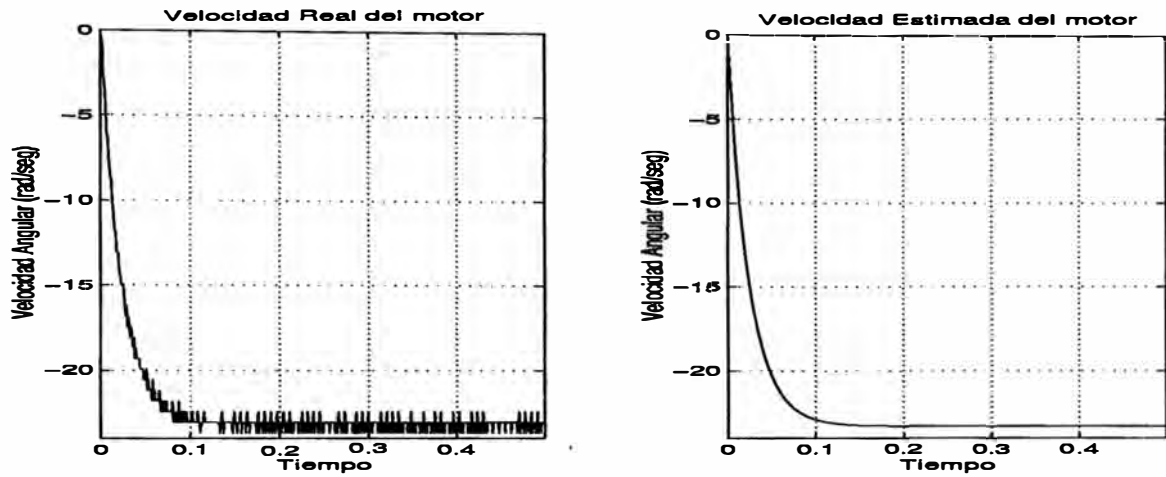


Figura 7.3: La primera figura muestra la velocidad del sistema real y la segunda, la velocidad estimada del modelo.

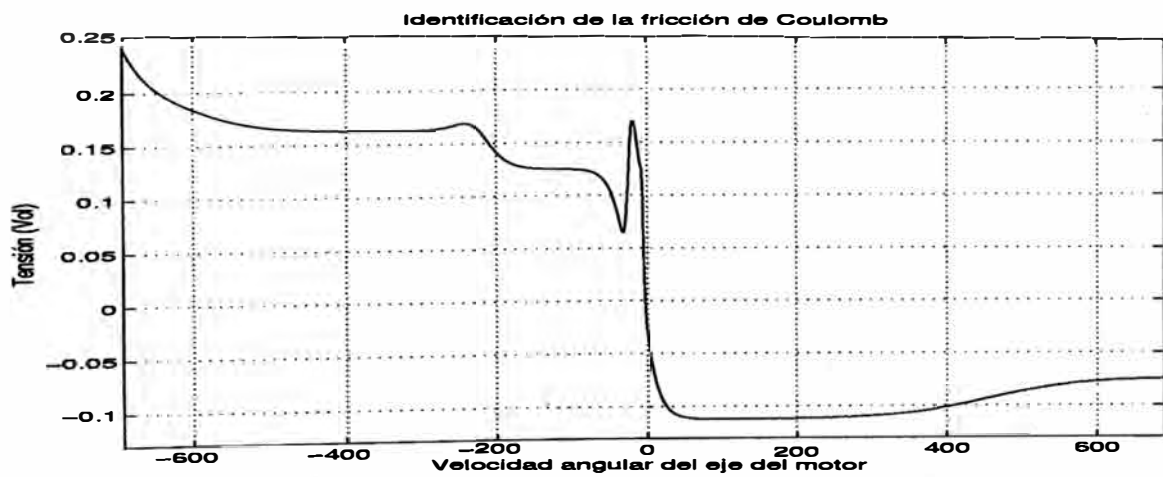


Figura 7.4: Identificación de la fricción estática y de Coulomb

el experimento, al no observarse una simetría promedio en la tensión identificada.¹ Al final de la identificación, los parámetros lineales en el modelo de la ecuación (5.5) convergieron a los siguientes valores: $G_1 = 0.9604$ y $G_2 = 0.9546$.

7.2.2 Identificación de la componente sinusoidal del peso de la varilla

La varilla al moverse en un plano vertical perpendicular al eje del motor genera una no linealidad debido al peso de ésta, la ecuación (5.14) muestra esta no linealidad ($\frac{RgL_0m}{2nKK_{act}} \sin(q_k)$), la que es identificada por la red neuronal NNT1 $\Pi_{1,20,10,1}^3$. El modelo de identificación empleado se muestra en la ecuación (5.15). La ley de identificación utilizada se muestra en la ecuación (5.17), la cual también se empleó para identificar los parámetros lineales del sistema BR1L. En la identificación se utiliza la red neuronal NNT entrenada en la sección anterior, la que compensa a la fricción estática y de Coulomb en este experimento. De la misma forma que en la sección anterior, primero se utiliza el modelo serie - paralelo y luego el paralelo.

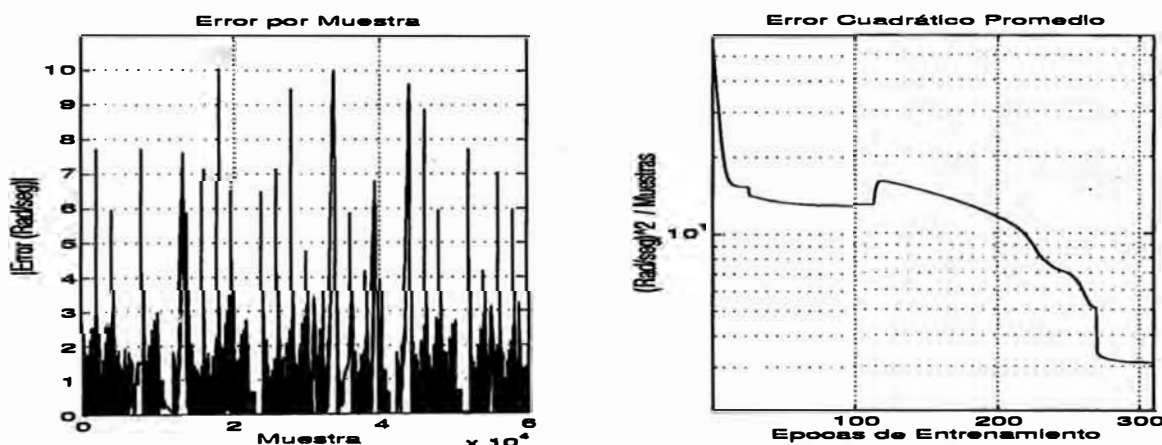


Figura 7.5: La primera figura muestra el error residual al final del entrenamiento y la segunda, el error cuadrático promedio.

En la figura 7.5 se muestra la evolución del error cuadrático promedio por época y el error residual al final del entrenamiento para un conjunto de entrenamiento. En la figura 7.6 se compara la velocidad del sistema real con la velocidad estimada del modelo después de identificar sus parámetros internos. La entrada a estos sistemas es una señal escalón de 1.0 Volt.

La velocidad del sistema BR1L en la figura 7.6 contiene ahora dos fuentes de ruidos: (1) El de la cuantificación. (2) El producido por la varilla al iniciar su movimiento que

¹Las tensiones identificadas nos dan una información indirecta de la fricción. La constante de proporcionalidad entre la tensión identificada y la fricción se puede calcular a partir de los parámetros lineales identificados.

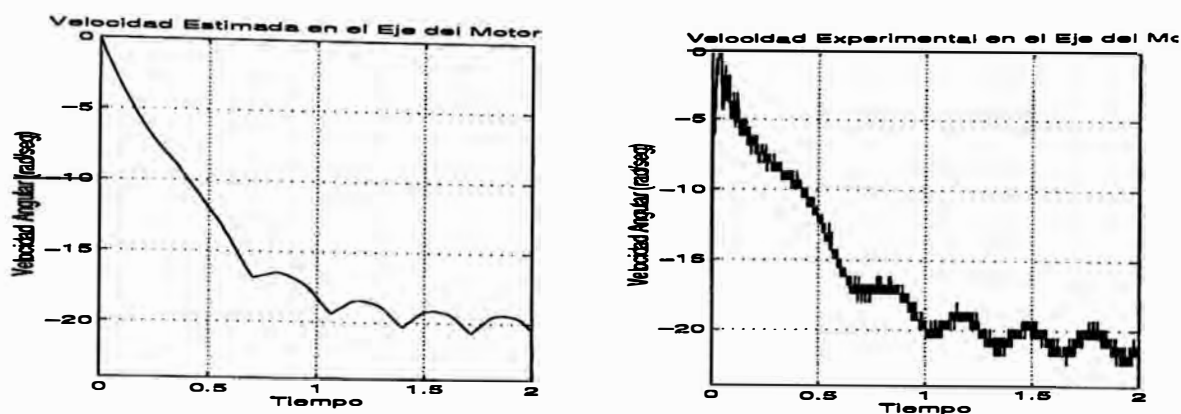


Figura 7.6: La segunda figura muestra la velocidad del sistema real y la primera, la velocidad estimada del modelo.

genera un estado oscilatorio en la velocidad que tiene una duración promedio de 0.25

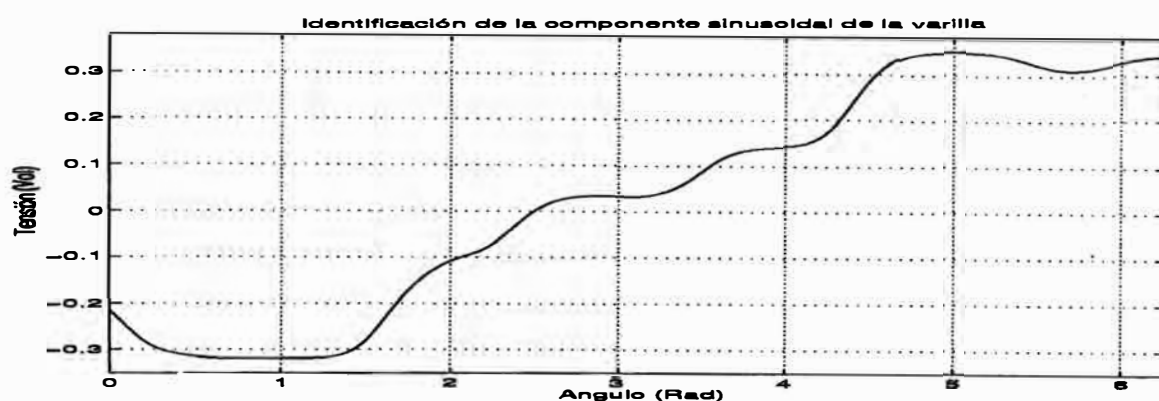


Figura 7.7: Identificación de la componente sinusoidal del peso de la varilla debida al movimiento de ésta en un plano vertical al eje del motor

seg para cada ejemplo de entrenamiento. El modelo identificado promedia estos ruidos eliminándolos. En la figura 7.5 se muestra el error residual al final del entrenamiento en donde se observan picos de ruidos de hasta 10 rad/seg debido al ruido explicado en (2), y un ruido promedio de 2 rad/seg causado por (1). En la figura 7.7 se muestra la identificación de $\frac{RgL_0m}{2nKK_{aet}} \sin(q_k)$ por la red neuronal NNT1. Aquí q_k es acotado en el rango de $0 \leq q_k \leq 2\pi$ y de la misma forma que en la sección anterior, la tensión identificada nos da una información indirecta del torque sinusoidal, debido al efecto del movimiento de la varilla. Los parámetros lineales del modelo (5.15) convergieron al final del entrenamiento a los siguientes valores: $G_1 = 0.9973$ y $G_2 = 0.0416$.

Como se puede observar $\frac{RgL_0m}{2nKK_{aet}} \sin(q_k)$ es un modelo ideal de la no linealidad a identificar. La velocidad estimada del modelo del sistema es calculada usando dos métodos: (1) Utilizando la no linealidad ideal. (2) Utilizando la red neuronal, en el segundo método se obtuvieron mejores resultados, esto se muestra en la figura 7.8.

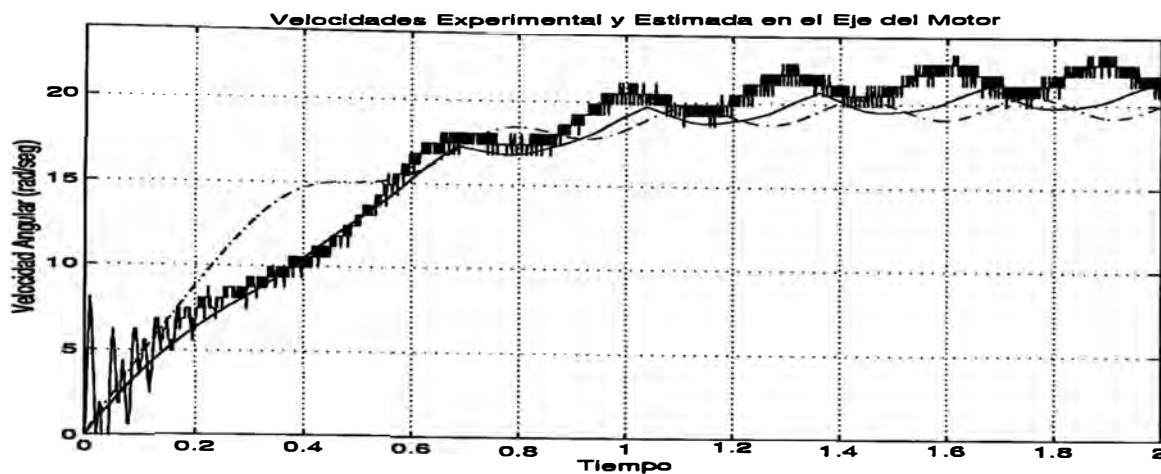


Figura 7.8: La línea punteada es la velocidad estimada usando un modelo ideal del torque sinusoidal. La línea continua es la velocidad estimada usando la red neuronal. La velocidad ruidosa es la respuesta del sistema BR1L para una entrada escalón de 3.5 Vol.

7.3 Compensación del sistema no lineal

Es claro observar en las ecuaciones (5.5) y (5.15) que las tensiones $G_4 \sin(q_k)$ y $G_3 \text{Sgn}(n\omega_k)$ son los torques no lineales trasladados hacia la tensión de entrada del sistema para tener acceso a ellos. La eliminación de estas no linealidades puede ser realizada por una realimentación de compensación. Es decir, generando una tensión de igual magnitud que $G_3 \text{Sgn}(n\omega_k)$ y $G_4 \sin(q_k)$ pero de signo contrario, de tal manera que al adicionarla a la tensión de entrada, elimine los torques de fricción no lineales. Las redes neuronales NNT y NNT1 se utilizaron para realizar esta realimentación de compensación.

7.3.1 Compensación del motor sin carga

El primer gráfico de la figura 7.9 muestra la velocidad del motor sin carga para una entrada $0.5 \sin(\frac{2\pi t}{3})$ sin utilizar compensación; el segundo gráfico compensa esta no linealidad. Se observa que la velocidad aumenta y la zona muerta del motor desaparece.

7.3.2 Compensación del motor con carga

En el primer gráfico de la figura 7.10 se muestra la velocidad del motor con carga para una entrada $\sin(\pi t)$ sin compensar. En el segundo gráfico se muestra el efecto de la compensación del sistema.

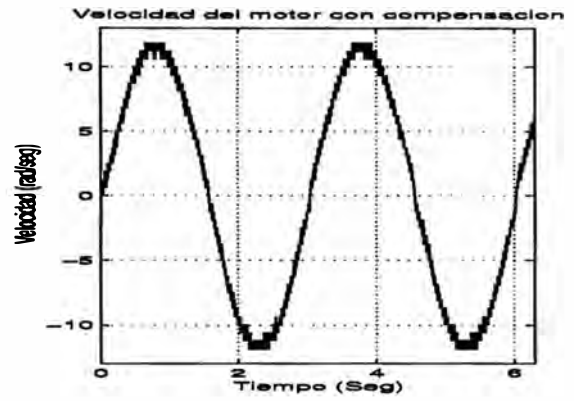
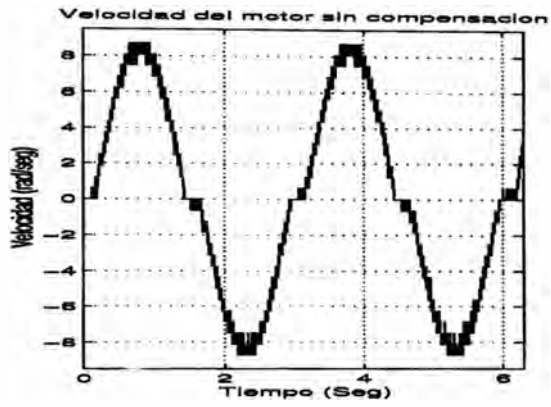


Figura 7.9: El primer gráfico muestra la respuesta del sistema sin compensar y el segundo, compensado.

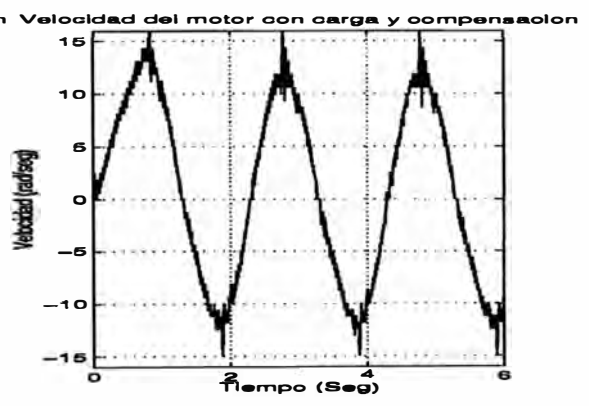
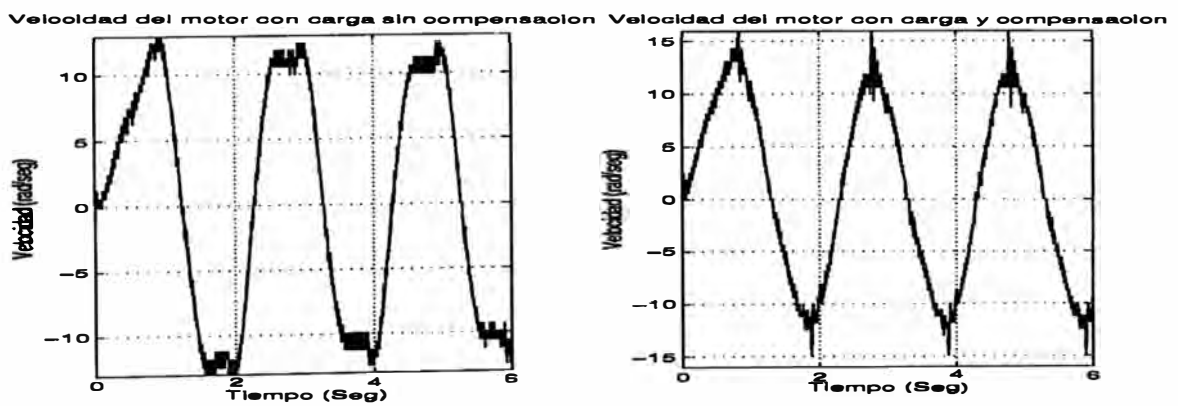


Figura 7.10: El primer gráfico muestra la respuesta del sistema sin compensar y el segundo, compensado.

7.4 Sistema de control implementado

Los sistemas de compensación encontrados se probaron implementado un sistema de control PID con una realimentación de compensación. Un esquema de este sistema se muestra en la figura 5.28. El funcionamiento de estos sistemas son comparados con un sistema de control PID simple. Primero se analizará el caso del motor sin la varilla y luego del sistema BR1L.

7.4.1 Control PID del motor sin varilla

En el gráfico de la figura 7.11 se muestra la respuesta del motor sin varilla. En el primer gráfico se observa el efecto de la zona muerta del motor y en el segundo se muestra su compensación. La trayectoria a seguir es $r = 0.8t - 3\sin(3t)$ para el tiempo $0 \leq t \leq 5$.

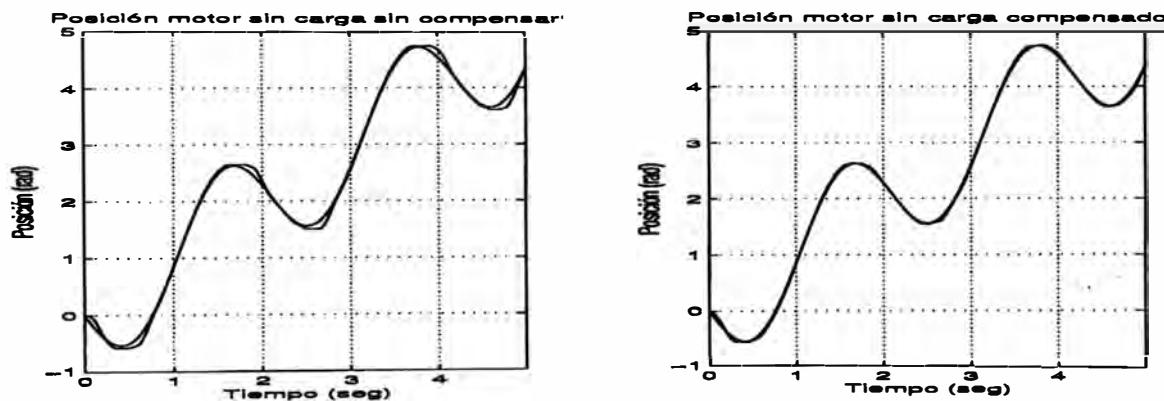


Figura 7.11: Control PID de un motor sin varilla

7.4.2 Control PID del sistema BR1L

En el gráfico de la figura 7.12 se muestra la respuesta del control del sistema BR1L. En el primer gráfico se muestra la respuesta sin compensación y en el segundo se muestra su compensación. La trayectoria a seguir es $r = 0.8t - 2.5\sin(3t)$ para el tiempo $0 \leq t \leq 5$. Para el control, se sintonizaron los parámetros del controlador en ambas experiencias.

La compensación disminuyó el error cuadrático promedio (ecuación (5.4)) de seguimiento de la trayectoria en la mitad, tal como se observa en la tabla 7.1. En el gráfico de la figura 7.13 se muestran estos errores.

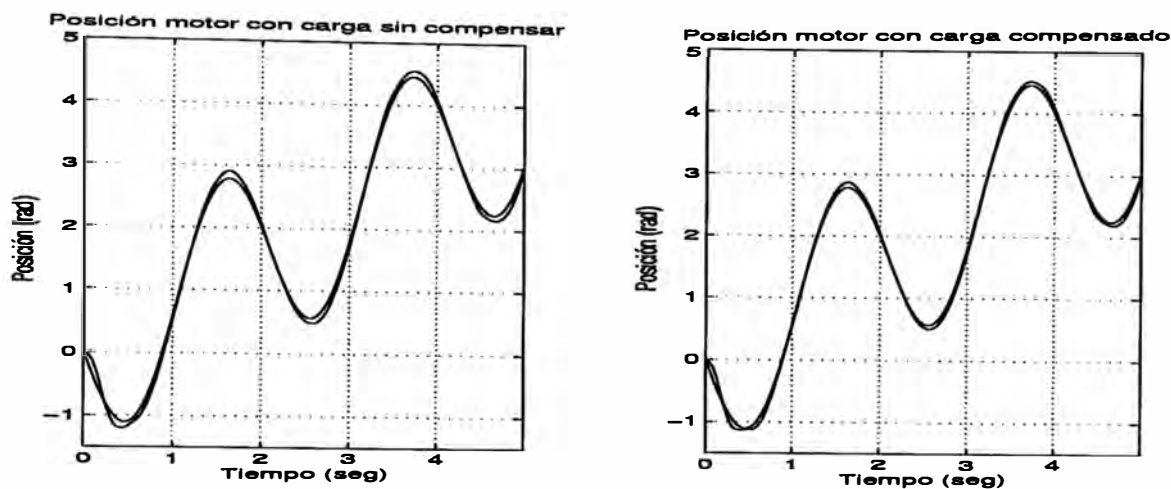


Figura 7.12: Control PID del sistema BR1L

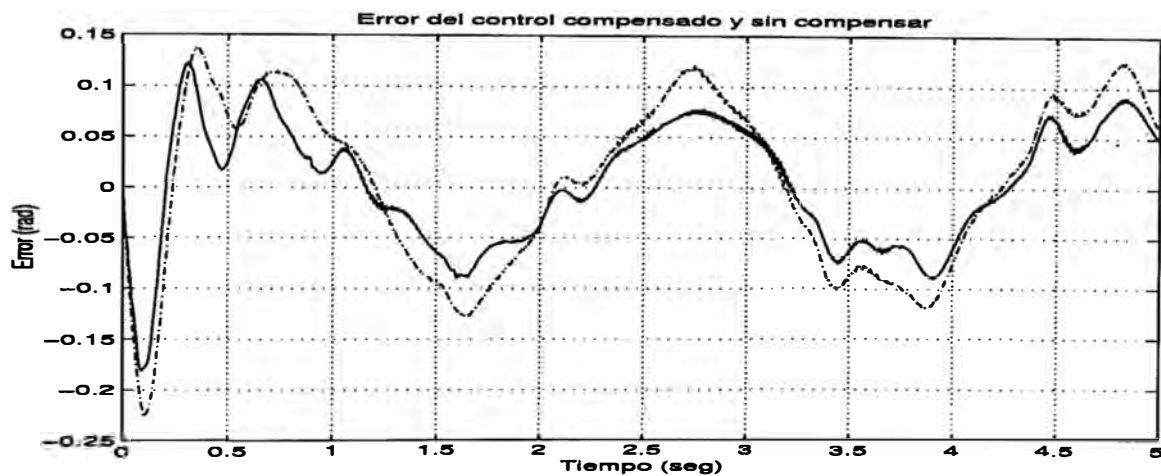


Figura 7.13: Error en el seguimiento de la trayectoria para el sistema BR1L. La línea continua muestra el error utilizando compensación y la punteada sin compensación

Motor	Error cuadrático promedio $\left[\frac{rad^2}{seg^2 \times (\text{Número Muestras.})} \right]$
Sin compensación	0.0071
Con compensación	0.0036

Tabla 7.1: Error cuadrático medio

CONCLUSIONES Y FUTUROS TRABAJOS

En esta sección se presentan las principales conclusiones derivadas de la adquisición de datos, identificación y control realizadas en el sistema BR1L.

1. Al analizar la respuesta en frecuencia del motor, se encontró que la respuesta del sistema se puede aproximar por un sistema de primer orden (ecuación (6.6)). Por tanto, la inductancia (L) del motor se puede despreciar. Esta característica del motor hace posible la utilización de la realimentación de compensación y de los modelos de identificación utilizados (ecuaciones (5.5) y (5.16)).
2. Las fricciones no lineales discontinuas estática y de Coulomb pueden ser compensadas utilizando un modelo ideal de torque (ecuación (6.3)). Este modelo se utilizó en la identificación de parámetros del motor usando el método de la respuesta en frecuencia, cuyos valores fueron utilizados sólo en las simulaciones. El modelo ideal considera que la fricción de Coulomb tiene el mismo valor para un sentido de giro del motor, lo cual no es cierto. La modelación de esta no linealidad utilizando redes neuronales demostró que el torque de fricción de Coulomb es una función de la velocidad (figura 7.4).
3. Se demostró que el torque no lineal debida a la componente sinusoidal del peso de la varilla se puede identificar satisfactoriamente utilizando una red neuronal. Se comparó la respuesta de la velocidad estimada de la varilla en el sistema BR1L utilizando un modelo ideal y neuronal con respecto a la velocidad real de la varilla del BR1L y se observó un mejor funcionamiento utilizando el modelo neuronal (figura 7.8).
4. Se demostró que los torques no lineales del sistema BR1L puede ser estimados satisfactoriamente utilizando una red neuronal $\Pi_{1,20,10,1}^3$.
5. Se demostró que las redes neuronales pueden formar parte de un sistema de realimentación de compensación. La figura 7.9 presenta la compensación de la fricción estática y de Coulomb y la figura 7.10 presenta la compensación del torque no lineal debida a la componente sinusoidal del peso de la varilla. Cuando la aceleración esta alrededor de cero, se observa distorsiones en la velocidad; esto se debe a que no se están considerando otras no linealidades como el backlash en los engranajes del motor y en el conector varilla - eje del motor, por ejemplo.

6. La velocidad pudo ser hallada satisfactoriamente a partir de la posición. El ruido en la velocidad debido a la cuantificación de los contadores puede ser despreciado cuando se realiza la identificación y control en el sistema BR1L.
7. El control del sistema linealizado usando redes neuronales compensa satisfactoriamente los términos no lineales del sistema BR1L. Se encontró que la señal de control generada esta dentro del rango lineal del amplificador de potencia.
8. También se encontró que la duración del algoritmo de control usando redes neuronales implementado fue de $0.6mseg$. Por tanto, la máxima frecuencia de muestreo que se puede utilizar es de 1700 Hz. Sin embargo, frecuencias mayores que 600Hz son suficientes para lograr un buen funcionamiento.
9. El sistema implemetado es capaz de trabajar con motores DC de mayor potencia. Todo depende de la corriente y tensión que manejan los conmutadores. La placa impresa del driver se ha diseñado de tal forma que los conmutadores puedan ser extraídos y reemplazados fácilmente. Por tanto, el estudio de otros tipos de control tales como controladores difusos, neuro-difusos, adaptivos, entre otros, son los futuros trabajos que pueden ser implementados sobre el sistema BR1L. Si la velocidad de procesamiento del algoritmo de control empleado fuera el problema, en la actualidad podemos usar PC's con más del doble de velocidad de la que se utilizó, para zanjar tal dificultad.

ANEXO A
CIRCUITOS DEL DRIVER DEL MOTOR

En el presente apéndice se presenta la descripción de los componentes que conforman el circuito generador PWM y el amplificador de potencia PWM. Adicionalmente se presenta los diagramas circuitales de los mismos.

Nombre	Descripción	Valor
R27	Resistencia	$4.7K\Omega$
R28	Resistencia	330Ω
R29	Potenciometro	$1K\Omega$
R30	Potenciometro	$1K\Omega$
C3	Condensador Cerámico	$0.1\mu F$
C4	Condensador Cerámico	$0.1\mu F$
C5	Condensador Cerámico	$0.015\mu F$
C6	Condensador Cerámico	$9nF$
C7	Condensador Cerámico	$1.8nF$
U3	Compuerta Or, 74LS32	—
U4	Generador de PWM, LM3524	—
U5	Compuerta Or, 74LS32	—
U6	Compuerta Not, 74LS02	—
U7	Monoestable, 74LS123	—

Tabla A.1: Tabla de elementos del circuito generador de PWM con tiempo muerto

Nombre	Descripción	Valor	Nombre	Descripción	Valor
R5	Resistencia	4.7K Ω	R24	Resistencia	3.3K Ω
R6	Resistencia	30 Ω	D1	Diodo Rectificador	—
R7	Resistencia	30 Ω	D2	Diodo Rectificador	—
R8	Resistencia	30 Ω	C1	Condensador Cerámico	22 μF
R9	Resistencia	30 Ω	C2	Condensador Cerámico	22 μF
R10	Resistencia	3.3K Ω	C15	Condensador Thantalo	22 μF
R11	Resistencia	3.3K Ω	Q1	Transistor, 2N2222	—
R12	Resistencia	3.3K Ω	Q2	Transistor, 2N2222	—
R13	Resistencia	4.7K Ω	Q3	Transistor, 2N2907	—
R14	Resistencia	2K Ω	Q4	Transistor, 2N2907	—
R15	Resistencia	2K Ω	Q5	Mosfet, Buz 80	—
R16	Resistencia	3.3K Ω	Q6	Mosfet, Buz 80	—
R17	Resistencia	3.3K Ω	Q7	Mosfet, Buz 80	—
R18	Resistencia	4.7K Ω	Q8	Transistor, 2N2222	—
R19	Resistencia	30 Ω	Q9	Transistor, 2N2907	—
R20	Resistencia	500 Ω	Q10	Transistor, 2N2222	—
R21	Resistencia	500 Ω	Q11	Transistor, 2N2907	—
R22	Resistencia	4.7K Ω	Q12	Transistor, 2N2222	—
R23	Resistencia	30 Ω	Q13	Transistor, 2N2222	—
Q14	Mosfet, Buz 80	—	U2	74LS06	—

Tabla A.2: Tabla de elementos del Amplificador del PWM

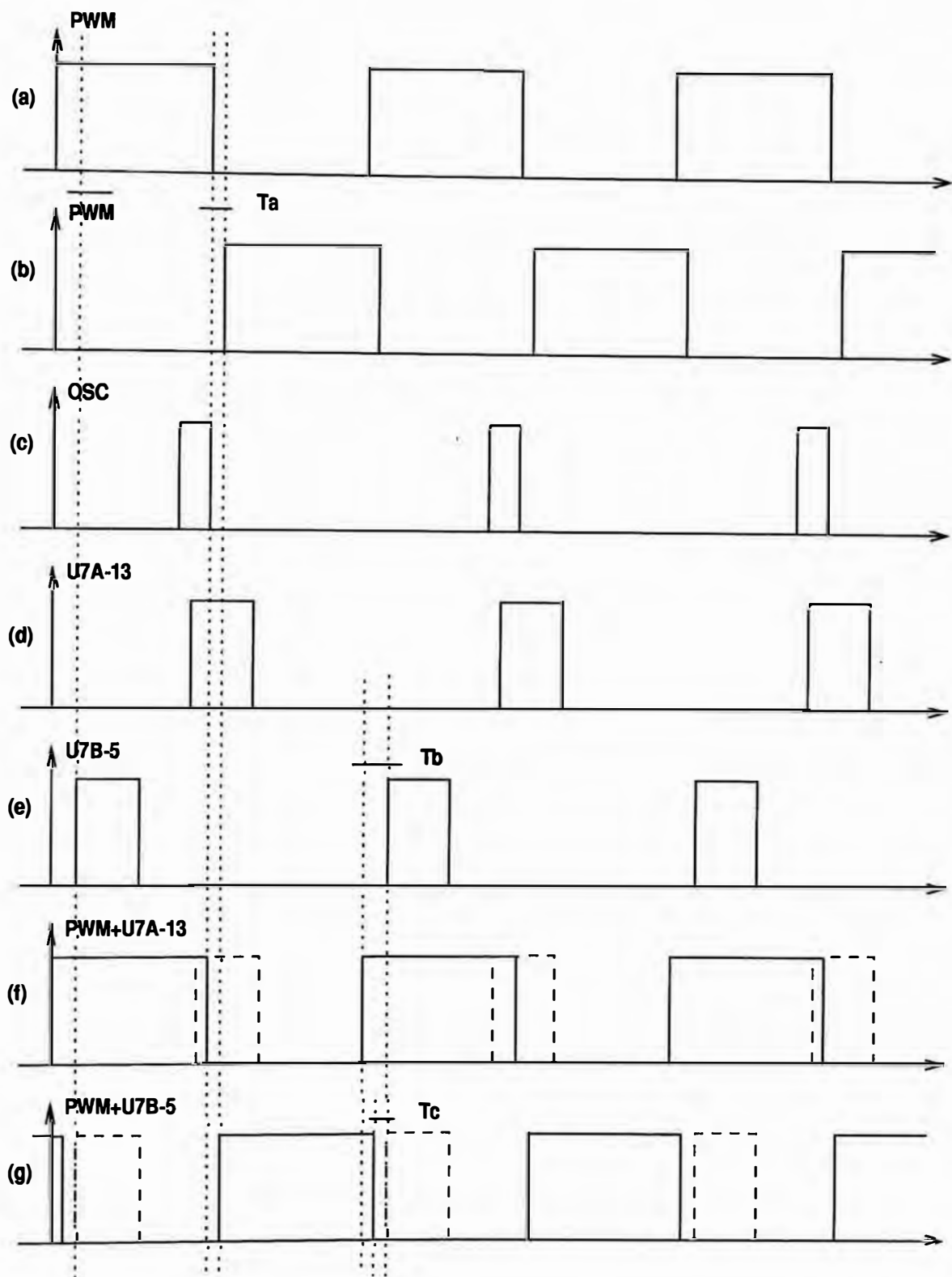
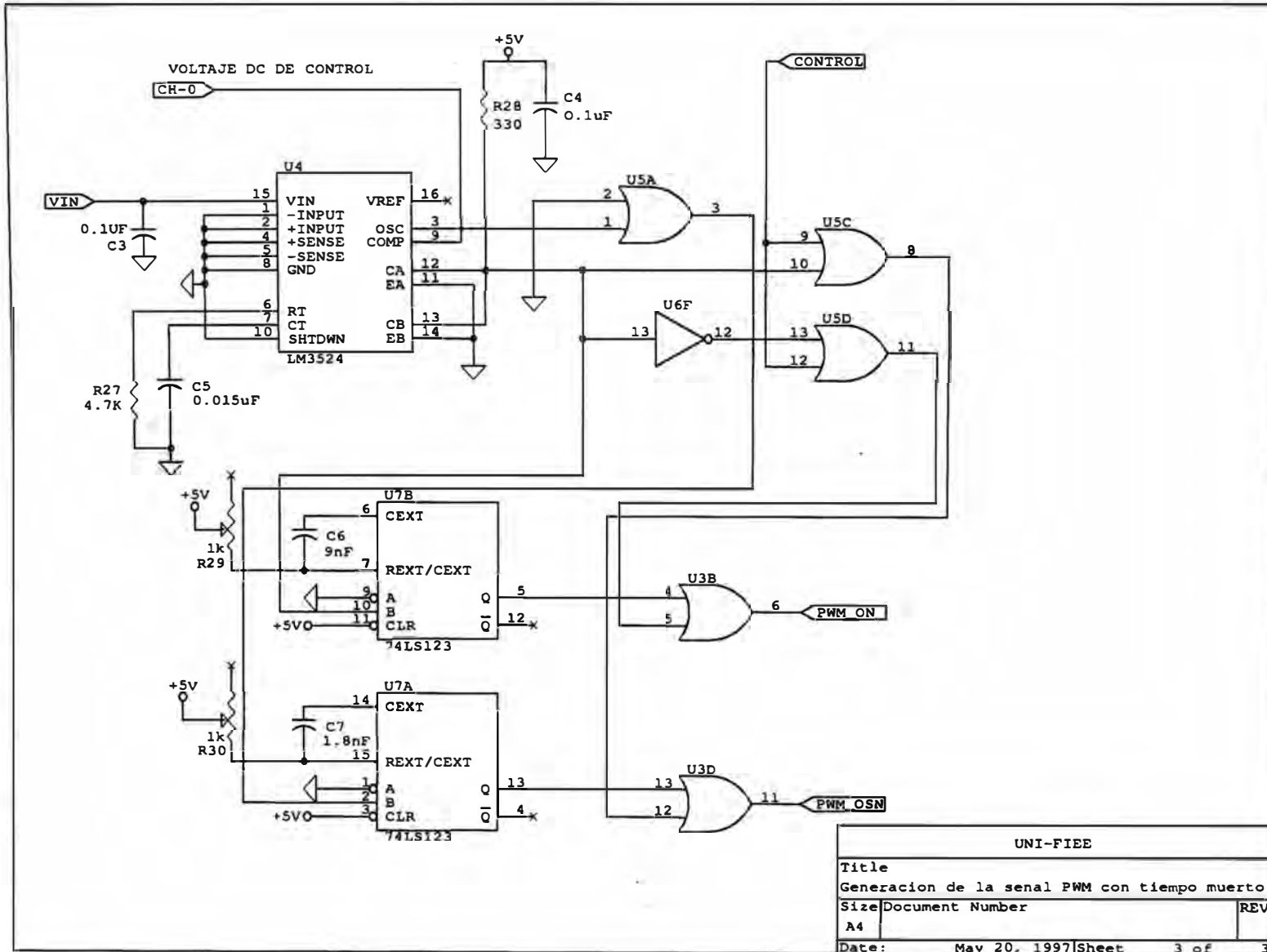


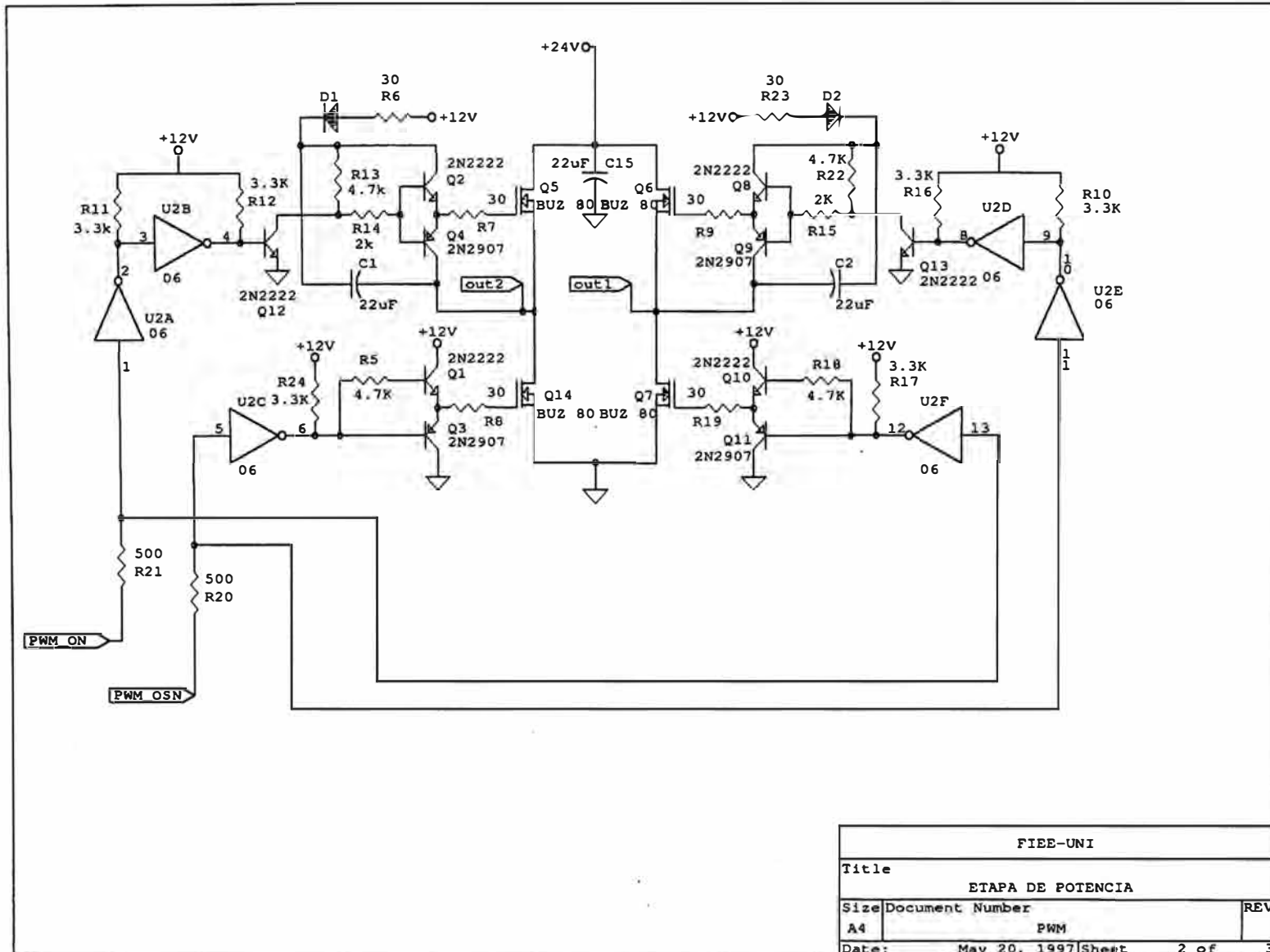
Figura A.1: Diagrama de tiempos del circuito generador de PWM con zona muerta

Figura A.2: Circuito generador de PWM con zona muerta.



UNI-FIEE	
Title	
Generacion de la senal PWM con tiempo muerto	
Size	Document Number
A4	REV
Date:	May 20, 1997 Sheet 3 of 3

Figura A.3: Circuito amplificador PWM



FIEE-UNI		
Title		
ETAPA DE POTENCIA		
Size	Document Number	REV
A4	PWM	
Date:	May 20, 1997	Sheet 2 of 3

BIBLIOGRAFIA

- [1] Katsuhiko Ogata. *Ingenieria de Control Moderna*, Prentice Hall, 1993.
- [2] James A. Freeman y David M. Skapura, *Redes Neuronales. Algoritmos, aplicaciones y técnicas de programación*, Addison Wesley Iberoamericana, S. A. 1993.
- [3] Simon Haykin, *Neural Networks*, Macmillan College Publishing Company. Inc 1994.
- [4] Kumpati S. Narendra and Kannan Parthasarathy, *Identification and Control of Dynamical Systems Using Neural Networks*, IEEE Transactions on Neural Networks. Vol. I. No. 1, March 1990.
- [5] Sanjay I. Mistry and Satish S. Nair *Identification and Control Experiments Using Neural Designs*, IEEE Control Systems. June 1994.
- [6] Octave, *Software para cálculos matemáticos.*, University of Wisconsin-Madison, <ftp://ftp.che.wisc.edu/pub/octave>.
- [7] Astrom, K.J. y Wittenmark B., *Computer Controller Systems: Theory and Desing.*, Prentice Hall, 2da edición, 1990.
- [8] Priyadarshee D. Mathur, *Servo Desing for High-Speed Low-Tension Tape Transport.*, Carnegie Mellon University Pittsburg, Pennsylvania, December 1994.