

72



UNIVERSIDAD NACIONAL AUTONOMA
DE MEXICO

FACULTAD DE INGENIERIA

IMPLEMENTACION DE UN CONTROLADOR PARA
UN SERVOMOTOR DE C.D. BASADO EN REDES
NEURONALES

T E S I S

QUE PARA OBTENER EL TITULO DE
INGENIERO MECANICO ELECTRICISTA
P R E S E N T A :
RENE RAMIREZ CAMARILLO



DIRECTOR: ING. JUAN MANUEL GOMEZ GONZALEZ

MEXICO, D.F.

2000

280974



Universidad Nacional
Autónoma de México

Dirección General de Bibliotecas de la UNAM

Biblioteca Central



UNAM – Dirección General de Bibliotecas
Tesis Digitales
Restricciones de uso

DERECHOS RESERVADOS ©
PROHIBIDA SU REPRODUCCIÓN TOTAL O PARCIAL

Todo el material contenido en esta tesis esta protegido por la Ley Federal del Derecho de Autor (LFDA) de los Estados Unidos Mexicanos (México).

El uso de imágenes, fragmentos de videos, y demás material que sea objeto de protección de los derechos de autor, será exclusivamente para fines educativos e informativos y deberá citar la fuente donde la obtuvo mencionando el autor o autores. Cualquier uso distinto como el lucro, reproducción, edición o modificación, será perseguido y sancionado por el respectivo titular de los Derechos de Autor.

DEDICADA A:

Mis padres:

Catalina Camarillo.
Gabriel Ramírez.

Les dedico la presente tesis como reconocimiento al esfuerzo que juntos hemos realizado para lograr esta meta; por que ha sido de provecho la forma de educarme en casa durante tantos años y por ayudarme a tener un desarrollo físico y mental adecuados; por que me dieron una "mente sana en un cuerpo sano". Gracias por no haberme dado siempre todo lo que les pedí y por haberme hecho mandatos aunque no me agradaran, por ayudarme en momentos de decisiones difíciles; pues ello me ha convertido en un mejor ser humano capaz de luchar por algo y con sentido común. En general por que me han ayudado en el camino, me han guiado y apoyado durante toda mi vida.

Mis hermanos:

Gabriel, Román, Adrián, Araceli.

Por tener siempre alguien con quien aprender, alguien en quien confiar y alguien en quien apoyarse en cualquier momento. Por que han sido siempre una agradable compañía y por que son con quien he compartido muchos grandes momentos especiales.

Lo más importante no es "trabajar", sino producir y disfrutar el fruto de nuestro trabajo.

AGRADECIMIENTOS

A mis padres

Por creer en mi y darme toda su confianza y apoyo; por que con su esfuerzo me han brindado la oportunidad de lograr las metas que me propuse y por la paciencia que me han tenido para alcanzarlas.

A mis hermanos

Por todo el apoyo que siempre me han dado; además de las alegrías y pesares, por su confianza y todos los buenos ratos que vivimos hasta ahora.

Con mucho cariño y respeto.

A mi director de tesis Ing. Juan Manuel Gómez González

Por aceptarme dentro de su grupo de trabajo, tenerme confianza y paciencia para realizar mi tesis y por aceptar dirigirme y apoyarme en esta etapa final de mi carrera.

Al Dr. José Ismael Espinosa Espinosa

Por permitir integrarme al laboratorio de Cibernética de la Facultad de Ciencias y hacer uso de él con toda la confianza del mundo.

Al Ing. Soria Villegas

Por su confianza dada para usar el material necesario y la oportunidad de aprovecharlo al máximo.

Al Ing. Rafael Iriarte encargado del Laboratorio de Control Analógico de la Facultad de Ingeniería

Por las facilidades brindadas para tener un buen material de trabajo y el apoyo dado durante la utilización de él.

A mis compañeros del Laboratorio de Cibernética de la Facultad de Ciencias: Irma, Javier, Eduardo, Tere y Roberto

Por permitirme tener un espacio dentro del laboratorio y la confianza y paciencia que han tenido conmigo.

A mis amigos Agustín, José Luis, Pedro y Eduardo

Por su amistad desde los primeros semestres de la carrera, los momentos de diversión y de apuros que hubo y por su apoyo y ayuda para llevar a cabo esta tesis.

A Nancy

Alguien especial que me ha acompañado y colaborado conmigo durante este tiempo y además por la comprensión y paciencia que me ha brindado.

A la Universidad Nacional Autónoma de México

Por darme una gran formación tanto académica como deportiva y hacer una persona mejor para la sociedad en cuerpo y en espíritu; por ser la mejor universidad y con el mejor material físico y humano.

Y de una manera especial a todas las personas cercanas a mí que en algún momento y sin interés alguno me han dado aliento y apoyo para dar un paso importante en la vida y de las cuales he aprendido algo nuevo cada día; a todos ellos: GRACIAS.

INDICE

CAPITULO I.

Introducción.....	9
-------------------	---

CAPITULO II. Las redes neuronales basadas en el algoritmo de retropropagación

2.1 Las redes neuronales.....	15
La unidad básica de procesamiento: la neurona	17
Retropropagación.....	20
Función de activación o de mapeo.....	25
2.2. Entrenamiento y herramientas usadas.....	27
Conjunto de datos de entrenamiento.....	28
Topología de la red.....	30
2.3 Error.....	32

CAPITULO III. Control de posición

3.1 Control de posición.....	39
Servomecanismos.....	39
Servomotor de c.d.	39
Módulos del servomotor de c.d.	40
3.2 Función de transferencia de la planta.....	42
3.3 Control de posición del servomotor de c.d. con la red neuronal.....	52

CAPITULO IV. Comparación de control lineal y control mediante redes neuronales

4.1 Comparación de control lineal y control mediante redes neuronales.....	57
--	----

CAPITULO V.

5. Discusión..... 74

CAPITULO VI.

6. Conclusiones..... 80

APENDICE A.

Tarjeta de adquisición de datos PCLAB812..... 83

APENDICE B.

Programa Servo 90

BIBLIOGRAFIA..... 94

CAPITULO I.

INTRODUCCION.

1. INTRODUCCION

Los sistemas de control ejercen una poderosa influencia en la vida moderna. Las secadoras, lavadoras automáticas, los hornos de microondas, satélites espaciales, plantas de proceso químico, sistemas de orientación y navegación, control de la contaminación, tránsito colectivo, etc. son unos cuantos ejemplos. Los componentes de los sistemas de control son básicamente dispositivos eléctricos, electrónicos, mecánicos y electromecánicos.

La parte de un sistema que va a ser controlada se denomina planta o proceso. Esta parte es afectada por las entradas y produce una señal de interés particular o salida; figura 1.1.

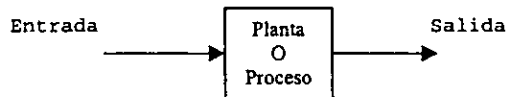


Figura 1.1. Diagrama de bloques de una planta o proceso.

Se puede utilizar un controlador para producir un comportamiento deseado de la planta. El controlador genera las señales de entrada a la planta, las cuales han sido diseñadas para producir las salidas planeadas. Algunas de las entradas a la planta son accesibles al diseñador y otras, por lo general, no lo son. Las señales de entrada no accesibles con frecuencia son perturbaciones externas a la planta. El sistema de la figura 1.2 se denomina de lazo abierto porque las entradas de control no son influenciadas por las salidas de la planta; es decir, no se produce realimentación alrededor de la planta.

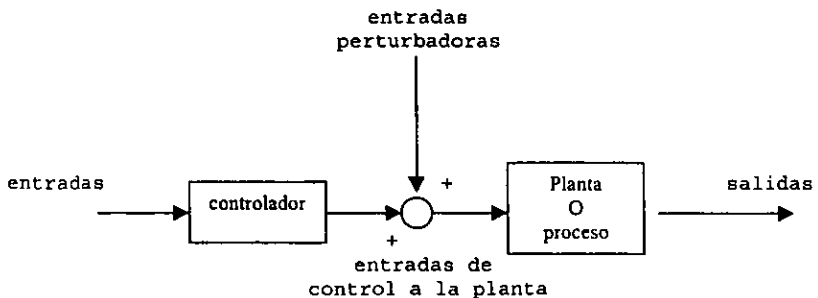


Figura 1.2. Sistema de control de lazo abierto afectado por entradas perturbadoras.

Este sistema de control de lazo abierto posee la ventaja de la simplicidad, pero su funcionamiento es fuertemente dependiente de las propiedades de la planta, las cuales pueden variar con el tiempo. Las perturbaciones en la planta pueden también crear respuestas indeseables que sería conveniente reducir. Las perturbaciones pueden ser totalmente aleatorias de la naturaleza; pero en ocasiones pueden ser predeterminadas con alguna exactitud.

Si las exigencias del sistema no pueden satisfacerse con un sistema de control de lazo abierto, es deseable un lazo cerrado o un sistema de realimentación, como es el caso de la presente tesis. Se dispone entonces de una trayectoria de retorno de la salida al controlador; figura 1.3. Una parte de las salidas del sistema, o todas ellas, se miden y son utilizadas por el controlador. El controlador puede entonces comparar una salida deseada de la planta con la salida real y actuar en consecuencia para reducir la diferencia entre las dos.

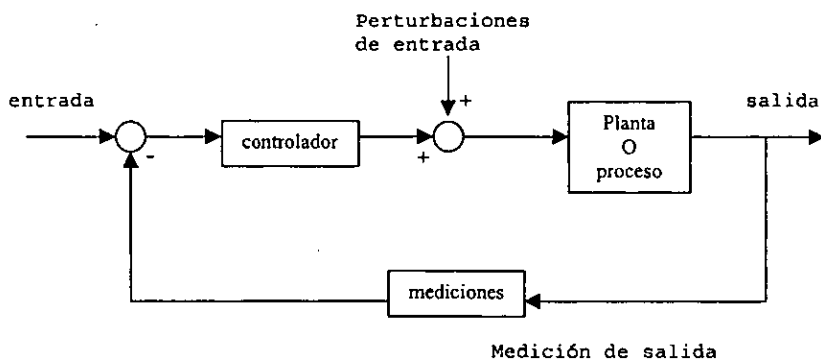


Figura 1.3. Sistema de control con realimentación afectado con entradas perturbadoras.

Algunas de las ventajas que ofrece el control con realimentación son:

1. *Incremento en la exactitud.* El sistema de ciclo cerrado se puede diseñar para llevar a cero el error entre las respuestas deseada y medida.
2. *Pequeña sensibilidad a los cambios en los componentes.* El sistema puede diseñarse para tratar de obtener error cero, a pesar de los cambios en la planta.
3. *Reducidos efectos de las perturbaciones.* Se pueden atenuar notablemente los efectos de las perturbaciones al sistema.

El controlador de la figura 1.3 puede ser un operador humano; el sistema es entonces controlado manualmente. En un sistema de control automático una máquina reemplaza al hombre. Como regla general los sistemas de control automático son para lograr rapidez en la respuesta y precisión.

El trabajo de un ingeniero esta centrado propiamente en el diseño de un controlador que realice un trabajo específico. La complejidad del controlador que finalmente se escoge depende de la complejidad misma de la planta y la exigencia de control que se requiera. Por lo cual los controladores pueden ser desde dispositivos que comparan la entrada del sistema con su salida, como es el caso de los servomecanismos, hasta controladores digitales por computadora en complejos procesos industriales multidimensionales.

Los tipos de controladores más comunes son los controladores lineales continuos, tales como el proporcional (P), integral (I) o derivativo (D) y las combinaciones de estos: proporcional-integral (PI), proporcional-derivativo (PD), proporcional-integral-derivativo (PID), etc.

También se pueden diseñar controladores con el uso de redes neuronales; las cuales pueden funcionar de forma similar a diferentes tipos de controladores. Para la ingeniería de control el estudio e implementación de las redes neuronales ha sido de gran importancia ya que la planta controlada puede tener la respuesta deseada sin tener tantas alteraciones debido a perturbaciones externas. Gran variedad de problemas de control, en los que se han utilizado redes neuronales como controladores de plantas que comúnmente son controladas por controladores lineales, han tenido buen funcionamiento con la implementación de redes neuronales como controlador.

Una red neuronal puede resolver problemas rápida y adecuadamente, en ocasiones de una forma muy precisa. Las computadoras convencionales aproximan las soluciones a problemas desde una orientación digital donde la información debe ser representada por unos o ceros y las respuestas por cierto o falso, o blanco o negro, o también por uno o cero, siendo estas salidas inequívocas, tanto lógicas como matemáticas; es decir sin opción a una salida con un error pequeño. Para lograr tal certeza cuando tratamos con problemas complejos o con información ambigua las computadoras tradicionales deben resolver problemas sofisticados que llevan tiempo en su desarrollo y son más costosos. En ocasiones también se requiere de una computadora poderosa para correr el programa eficientemente y aún así puede o no dar los resultados deseados.

No todas las actividades que pueden ser auxiliadas por las computadoras requieren de una gran precisión; actualmente muchas aplicaciones requieren o provocan aproximaciones o "semejanzas" y tratan con información que es variable, incompleta o "con ruido" donde tal vez una respuesta "gris" es necesaria; las redes neuronales son aplicables para este tipo de actividades.

El desarrollo de esta tesis presenta una aplicación de las redes neuronales en el área de control; utilizando para ello un servomecanismo operando como un sistema de lazo cerrado y donde una red neuronal será el controlador que, implementado en software, sustituye a un controlador lineal usado en el sistema, en este caso un controlador proporcional. La posición del eje del motor del servomecanismo será controlada por una red neuronal implementada en una computadora personal y entrenada para dicho objetivo con el algoritmo de entrenamiento de retropropagación del error.

Las respuestas de la planta tanto con la red neuronal como con un controlador proporcional serán comparadas y se podrá hacer un análisis cualitativo de las respuestas obtenidas. Determinando así si el uso de una red neuronal representa otra opción para llevar a cabo la acción de control en un sistema determinado y hacer su implementación.

CAPITULO II.

LAS REDES NEURONALES BASADAS EN EL ALGORITMO DE RETROPROPAGACION.

LAS REDES NEURONALES BASADAS EN EL ALGORITMO DE RETROPROPAGACION

2.1 LAS REDES NEURONALES

El trabajo en las redes neuronales artificiales comúnmente llamadas "redes neuronales" (RNA) ha sido motivado por el principio de que el cerebro funciona de forma completamente diferente a la forma en que funciona una computadora digital convencional. El trabajo de Ramón Cajál (1911), introdujo la idea de las neuronas como una estructura compuesta del cerebro. Típicamente las neuronas son de cinco a seis veces más lentas que las compuertas de silicio; en un chip pueden ocurrir eventos en el orden de los nanosegundos mientras que en las neuronas ocurren eventos en el rango de milisegundos. Sin embargo, el cerebro compensa esta relativa velocidad de operación de una neurona teniendo un efectivo y sorprendente número de neuronas (células nerviosas) con interconexiones masivas entre ellas; se estima que este número puede ser del orden de 10 billones de neuronas en la corteza humana y 60 trillones de sinapsis o conexiones entre ellas. El resultado de la red es el cerebro como una estructura enormemente eficiente.

El cerebro es una computadora no lineal y altamente compleja (sistema de procesamiento de información en paralelo); tiene la capacidad de organizar las neuronas para ejecutar ciertos cálculos, por ejemplo reconocimiento de patrones, percepción y control motriz, mucho más rápido que las computadoras digitales más rápidas de la actualidad.

Al nacer el cerebro tiene una gran estructura con la capacidad de crear sus propias reglas a través de lo que usualmente se conoce como experiencia; claro la experiencia se construye a través de los años, teniendo el mayor desarrollo del cerebro humano en los primeros dos años después del nacimiento; sin embargo, este desarrollo continúa en cada etapa del ser humano. Durante las primeras etapas de desarrollo alrededor de 1 millón de sinapsis se forman por segundo.

Biológicamente las redes neuronales en el cerebro constan de neuronas individuales como elementos procesadores independientes y las cuales a su vez se conforman de las siguientes partes básicas:

Sinapsis es una estructura elemental y unidad funcional que media las interacciones entre neuronas. El grupo más común es el de la sinapsis química. Una sinapsis convierte una señal eléctrica presináptica a una señal química y luego otra vez a una señal eléctrica postsináptica. En una descripción más tradicional de organizaciones neuronales, se asume que una sinapsis es una simple conexión que puede imponer excitación o inhibición de la neurona receptora; es importante remarcar que no excita o inhibe ambas neuronas, solamente a la receptora.

Los axones, líneas de transmisión y las dendritas, las zonas de recepción, constituyen dos tipos de filamentos de la célula. Un axón tiene una superficie llana, con pocas ramificaciones y de longitud grande; mientras que una dendrita (llamada así por sus ramificaciones) tiene una superficie irregular y muchas ramificaciones.

En otras palabras un axon lleva los impulsos desde la neurona, como una salida. Similarmente las dendritas llevan impulsos hacia la neurona, pueden ser consideradas como entradas. Cada salida de una neurona se ramifica o va a la entrada de muchas otras neuronas; punto en el cual se conectan con las sinapsis de otras neuronas.

La figura 2.1 muestra una neurona en forma piramidal, que es el tipo de célula más común del cerebro.

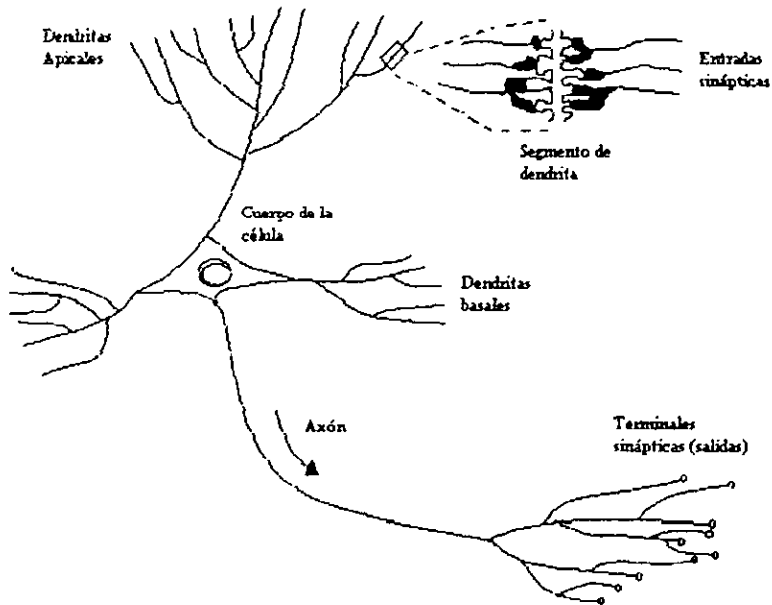


Figura 2.1. La neurona piramidal.

Las sinapsis proveen rutas de intensidades variables para la transmisión de impulsos de una neurona hacia otra, formando la red neuronal. Los impulsos transmitidos a través de la entrada de la neurona son el resultado de salidas de muchas otras neuronas, o de entradas sensadas del mundo exterior.

Como la salida de cada neurona puede interconectarse con muchas otras neuronas al mismo tiempo se alcanza un paralelismo masivo. Entonces la velocidad con la que funciona el cerebro no se puede determinar por la velocidad de procesamiento de cada neurona individualmente sino por el número de conexiones realizadas para resolver un problema.

En una forma muy general una red neuronal artificial (RNA) es una máquina diseñada para modelar la forma en la cual el cerebro ejecuta tareas particulares o funciones de interés; la red es usualmente implementada usando componentes electrónicos o simulada en software en una computadora digital.

Una red neuronal artificial se puede constituir de múltiples capas de neuronas interconectadas con otras neuronas de una misma o de diferentes capas. La topología de conexión de una neurona con otras neuronas puede variar desde una completa interconexión de todas las neuronas de una capa con la posterior; hasta una conexión parcial. Las capas de neuronas están referidas como capas de entrada, capas intermedias u ocultas y capa de salida.

Desde el punto de vista de máquinas adaptables una red neuronal artificial puede definirse como un conjunto de procesadores distribuidos en paralelo que guardan conocimientos de la experiencia y los tienen disponibles para usarlos. Esto es parecido al cerebro en dos aspectos:

1. El conocimiento es adquirido por la red por medio de un proceso de aprendizaje.
2. La fuerza o intensidad de conexión entre neuronas conocida como pesos sinápticos son usados para guardar conocimientos.

Los pesos sinápticos, o simplemente llamados pesos de conexión, en las redes neuronales artificiales representan lo mismo que las sinapsis en las redes neuronales del cerebro. Pero estos pesos de interconexión son procesados por una función suma (sumatoria de pesos) y este resultado es mapeado por una función de activación, comúnmente no lineal. La salida de esta función de activación es la salida de la neurona; generalmente cuando la función de activación es no lineal es una función sigmoide. Las conexiones son las uniones entre una o más neuronas en las redes neuronales artificiales e indican la relación existente entre ambas; son análogas a lo que son los axones y las dendritas en las redes neuronales del cerebro.

La unidad básica de procesamiento: la neurona

La unidad básica de procesamiento, también llamada elemento procesador (EP), en una red neuronal artificial es la neurona; cuyo modelo se muestra en la figura 2.2. En los modelos de las redes

neuronales artificiales éstos elementos realizan las funciones de evaluación de las señales de sus entradas y la suma de ellas.

Las entradas provienen de otras neuronas o del mundo exterior, y son primero multiplicadas por un peso de conexión almacenado (también llamado simplemente peso). La mayoría de los modelos de las redes neuronales tienen una entrada extra denominada bias con la cual se alcanza una convergencia más rápida de la red aunque su uso dependerá de la aplicación. Las entradas multiplicadas son sumadas con el peso de bias formando una pre-salida. Una neurona de bias puede ser visualizada como una entrada con un valor constante de +1.0, asociada con un peso de conexión llamado peso de bias. Durante el entrenamiento, el algoritmo de entrenamiento ajusta el peso de bias igual que como lo hace con los otros pesos de conexión. Así el peso de bias, al igual que los demás pesos de conexión, puede tomar un rango de valores positivos o negativos. Finalmente esta pre-salida es mapeada a través de la función de activación para obtener la salida total de la neurona. La función de mapeo, típicamente no lineal, modifica la sumatoria de la neurona antes de pasar la señal hacia las otras neuronas.

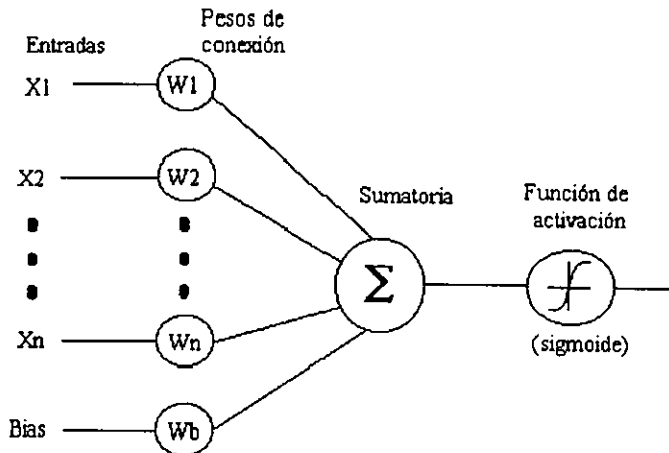


Figura 2.2. Modelo de la unidad básica de una RNA: neurona o elemento procesador (EP).

En forma matemática:

$$Pre_salida = (\sum pesos \times entrada) + valor_bias$$

Las neuronas artificiales son organizadas en capas (también referidas como módulos) con conexiones bien conocidas. El arreglo de entrada, consiste de las entradas a la red neuronal y suministra las entradas a la primer capa. Cada neurona es conectada a cada una de las neuronas de la siguiente capa y esta capa puede conectarse a capas subsecuentes y así sucesivamente hasta llegar a la capa final.

En la figura 2.3 se muestra una red neuronal de dos capas y con seis neuronas, cuatro en la primer capa y dos en la segunda capa. Para este caso el arreglo de entrada conecta a cada una de las neuronas de entrada con las de la capa oculta (las capas intermedias son referidas como capas ocultas por que sus salidas son internas de la red). Cada una de las neuronas en la capa oculta se conecta a cada una de las neuronas en la capa final de la red. La capa final de una red es llamada capa de salida.

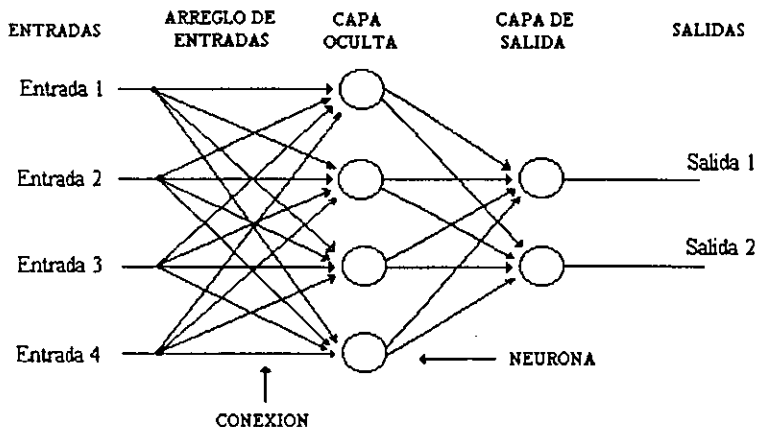


Figura 2.3. Red neuronal de dos capas (feed-forward).

El procedimiento para llevar a cabo el proceso de aprendizaje se llama algoritmo de aprendizaje; en el cual se modifican los pesos sinápticos de la red ordenadamente a manera de alcanzar un objetivo de diseño deseado. La modificación de pesos sinápticos es el método tradicional para el diseño de redes neuronales.

Retropropagación

Las redes neuronales artificiales también deben "aprender" como procesar la información de entrada antes de que esta pueda ser utilizada en una aplicación. El proceso de entrenamiento de una red neuronal involucra el ajuste de los pesos de entrada en cada neurona hasta que la salida de la red se aproxima a la salida deseada. Una vez que la red ha sido entrenada ésta "recuerda" la respuesta que "aprendió" durante el entrenamiento para cada patrón de entrada que se le presente.

Una red neuronal artificial que se ha encontrado útil en problemas que requieren de reconocimiento de patrones y ejecutar funciones de mapeo no triviales es la red neuronal entrenada con el algoritmo de retropropagación. La arquitectura general de este tipo de red se ilustra en la figura 2.4 y puede ser diseñada para operar con varias capas, propagación hacia delante (feed-forward) y usando modo de aprendizaje supervisado.

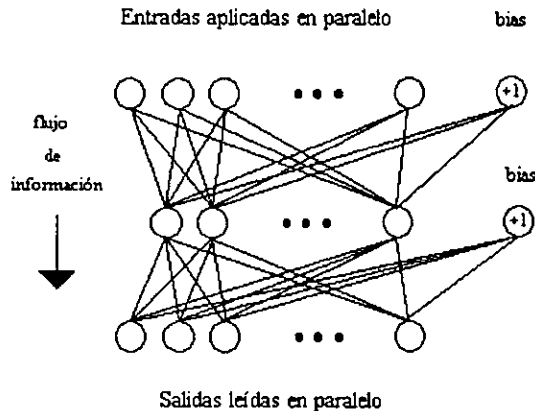


Figura 2.4. Arquitectura general de una RNA de retropropagación.

Como la salida de cada una de las neuronas alimenta a las neuronas de las capas subsiguientes se dice que la red tiene una topología de propagación hacia delante (feed-forward). Existe otro tipo de topología llamada topología recurrente, donde la salida de una neurona se realimenta así misma y realimenta a otras neuronas en su misma o en previas capas. El aprendizaje es considerado como un porcentaje en el cambio de los pesos o memoria de la red. El tipo de aprendizaje se debe a que el diseñador, según la arquitectura de la red empleada, puede considerar un proceso que obligue a la red a entregar una respuesta dada ante una entrada en específico. Una respuesta particular puede o no ser

especificada. El aprendizaje supervisado es empleado en redes en las cuales los patrones de entrenamiento están compuestos de dos partes fundamentales, un vector de entrada y un vector de salida, asociando la entrada con su correspondiente salida en cada neurona o elemento procesador, al cual se le denomina nodo. Cada tipo de topología es aplicable para solucionar cierto tipo de problemas. Para la presente tesis la red es entrenada con el algoritmo de retropropagación de error y se diseña para operar con propagación hacia delante y con aprendizaje supervisado.

Cuando se intenta automatizar aplicaciones complejas de reconocimiento de patrones a menudo aparecen problemas; principalmente de tiempo-hombre y tiempo-máquina para resolverlos. Por ejemplo; consideremos un programa de computadora que pueda trasladar una matriz de números binarios de 5x7, que representa la imagen de un carácter alfanumérico en mapa de bits, a su equivalente código ASCII de ocho bits. Este problema básico, ilustrado en la figura 2.5, donde se representa una letra A en la matriz de 5x7, parece ser relativamente trivial a primera vista; pero no es obvia la función matemática que realice este traslado deseado.

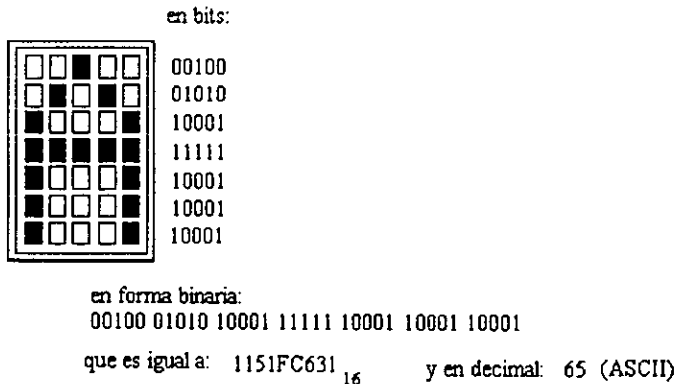


Figura 2.5. Imagen de un carácter mapeada a su correspondiente código ASCII.

Encontrar la función que realice la correlación pixel por pixel de la imagen sería muy tardado; por lo que una mejor solución sería usar una tabla de comparación. La tabla tendría primero el equivalente numérico del código de patrón de bits, generado moviendo los siete renglones de la matriz a un solo renglón considerando el resultado como un número binario de 35 bits, y después estaría el código ASCII asociado con el carácter. La tabla contendría exactamente el mismo número de pares ordenados como caracteres a convertir.

Aunque este método parece bastante rápido y sencillo existen muchas situaciones que ocurren en sistemas reales que no podrían manejarse con este método. Por ejemplo, si el scanner altera un pixel aleatorio; en la matriz de imagen de entrada habría ruido cuando fuera leída. Este simple error en un pixel causaría en el algoritmo de comparación regresar un código ASCII nulo o erróneo, por que el patrón de entrada y el patrón de salida deben ser exactos.

Ahora consideremos la cantidad de software adicional (y por lo tanto tiempo-máquina) que debemos agregar al algoritmo de tabla de comparación para mejorar la capacidad de reconocer a que carácter corresponde la imagen con ruido. Errores de un solo bit pueden ser encontrados y corregidos, pero la dificultad se incrementa conforme el número de errores de bits aumenta. Para complicar aún más el problema como podría el software reconocer si el ruido ocurrió para hacer una "Q" que parece una "O" o una "E" que parece una "F"? podría pasar un tiempo-máquina excesivo tratando de eliminar el ruido del patrón de entrada para intentar trasladarlo a su código ASCII.

Una solución a este problema es tomar ventaja del uso de las redes neuronales y reducir el tiempo requerido por un procesador secuencial para efectuar el mapeo; así se tiene una forma más óptima y automatizada de solucionar el problema de reconocimiento de patrones.

Al igual que en el ejemplo anterior existen muchas otras aplicaciones que resultan difíciles de realizar por la dificultad de llegar a la solución mediante procesos secuenciales; generalmente en las que hay que realizar traducciones de datos complejos sin una función de correspondencia definida para el proceso de traducción o en las que deben aproximar su salida datos de entrada ruidosos. Las redes neuronales se usan debido a su capacidad de adaptarse, aprender, generalizar y organizar datos cuyas operaciones están basadas en procesos paralelos. El algoritmo de retropropagación de error es útil para éstos problemas y para los que requieran el reconocimiento de tramas complejas y realización de funciones de correspondencia no triviales; así como para arquitecturas de redes multicapas, con propagación hacia adelante y con aprendizaje supervisado.

En una red entrenada con el algoritmo de retropropagación, durante la etapa de entrenamiento los pesos de la red convergen en un punto en el espacio de pesos de la red, en el que el problema es conocido por la red; cuando la red ha alcanzado este punto puede dar la salida correcta para cada entrada dada. Debido a la naturaleza de la red no siempre se puede aprender el problema exactamente; la salida producida por la red al presentar una entrada puede ser una aproximación de la salida deseada; por lo cual no es fácil decir cuando la red ha aprendido un problema.

Como se mencionó anteriormente, figura 2.2, a una red de retropropagación, (refiriéndonos a una red entrenada con el algoritmo de retropropagación), a cada conexión directa entre cada uno de los elementos procesadores se le puede asociar un peso y estos pesos pueden ser actualizados durante el entrenamiento con la ayuda de una regla de

aprendizaje; la regla de aprendizaje son los vectores de entradas y su correspondiente vector de salida.

Este tipo de redes están organizadas en "n" capas con conexiones directas entre los elementos procesadores de cada capa; con un mínimo de tres capas; capa de entrada, capa intermedia u oculta y capa de salida.

Para describir el funcionamiento de una red de retropropagación se toma como referencia la figura 2.6. Con la arquitectura general de una red de retropropagación.

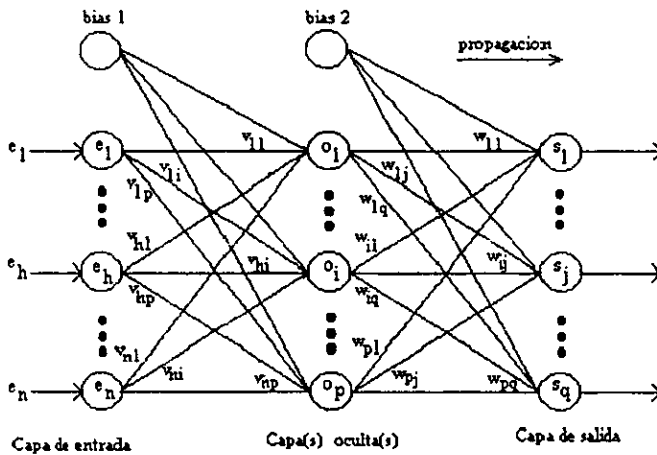


Figura 2.6. Arquitectura general de una red de retropropagación.

Donde:

- e_1 a e_n representan a las neuronas de la capa de entrada (señales de entrada).
- v_{11} a v_{np} representan los pesos de conexión entre las neuronas de la capa de entrada y las de la primera capa oculta.
- o_1 a o_p son las neuronas de la capa oculta.
- w_{11} a w_{pq} representan los pesos de conexión entre las neuronas de la capa oculta y la capa de salida.
- s_1 a s_q neuronas de la capa de salida (señal de salida).

Es importante recordar que la neurona de bias se toma en cuenta para un entrenamiento más ágil y lograr además una convergencia de la red más rápida. Estas neuronas de bias no reciben entrada alguna, por que por definición las neuronas de bias toman un valor constante de +1; siendo entonces un factor en el resultado final del entrenamiento haciendo que los pasos hacia el mínimo total a alcanzar sean mayores; o sea lograr una convergencia más rápida.

Las señales de entrada son presentadas por los patrones (en paralelo) e_1 a e_n a cada una de las neuronas de la capa de entrada y transmitidas a las neuronas de la capa siguiente (la primera de las capas ocultas). En cada una de las neuronas de la capa intermedia, O_1 a O_n , se efectúa la multiplicación de sus entradas por los pesos, V_{11} a V_{np} , correspondientes a las interconexiones de las neuronas e_1 a e_n con las neuronas O_1 a O_p , para posteriormente efectuar la sumatoria de los productos:

$$O_p = (bias1)(V_{bias}) + \sum_{x=1}^n V_{xp} e_x$$

Una vez realizada la sumatoria, el resultado es valuado por la función de activación o de mapeo, en el caso de esta tesis es una función sigmoide, mostrada como:

$$Sigmoide_p = f(O_p)$$

El valor obtenido es la entrada aplicada a cada uno de los elementos de la siguiente capa, en la figura 2.6 las neuronas S_1 a S_q , en las cuales nuevamente se efectúa la sumatoria del producto de éstas entradas por los pesos correspondientes (W_{11} a W_{pq}):

$$S_q = (bias2)(W_{bias}) + \sum_{y=1}^p W_{yq} f(O_p)$$

Este resultado vuelve a ser valuado por la función de activación, entonces:

$$Sigmoide_q = f(S_q)$$

donde $Sigmoide_{eq}$ es la salida de la red para la neurona S_q .

Cuando se entrena la red con un aprendizaje supervisado, el conocimiento de la salida por cada elemento procesador es suficiente para calcular el error total producido por la red.

La función de activación o de mapeo

De acuerdo a la figura 2.2 para obtener la salida definitiva de cada neurona, la pre-salida es mapeada en una función para así crear el valor de activación para cada neurona subsecuente. Esta función de activación mapea, escala o limita el dominio de las neuronas en su entrada a un rango específico a la salida del mismo. Algunas funciones de activación más comunes son:

- a) Función lineal
- b) Función rampa
- c) Función escalón
- d) Función sigmoide

La función de activación utilizada para entrenar la red neuronal de ésta tesis es la función sigmoide; cuya gráfica es la de la figura 2.7.

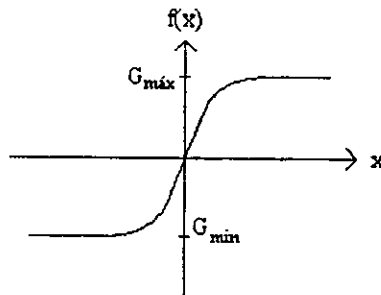


Figura 2.7. Gráfica de la función sigmoide mostrando ganancias mínima y máxima.

Esta función de activación acotada, decreciente y con forma de S, provee una respuesta no lineal, (siendo esto una ventaja para la salida de la red neuronal); esta función no lineal es determinada por la ecuación de $f(x)$.

Consideremos:

$$O(x) = \frac{1}{1 + e^{-Gx}}$$

entonces:

$$f(x) = \text{Sigmoide}(x) = \text{min} + (\text{máx} - \text{min}) O(x)$$

donde:

G - es la ganancia de la función de activación.
min - es el valor mínimo de la función de activación.
máx - es el máximo de la función de activación.
e - es la base del logaritmo natural.
x - es la suma de los productos.

En general:

$$\frac{df(x)}{dx} = (\text{máx} - \text{min}) O(x) [1 - O(x)]$$

Una función sigmoide escala el primer valor de salida de cada neurona antes de que la señal sea enviada a la siguiente capa de neuronas. Una primer salida con un valor grande y negativo será escalado al valor mínimo (min) de la función de activación; igualmente, un valor grande positivo es escalado al valor máximo (máx) de la función de activación. La ganancia de la función de activación es el rango en el cual la función se incrementa desde su mínimo hasta su máximo.

Como la función de activación tiene la labor de convertir el valor de cada neurona en su correspondiente valor de salida, en algunas de las redes se usa un valor de umbral para la determinación de la salida de la neurona; esto es por que el valor de umbral determina cuando la neurona genera una señal activa o inactiva. Esto es:

Salida = f(sum) si sum > χ

Salida = 0 para cualquier otro valor

Donde χ es el valor de umbral correspondiente a cada neurona. Entonces, la neurona puede generar la señal de salida si la suma de las entradas multiplicadas por su pesos correspondientes es mayor que el umbral, de lo contrario la neurona no genera ninguna señal o genera una señal inhibidora.

2.2 ENTRENAMIENTO Y HERRAMIENTAS USADAS.

Como se mencionó anteriormente una red neuronal necesita primero "aprender" las respuestas que debe dar para cada entrada específica que se le presente; es decir requiere de un periodo de entrenamiento mediante el cual pueda posteriormente generar las salidas deseadas (aprendidas en su entrenamiento) dependiendo de la aplicación que se trate.

Para poder llevar a cabo el entrenamiento de una red neuronal primero se deben de tener en cuenta dos puntos básicos:

- Preparar el conjunto de datos de entrenamiento. Reunir, seleccionar y preprocesar los datos; es decir, tener en consideración con que tipo de datos se cuenta para el entrenamiento, como se van a obtener estos datos y de que manera van a servir a la red, para posteriormente llevar acabo el entrenamiento.
- Crear la red propia. Determinar la topología de la red a utilizar; incluyendo el número de neuronas, el tipo y la arquitectura de la red. Es importante tener en cuenta en que aplicación se va a usar la red y el conjunto de datos con los que se cuenta para entrenarla.

Una vez definidos los puntos anteriores se procede a entrenar la red neuronal, probarla y una vez que funcione satisfactoriamente usarla en la aplicación correspondiente.

Para esta aplicación en particular las dos primeras herramientas más importantes utilizadas son los paquetes de simulación Simnon y Dynamind. Con Simnon, programa de simulación de sistemas, se obtienen los primeros datos para entrenar la red; con Dynamind, software simulador de redes neuronales, se lleva a cabo el entrenamiento.

Conjunto de datos de entrenamiento.

Para llevar a cabo el entrenamiento de la red se deben reunir los datos con los cuales la red "aprenderá" durante el entrenamiento. Estos datos deben tener un formato adecuado dependiendo del programa con el cual se vaya a hacer el entrenamiento. Se deben construir los archivos para ser introducidos al programa; para este caso el simulador de redes neuronales Dynamind.

El programa simulador de sistemas Simnon es la herramienta utilizada para obtener el conjunto inicial de datos ya que la planta puede ser simulada en Simnon de acuerdo a su diagrama de bloques y a su modelo matemático, lo cual se explica con mayor detalle en el capítulo siguiente, y así obtener archivos con valores de entrada y de salida del controlador. Los archivos contienen datos del controlador en la siguiente forma:

Archivo de entrada		Archivo de salida	
0.00	1.00	0.00	1.4303
0.03	0.99	0.03	1.4160
0.06	0.96	0.06	1.3758
0.12	0.86	0.12	1.2363
0.18	0.72	0.18	1.0385
.	.	.	.
.	.	.	.
.	.	.	.
.	.	.	.

Donde para cada archivo la primer columna corresponde al tiempo, en segundos, y la segunda columna corresponde al valor de la señal en estudio en el mismo tiempo; para obtener estos dos archivos de Simnon se realizan dos simulaciones del sistema; ya que Simnon solo guarda datos en un archivo de un punto determinado del sistema por cada simulación llevada a cabo. Los archivos contienen valores de la señal para el mismo intervalo de tiempo y va de cero segundos hasta el tiempo en el cual la planta esta completamente en estado estable.

De estos dos archivos obtenidos del simulador Simnon se construyen los patrones de entrenamiento para entrenar la red con Dynamind.

Un patrón de entrada consiste en pares de datos de entrada y su correspondiente salida:

entrada 1	...	entrada n	salida 1	...	salida n
-----------	-----	-----------	----------	-----	----------

con tantas entradas y salidas como vaya a tener la red.

La construcción de los patrones de entrenamiento de la red se forman de acuerdo a la forma en que serán introducidos a la red neuronal. La figura 2.8 explica gráficamente como se acomodan los datos para formar dichos patrones para una red con cuatro neuronas de entrada y dos neuronas de salida.

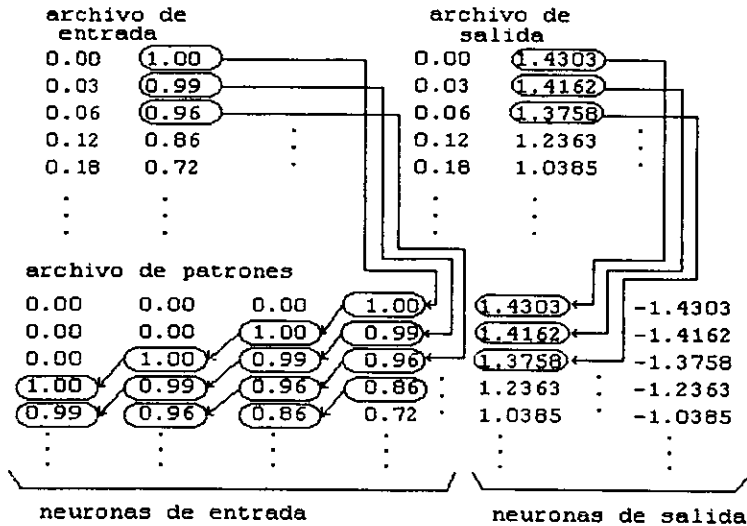


Figura 2.8. Patrones de entrenamiento.

Cada línea de datos es un patrón de entrenamiento. Donde el primer patrón consta de ceros, el primer valor de entrada y sus salidas correspondientes; para los siguientes patrones los datos van teniendo un corrimiento de un lugar para formar así un nuevo patrón con el siguiente valor de entrada y sus salidas correspondientes y así consecutivamente hasta terminar de formar los patrones de entrenamiento para la red; por ahora solo es importante ver el tipo de formato del archivo de patrones que requiere Dynamind.

Para realizar el entrenamiento en el programa Dynamind ya debe existir el archivo de patrones, con el número de patrones necesario dependiendo de cada aplicación de la red, y saber la topología de la red a formar.

Topología de la red.

Definir la topología de la red también es importante antes de comenzar a entrenar una red con Dynamind; pues es la que define cuantos datos de entrada y cuantos de salida se deben de poner en los pares que forman los patrones de entrenamiento. Para la figura 2.8 se tiene una red con cuatro neuronas de entrada, tres neuronas en la capa oculta y dos neuronas de salida con valores absolutos iguales pero signo contrario, esto no tiene algún fin determinado pero sirve para entrenar la red, ya que la neurona con valores positivos es la que nos interesa más y por ende la que se utilizará en la implementación.

La red que se construyó para estos patrones iniciales de entrenamiento es la de la figura 2.9.

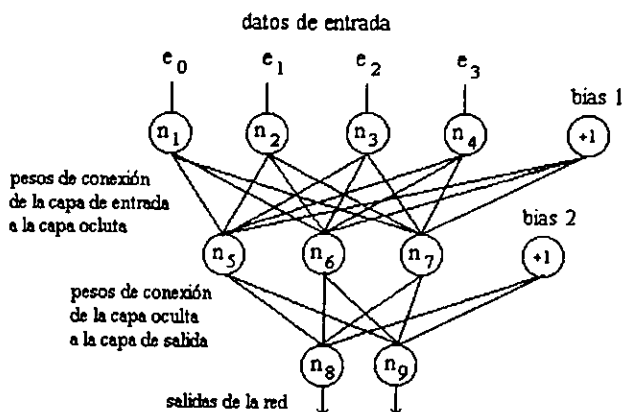


Figura 2.9. Topología de la RNA.

Una vez definida la topología de la red a entrenar y con el archivo de patrones construido se puede comenzar el entrenamiento en el simulador de redes neuronales Dynamind.

Dynamind necesita que ya este construido un archivo de datos ordenados (archivo con extensión *.dat) con el cual el programa iobuild de Dynamind construye otro archivo de patrones (archivo con extensión *.io) con el que directamente es realizado el entrenamiento de la red. Una vez que se esta dentro de Dynamind se le indica la topología de la red a entrenar (forma el archivo con extensión *.net) y en el cual se guardan los datos y resultados del entrenamiento de la red; tales como: pesos de conexión entre las neuronas, número de neuronas en cada capa,

datos de error, valores máximos y mínimos de la función sigmoide, etc. Este archivo con extensión *.net es el más importante que entrega Dynamind, ya que es el que contiene los pesos de conexión, los cuales servirán para implementar la red neuronal vía software en la computadora.

2.3 ERROR

La red de la figura 2.9 se entrenó con un total de 52 patrones debido a que con ese número de datos el sistema simulado esta ya en su estado estable, y se realizaron diferentes entrenamientos con diferentes duraciones; esto es, con diferentes número de veces que Dynamind muestra todos los patrones a la red, para disminuir el error de aprendizaje; cada vez que se le muestran todos y cada uno de los patrones a la red durante su entrenamiento se cuenta como una época. El número adecuado de épocas para entrenar dicha red fue de 20 000, ya que aunque se aumentara el número de épocas del entrenamiento la red experimentaba prácticamente el mismo error de aprendizaje; por el contrario al disminuir el número de épocas el error de aprendizaje aumentaba. Con 20 000 se tuvo un error por época en la red de $1.81e^{-05}$ y al aumentar a 30 000 épocas el entrenamiento de la red se obtuvo el mismo valor de error por época; además al analizar el archivo de pesos que entrega Dynamind tanto para 20 000 como para 30 000 se tenían pesos de conexión que prácticamente eran iguales, en valores con cuatro decimales solo había cambio en diezmilésimas o cuando mucho en milésimas.

Con respecto al valor máximo y mínimo de la función sigmoide se fijó en +1.5 y -1.5 puesto que el valor del dato de salida más alto en los patrones es de 1.4303.

Para encontrar un número adecuado de neuronas en la capa oculta de una red lo que se hizo fue construir tres redes con diferentes números de neuronas en la capa oculta; pues hasta ahora es la única que se puede variar en este aspecto. La primera con dos neuronas, la segunda con tres neuronas y la tercera con cuatro neuronas en la capa oculta; se entrenaron estas redes con exactamente los mismos patrones y los mismos parámetros de entrenamiento: 20 000 épocas y valores de la función sigmoide entre 1.5 y -1.5.

Las siguientes gráficas indican el grado de aprendizaje de las redes mencionadas en el párrafo anterior; muestran el error contenido en cada red (dos, tres y cuatro neuronas en su capa oculta) en relación con la salida mostrada en los patrones de entrenamiento. Figuras 2.10.A a 2.10.D.

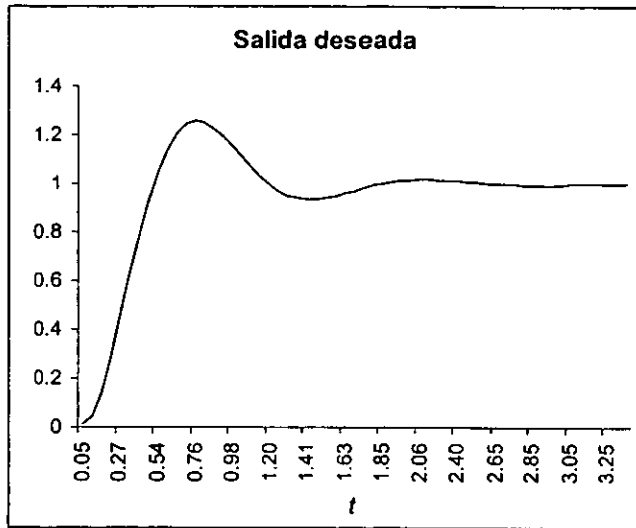


Figura 2.10.A. Curva formada por las salidas de los patrones de entrenamiento y a la que se busca ajustar las salidas de las RNA en entrenamiento.

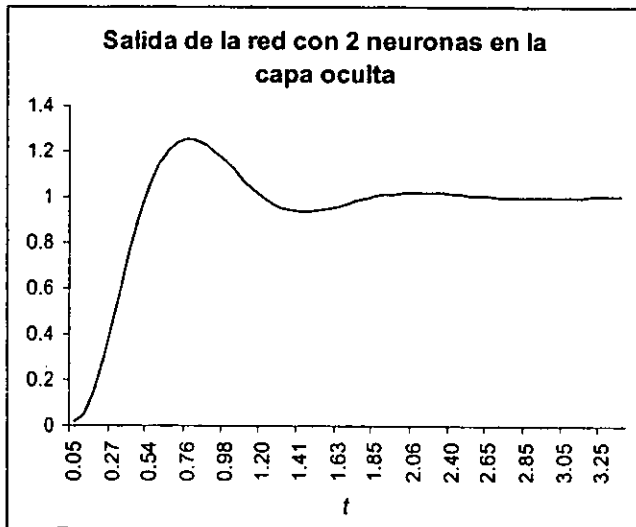


Figura 2.10.B. Curvas donde se muestran las salidas obtenidas de la primer RNA entrenada y mostrándole los mismos patrones de entrenamiento; la diferencia con la anterior es mínima.

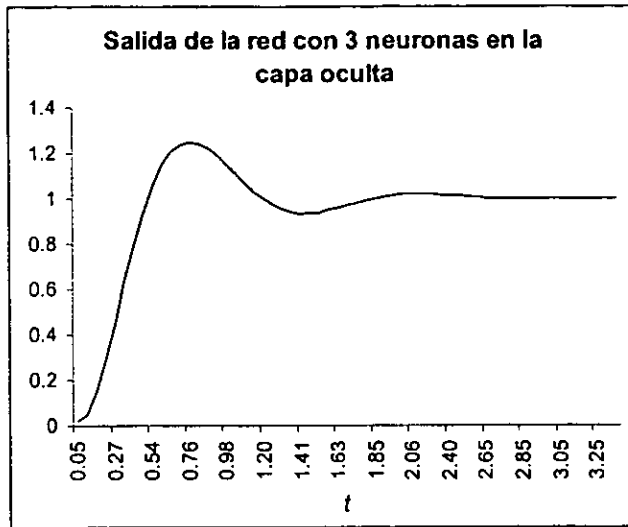


Figura 2.10.C. Salidas obtenidas con la segunda RNA entrenada y los mismos patrones de entrenamiento; obsérvese que el sobrepaso y tiempo de asentamiento son prácticamente iguales a los de la gráfica de patrones de entrenamiento.

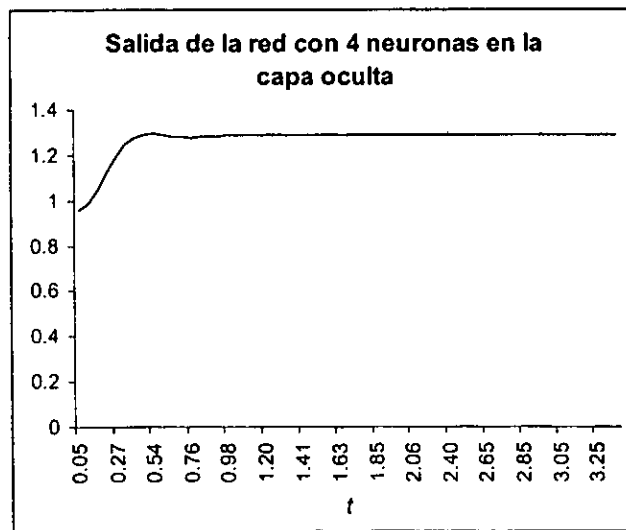


Figura 2.10.D. Salidas de la red con 4 neuronas en la capa oculta. Donde de acuerdo a las gráficas anteriores sus salidas tienen una diferencia

bastante considerable al presentarle los mismos patrones de entrenamiento. Este error mayor, con relación a las RNA anteriores, indica que esta red no tuvo un buen "aprendizaje" durante su entrenamiento y puede dejar de tomarse en cuenta.

Al ver las salidas de las figuras 2.10 tenemos que las redes con dos y tres neuronas en la capa oculta son muy similares; sin embargo se tiene que ver cual de las dos es mejor en cuanto al error en su aprendizaje; por lo cual se analizan los errores numéricos dentro de Dynamind y se tienen los siguientes datos de error para cada una de ellas.

No. de neuronas en capa oculta	error por época	error prom. por unidad	error cuadrático por unidad
2	$2.08e^{-05}$	$3e^{-05}$ a $1e^{-05}$	$7e^{-15}$
3	$1.81e^{-05}$	$2e^{-05}$	$7e^{-15}$

Donde se puede ver que en la red con tres neuronas en la capa intermedia existe un error menor y también existe menos variación en el error promedio por unidad de procesamiento o neurona. Por lo cual tenemos que la red con tres neuronas intermedias tiene un mejor aprendizaje. Recordemos que según el aprendizaje de la red neuronal el error irá o no disminuyendo y ésta podrá generalizar o responder mejor a entradas no mostradas en los patrones de entrenamiento que se le puedan presentar.

A continuación se presenta un fragmento de un archivo formado por los patrones de entrenamiento y las salidas entregadas por cada una de las tres redes hasta ahora mencionadas:

Patrones de entrada

salidas de las RNA
obtenidas con 2,
3 y 4 neuronas en
la capa oculta

entradas				salida			
0.000000	0.000000	0.000000	0.000000	@0.000000>	0.007943	0.008229	0.946
0.000000	0.000000	0.000000	0.008030	@0.011485>	0.018699	0.018778	0.957
0.000000	0.000000	0.008030	0.030476	@0.043590>	0.051260	0.051462	0.981
0.000000	0.008030	0.030476	0.101255	@0.144824>	0.150759	0.149795	1.047
0.008030	0.030476	0.101255	0.199588	@0.285547>	0.293295	0.294607	1.128
0.030476	0.101255	0.199588	0.313123	@0.447860>	0.457464	0.455826	1.196
0.101255	0.199588	0.313123	0.430911	@0.616332>	0.619838	0.618912	1.245
0.199588	0.313123	0.430911	0.543892	@0.777929>	0.778255	0.778091	1.275
0.313123	0.430911	0.543892	0.645150	@0.922757>	0.926216	0.925292	1.285
0.430911	0.543892	0.645150	0.729964	@1.044070>	1.051988	1.049619	1.295
.
.
.
0.697283	0.696605	0.696239	0.696150	@0.995704>	0.999943	0.997769	1.295
0.696605	0.696239	0.696150	0.696288	@0.995901>	1.000153	0.997983	1.295
0.696239	0.696150	0.696288	0.696598	@0.996345>	1.000618	0.998452	1.295
0.696150	0.696288	0.696598	0.697025	@0.996955>	1.001255	0.999094	1.295
0.696288	0.696598	0.697025	0.697515	@0.997655>	1.001984	0.999828	1.295
0.696598	0.697025	0.697515	0.698021	@0.998379>	1.002737	1.000583	1.295
0.697025	0.697515	0.698021	0.698505	@0.999072>	1.003455	1.001305	1.295
0.697515	0.698021	0.698505	0.698938	@0.999691>	1.004097	1.001948	1.295
0.698021	0.698505	0.698938	0.699300	@1.000210>	1.004632	1.002486	1.295
0.698505	0.698938	0.699300	0.699579	@1.000610>	1.005046	1.002899	1.295

Donde se puede ilustrar mejor que la red con tres neuronas en la capa oculta es la que tiene un menor error en sus salidas y por lo tanto un mejor aprendizaje.

CAPITULO III.

CONTROL DE POSICION.

posee dos ejes, uno llamado eje rápido que gira a una determinada velocidad en función del voltaje aplicado y un segundo llamado eje lento que gira en una proporción de 1:30 con respecto al eje rápido; o sea que el eje lento da una vuelta por cada 30 del eje rápido.

El sistema también requiere de un módulo llamado servoamplificador (SA); el cual controla directamente al motor y esta configurado para tener un control de armadura, con el que se logra un comportamiento más estable del sistema; (el servoamplificador puede ser configurado también con control de campo, pero así es más enfocado hacia el control de velocidad); el diagrama esquemático del servoamplificador conectado directamente al motor con control de armadura es el de la figura 3.2.

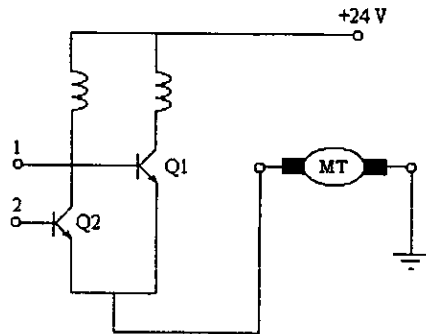


Figura 3.2. Diagrama esquemático del servoamplificador (SA) conectado al motor y con un control de armadura.

En el diagrama de la figura 3.2 se muestran las dos inductancias que muestran los embobinados de campo del motor y que son de las mismas características, número de vueltas y calibre del conductor, por lo cual al aplicarles voltajes del mismo valor la fem inducida tendrá el mismo valor, pero signo contrario, causando con esto que el motor no gire. Q1 y Q2 son dos transistores de potencia que al ser puestos en conducción ocasionan el giro del motor en uno o en otro sentido dependiendo del voltaje aplicado a cada transistor.

Para que el motor gire en un sentido o en otro es necesario que haya una diferencia de potencial en las terminales "1" y "2"; esto es, al aplicar voltajes positivos en la entrada "1" el motor girará en un sentido y por el contrario si se aplica el voltaje positivo a la terminal "2" el motor girará en sentido opuesto. El sentido del giro depende a cual de las dos terminales se aplique el mayor potencial eléctrico. Para que el motor empiece a girar es necesario la aplicación de un voltaje mínimo debido a la fricción estática, sin embargo el nivel de voltaje

3. El control Derivativo (D).
4. El control con las combinaciones de los anteriores: (PI), (PD), (PID).

Para llevar a cabo el control de posición angular del servomotor de corriente directa (c.d.) se realiza un alambrado con un controlador inicial de tipo proporcional; con la característica de comparar los voltajes de entrada y de retroalimentación para obtener el voltaje de error y bajo el concepto de comparación de voltajes al utilizar la configuración de retroalimentación negativa.

De forma gráfica y en diagrama de bloques, la acción de control para el servomotor de corriente directa se muestra en la figura 3.1.

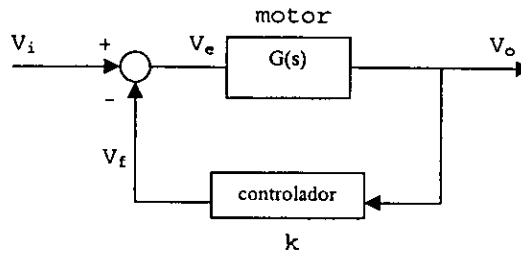


Figura 3.1. Diagrama de bloques para el control del servomotor.

Donde:

- V_i : representa la señal de entrada.
- V_e : la señal de error.
- V_f : la señal de retroalimentación.
- V_o : la señal de salida.
- $G(s)$: representa la función de transferencia de la planta.
- k : Representa el tipo de control proporcional.

Módulos del servomotor de corriente directa

El servomotor de corriente directa consta de un elemento fundamental que es un motor de c.d. con características particulares y el cual representa la planta en el diagrama de bloques. Este motor cuenta con un tacogenerador, transductor que genera un voltaje proporcional a la velocidad angular; sin embargo no es importante para la forma en que se realiza el control de la posición del eje del motor, puesto que no es usado. El motor convierte energía eléctrica en energía mecánica, utilizando para ello un voltaje de alimentación de +24 volts generados por otro módulo que es una fuente de poder. El motor de corriente directa

3.1 CONTROL DE POSICION

Servomecanismos

Durante la práctica profesional el ingeniero se encuentra con frecuencia sistemas o procesos trabajando con servomecanismos. El uso de los servomecanismos puede tener como finalidad el llevar a cabo acciones de control tales como la velocidad de máquinas centrífugas, generadores, motores, etc; así como controlar la posición de antenas parabólicas, rastreadores solares, elevadores, etc; o también se pueden usar los servomecanismos para controlar tanto la velocidad como la posición de robots industriales, prótesis mioeléctricas, brazos-robot, etc.

Servomotor de corriente directa

En el amplio campo de la industria es muy común encontrar diversos tipos de mecanismos que controlan la velocidad de giro o la posición de elementos rotatorios, entre muchos otros. Los mecanismos que llevan a cabo éstos tipos de control pueden ser tan complejos como lo sea el proceso que se esta controlando. Si lo que se desea es llevar a cabo un control de posición angular de una masa es necesario obtener información de la posición de salida y que esta pueda ser comparada con una cierta posición de referencia; para que así, mediante la acción de control, se llegue a la posición deseada.

Un control automático compara el valor de salida de una planta con el valor deseado, determina la desviación y produce una señal de control que reduce la desviación a un error pequeño; la forma como el control automático produce la señal de control recibe el nombre de acción de control.

Cuando se realiza un análisis teórico o el diseño de algún sistema de control uno de los primeros puntos y más importantes a considerar es que tipo de acción de control se utilizará para el sistema de interés. En ocasiones puede suceder que un controlador de un sistema no sea adecuado o eficaz para otro sistema diferente, ya que pueden no tener un mismo ancho de banda, una misma frecuencia de corte, el mismo rango de valores de ganancia para permanecer dentro de la región de estabilidad, o en general debido a las diferentes características intrínsecas de cada sistema.

Dentro de la teoría de control algunos modos más comunes para llevar a cabo acciones de control son:

1. El control Proporcional (P).
2. El control Integral (I).

requerido para que el motor comience a girar en un sentido no es el mismo que el requerido para que lo haga en el sentido contrario.

Para poder registrar los voltajes de salida de la planta se utiliza un potenciómetro de tipo angular (OP) de 10 k Ω ; el cual posee un filtro RC con efecto de minimizar los efectos de ruido y tener un comportamiento prácticamente lineal. Este potenciómetro está acoplado al eje lento del motor y funciona como un transductor, convirtiendo la posición de la flecha del motor a un voltaje proporcional al desplazamiento angular.

El diagrama eléctrico del servomotor de corriente directa funcionando como un sistema con control de posición es el de la figura 3.3.

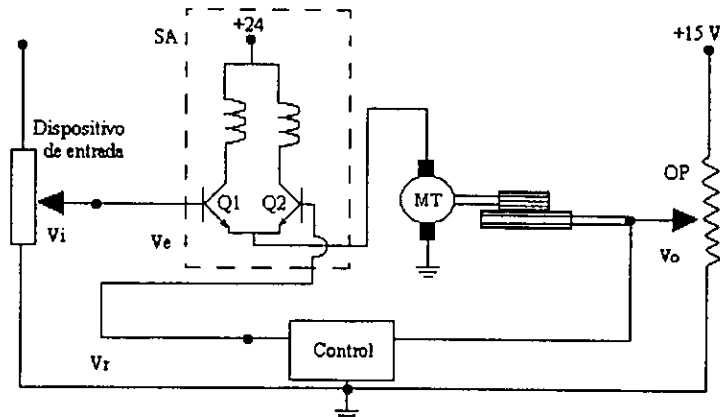


Figura 3.3. Diagrama del servomotor de c.d. funcionando como un sistema con control de posición.

Donde el dispositivo de entrada al sistema y el módulo de control pueden ser diferentes; por ejemplo, como entrada puede usarse un potenciómetro, similar al de salida, o directamente una computadora y como control una red neuronal o un control proporcional.

3.2 FUNCIÓN DE TRANSFERENCIA DE LA PLANTA

Para encontrar la función de transferencia del motor de corriente directa se elige como acción de control un controlador de tipo

Para poder obtener $G(s)$ se debe saber que tipo de respuesta tiene el sistema de lazo cerrado; esto puede determinarse conociendo la respuesta del sistema. Para ello puede usarse un graficador o un osciloscopio de memoria conectado a la salida del sistema; o sea, en el potenciómetro de salida (OP) de la figura 3.4.

El servomotor de c.d. con el control proporcional da una respuesta de tipo de segundo orden; figura 3.6; para cualquier valor de K_p ; por lo que la función de transferencia del motor se puede ir aproximando si se conocen algunos parámetros importantes de la respuesta de segundo orden del sistema y que pueden ser determinados de las gráficas obtenidas experimentalmente.

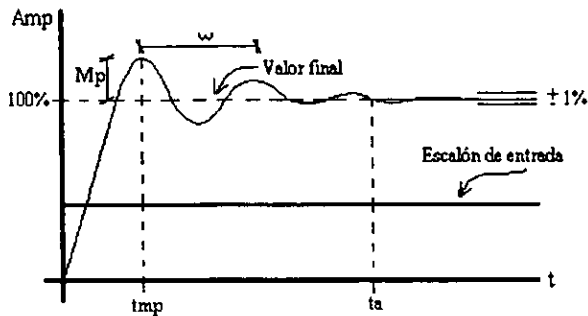


Figura 3.6. Respuesta del sistema de segundo orden.

Los parámetros que más interesan de la respuesta del sistema son el sobrepaso M_p y el tiempo de asentamiento t_a ; sin embargo también puede conocerse la frecuencia de oscilación ω ; debido a que en forma general para un sistema de segundo orden la función de transferencia se puede determinar como:

$$H(s) = \frac{\omega_n^2}{s^2 + 2\delta\omega_n s + \omega_n^2} \quad (\text{XIV})$$

donde δ es el factor de amortiguamiento y es determinado por la ecuación:

$$\delta = \frac{[\text{Ln}^2(M_p)]}{[\text{Ln}^2(M_p) + \pi^2]} \quad (\text{XV})$$

En la figura 3.5 G(S) representa la función de transferencia del motor y analizando los voltajes tenemos que:

$$V_o = G(s) V_e \quad (\text{III})$$

$$V_e = V_i - V_f \quad (\text{IV})$$

$$V_f = K_p V_e G(s) \quad (\text{V})$$

$$V_f = R' (V_o/R) \quad (\text{VI})$$

sustituyendo (VI) en (IV):

$$V_e = V_i - (R' V_o/R) \quad (\text{VII})$$

sustituyendo (VII) en (III):

$$V_o = G(s) [V_i - R' V_o/R] \quad (\text{VIII})$$

desarrollando la ecuación (VIII):

$$\frac{V_o}{V_i} = \frac{G(s)}{1 + [G(s)(R'/R)]} \quad (\text{IX})$$

como $R > 0$ y como generalmente es cierto que $G(s)(R'/R) \gg 1$:

$$\frac{V_o}{V_i} = \frac{G(s)}{G(s)(R'/R)} \quad (\text{X})$$

que es la función de transferencia del sistema completo y en lazo cerrado. Desarrollando (X):

$$\frac{V_o}{V_i} = \frac{R}{R'} = K_p \quad (\text{XI})$$

Entonces de la ecuación (XI) se comprueba que se tiene un sistema con acción proporcional K_p y cuyo valor es dado por la relación de resistencias R/R' , ubicadas en la trayectoria de retroalimentación y determinadas por la posición del cursor del potenciómetro de $2k\Omega$.

Por otra parte sustituyendo (XI) en la ecuación (X) para el sistema de lazo cerrado se obtiene la función de transferencia:

$$H(s) = \frac{G(s)}{1 + K_p G(s)} \quad (\text{XII})$$

y a partir de la cual se puede obtener la función de transferencia $G(s)$ para el motor:

$$G(s) = \frac{H(s)}{1 - K_p H(s)} \quad (\text{XIII})$$

proporcional con una ganancia k . Este tipo de controlador es implementado en el sistema con el diagrama de la figura 3.4.

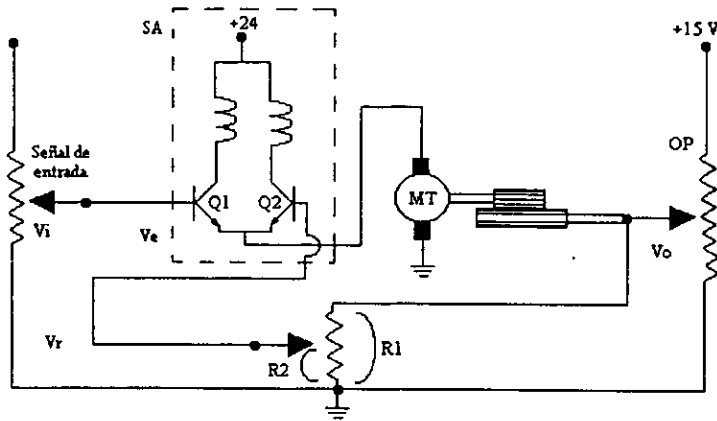


Figura 3.4. Servomotor de c.d. con control proporcional.

Para llevar a cabo la acción de control proporcional del sistema se utiliza como controlador un potenciómetro de tipo lineal de $2k\Omega$.

Por lo tanto el diagrama de bloques del sistema con control proporcional es el siguiente:

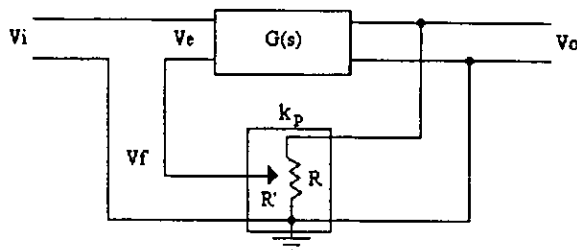


Figura 3.5. Diagrama de bloques del control proporcional.

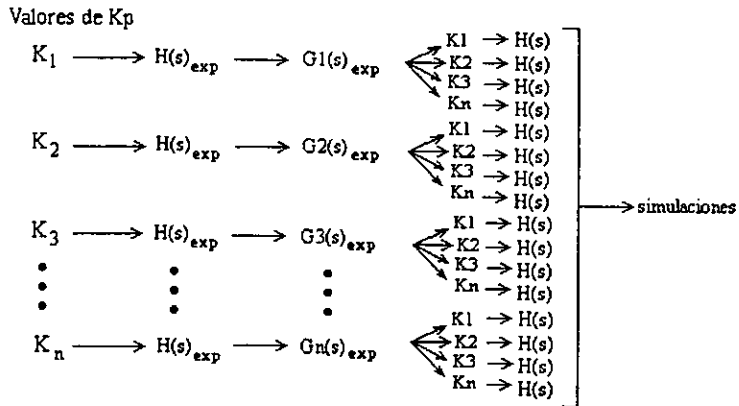
y sustituyendo la ecuación (XIV) en (XIII) la función de transferencia de la planta queda como:

$$G(s) = \frac{\omega_n^2}{s^2 + 2\delta\omega_n s + \omega_n^2(1-K_p)} \quad (\text{XVI})$$

Pudiendo ser medidos los parámetros ω_n y M_p directamente de gráficas obtenidas experimentalmente con un graficador y determinar también el tiempo de asentamiento. A continuación se presenta la secuencia llevada a cabo para llegar a determinar la función de transferencia del motor, representado en la figura 3.1 como la planta del sistema en lazo cerrado.

1. Obtención de las gráficas de respuesta del sistema para diferentes valores de K_p y medición del sobrepaso (M_p), frecuencia natural (ω_n) y tiempo de asentamiento (t_s). Para determinar por medio de la ecuación (XV) el factor de amortiguamiento (δ) y determinar entonces una primer función de transferencia $H(s)_{exp}$ de lazo abierto para cada valor de K_p .
2. Para cada una de las $H(s)_{exp}$ obtenidas en el punto anterior se obtiene la correspondiente $G(s)_{exp}$, por medio de la ecuación (XIV) o (XVI).
3. De acuerdo al diagrama de bloques del sistema en lazo cerrado, figura 3.1, y de la ecuación (XII), para cada una de las $G(s)_{exp}$ obtenidas en el punto anterior se asignan todos y cada uno de los valores de K_p escogidos, (con los que se obtuvieron experimentalmente las gráficas de respuestas), encontrando una nueva función de transferencia $H(s)$ del sistema en lazo abierto.
4. Con las nuevas aproximaciones de la función de transferencia $H(s)$ se realiza una primera simulación como un sistema de lazo abierto en CC.
5. El programa de simulación también dió respuestas del sistema del tipo de segundo orden, como las de la figura 3.6, y de ahí se pudieron escoger los valores de K_p con las cuales las respuestas tenían menos diferencia con las gráficas obtenidas experimentalmente. Estos valores de K_p fueron: $K_p = 1.7575, 1.5699, 1.4303, 1.1209, 1.0498, 1.0$.

En forma gráfica los puntos mencionados anteriormente se ilustran a continuación en la figura 3.7.

Figura 3.7. Aproximaciones para $H(s)$.

Con los valores de K_p seleccionados en el punto 5 y una vez realizadas las simulaciones de cada $H(s)$, obtenida de cada uno de los valores de dichas K_p , se compara cada una de las respuestas simuladas, con las gráficas de respuestas experimentales (obtenidas con el graficador); resultando que la que más se acerca a las reales son las $H(s)$ obtenidas con $K_p=1$; ya que tanto los porcentajes de sobrepaso como el tiempo de asentamiento se acercan más al medido en las gráficas obtenidas con el graficador. La función de transferencia del sistema como lazo abierto y la función de transferencia de la planta en lazo cerrado son las siguientes:

Para el sistema en lazo abierto:

$$H(S)_{exp} = \frac{15.20}{S^2 + 3.71S + 15.20} \quad (XVII)$$

Función de la planta del sistema en lazo cerrado:

$$G(s) = \frac{15.20}{S^2 + 3.71S} \quad (XVIII)$$

Para comprobar si en realidad estas ecuaciones caracterizan al sistema del servomotor de c.d. con control proporcional se toman cada una de las ecuaciones $G(s)_{exp}$, de las K_p seleccionadas en el punto 5 (1.7575,

1.5699, 1.4303, 1.1209, 1.0498, 1.0), se transforman a variables de estado y entonces como lazo cerrado se hacen otras simulaciones en el simulador SIMNON. Se toma $G(s)$ por que junto con el diagrama de bloques con control proporcional se puede hacer la simulación en Simnon; el cual tiene la posibilidad de dar las respuestas en diferentes puntos del diagrama de bloques a la vez; esto es que con simular una $G(s)$ se puede conocer la respuesta total del sistema y la de cualquier otro punto, como la salida del controlador, la entrada al sistema, la salida del punto suma, etc.

La secuencia utilizada para llegar a la función de transferencia de la planta $G(s)$ a partir de las nuevas simulaciones fue similar a la anterior:

1. Con las K_p elegidas en el punto 5 anterior se obtiene una $G(s)_{exp}$ y con estas se hace una simulación para todas y cada una de las seis K_p .
2. Se analizan las diferencias de las respuestas del sistema real y las respuestas de las simulaciones; simulando de forma similar al esquema de la figura 3.7; solo que las simulaciones se realizan directamente con la $G(s)_{exp}$, las K_p y el diagrama de bloques.
3. Una vez que se tienen las respuestas de las simulaciones se escoge la $G(s)_{exp}$ que entrega respuestas totales del sistema, para cada una de las seis K_p elegidas, con menor error respecto a las reales.

Al analizar todas las respuestas las curvas más parecidas a las reales son las que entregó $G(s)_{exp}$ obtenida de $K_p=1$ de donde $G(s)$ es:

$$G(s) = \frac{15.20}{s^2 + 3.71s} \quad (XIX)$$

la cual es la misma $G(s)$ obtenida con las simulaciones en CC, ecuación (XVIII) y es la $G(s)$ que caracteriza la planta del sistema; es decir el motor.

Tanto en el simulador CC como en Simnon las respuestas son muy parecidas. Las siguientes son tablas comparativas de los parámetros obtenidos del sistema al realizar las simulaciones con las ecuaciones: $H(s)$ en lazo abierto para CC, $G(s)$ en lazo cerrado para Simnon y los obtenidos con el graficador.

$K_p = 1.7575$	%Mp	t_a
CC	29.87%	2.3seg
Simnon	31.10%	2.6seg
Exp.	26.50%	2.7seg

$K_p = 1.5699$	%Mp	t_a
CC	27.53%	2.5seg
Simnon	29.16%	2.4seg
Exp.	25.16%	3.1seg

$K_p = 1.4303$	%Mp	t_a
CC	25.60%	2.6seg
Simnon	27.69%	2.6seg
Exp.	24.37%	2.6seg

$K_p = 1.1209$	%Mp	t_a
CC	20.58%	2.2seg
Simnon	22.32%	2.8seg
Exp.	24.54%	2.7seg

$K_p = 1.0498$	%Mp	t_a
CC	19.25%	2.3seg
Simnon	21.98%	3.0seg
Exp.	20.40%	2.6seg

$K_p = 1.0$	%Mp	t_a
CC	17.78%	2.4seg
Simnon	20.79%	2.3seg
Exp.	17.78%	2.4seg

Una vez que se tiene la función de transferencia de la planta o motor pueden hacerse cambios en el diagrama de bloques; ya que se tienen las ecuaciones tanto de la planta $G(s)$, como del control proporcional K_p ; por lo que podría modificarse el sistema sin problema a un diagrama de bloques como el de la figura 3.8.

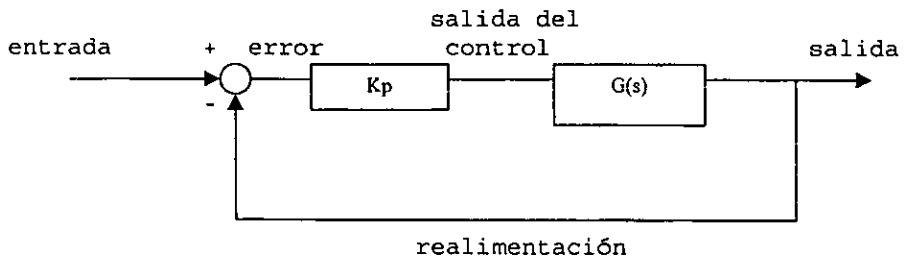


Figura 3.8. Diagrama de bloques modificado.

Al simular en Simnon el diagrama de bloques de la figura 3.8 la respuesta del sistema siguió siendo del tipo de segundo orden y no tuvo variación significativa con respecto a la respuesta real del sistema en cuanto a los parámetros importantes M_p y t_a ; lo que lleva a que la $G(s)$ encontrada caracteriza al motor; figura 3.9.

Figura 3.9. Simulación del sistema de la figura 3.8.
(A) (B)

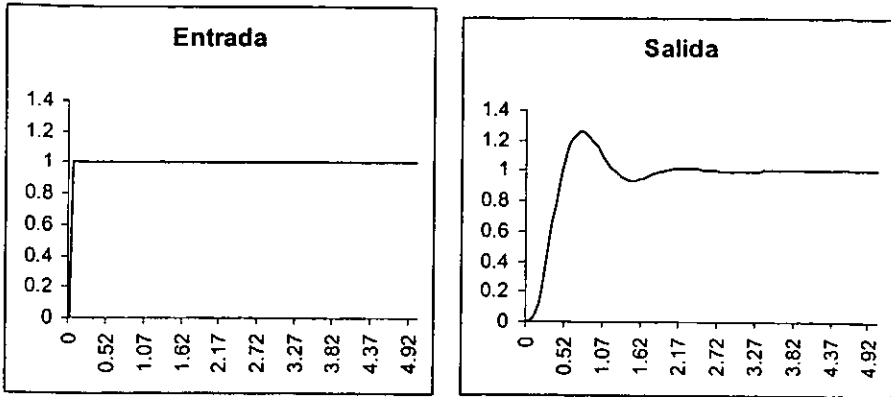
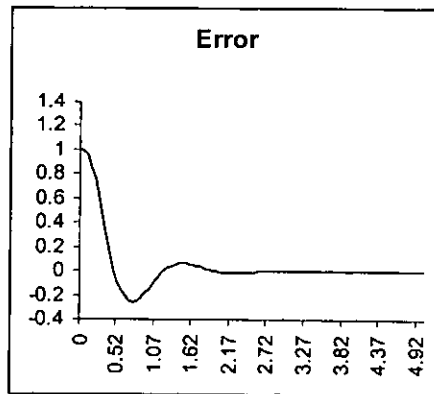


Figura 3.9. (C)



La figura 3.9 A muestra el escalón de entrada de 1 volt aplicado al sistema; la figura 3.9 B muestra la respuesta total del sistema según la entrada escalón dada, el tiempo de asentamiento así como el sobrepaso que tiene; en la figura 3.9 C se observa como cambia el error en el punto suma del sistema según la diferencia entre la señal de referencia y la de realimentación.

En el servomotor de corriente directa se puede realizar dicho cambio, sin embargo para la realización de esta tesis no se lleva a cabo por dos motivos entre otros: no se puede cambiar de lugar el motor físicamente; ya que tanto el motor como el servoamplificador, que funciona como punto suma, se encuentran unidos internamente y por que el

punto principal de la presente tesis es la implementación del controlador con una red neuronal artificial. Sin embargo, este pequeño análisis del diagrama de bloques de la figura 3.8 y las gráficas obtenidas en la figura 3.9 nos ayudan a comprobar que la aproximación de la función de transferencia obtenida es adecuada.

Entonces de acuerdo al diagrama de bloques original la simulación en Simnon del diagrama de bloques de la figura 3.1 con $G(s)$ de la ecuación (XIX) y con una señal de entrada en forma de escalón de amplitud unitaria, presenta las siguientes curvas de respuestas:

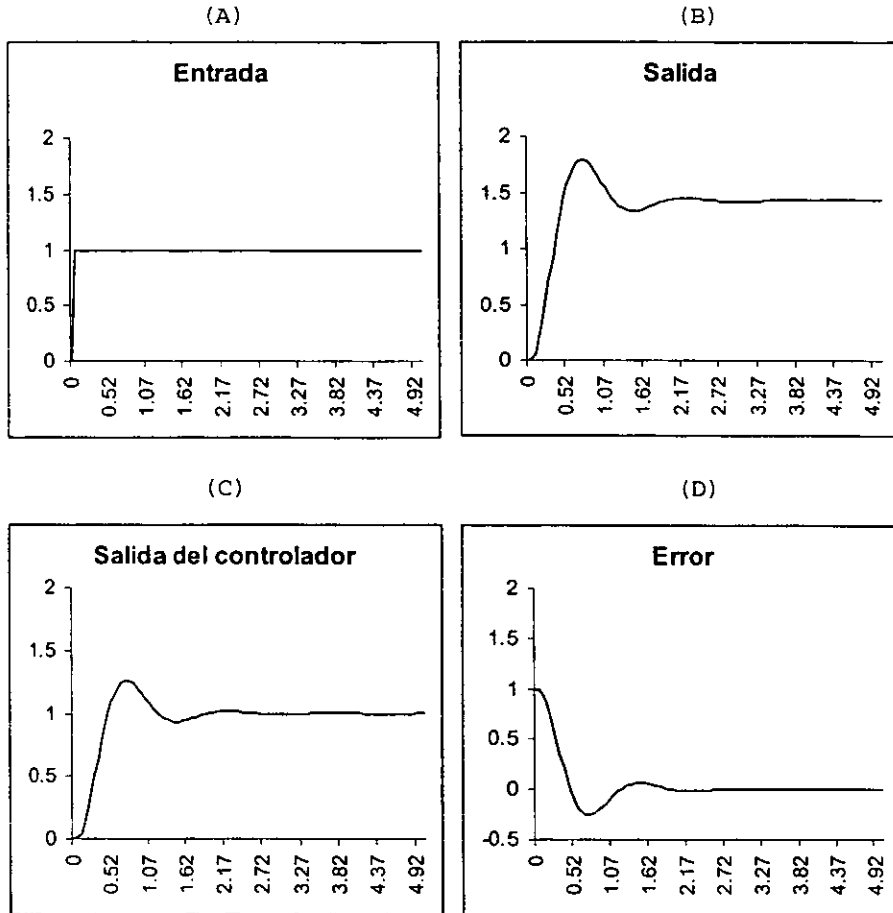


Figura 3.10. Simulación del sistema de la figura 3.1 con una entrada escalón unitario, (A); respuesta del sistema de acuerdo a la entrada escalón, (B); salida del control proporcional del sistema (C); error o diferencia entre la salida total y la referencia de entrada en el punto

suma del sistema. Es importante notar que tanto estas gráficas como las de la figura 3.9 son iguales, exceptuando la de salida del control, que cambia debido su ubicación en el diagrama de bloques.

En las gráficas de la figura 3.10 se tiene un sistema con control proporcional donde la K_p del sistema tiene un valor de 1.4303. La salida del sistema es mayor a la salida del control proporcional debido al tipo de controlador empleado, figura 3.5, el mínimo valor que puede tener K_p es 1, ecuación (XI), teniendo en este caso un voltaje máximo igual al de la salida del sistema. Al tener una $K_p = 1.4303$ la ecuación matemática del control proporcional queda como:

$$u = y/K_p \quad (XX)$$

donde u es la salida del control proporcional, y es la salida del sistema y K_p es la constante de proporcionalidad.

La señal de error, que físicamente es la salida del servoamplificador, va disminuyendo hasta llegar a un valor de cero; que es cuando en las dos entradas (transistores) del servoamplificador hay el mismo voltaje y cuando el motor ha llegado a la posición correspondiente. De acuerdo a la figura 3.4, $V_i = V_r$.

3.3 CONTROL DE POSICIÓN DEL SERVOMOTOR DE C.D. CON LA RED NEURONAL

Una vez caracterizado el sistema de control de posición se puede realizar el control de la planta con otro tipo de controlador: una red neuronal artificial (RNA).

La primera consideración que se debe hacer es que existe una diferencia con el control proporcional implementado anteriormente y es que este control es de tipo continuo; mientras que como la red neuronal artificial será implementada vía una computadora el control es de tipo discreto.

Del análisis hecho anteriormente tenemos que la respuesta del sistema es de segundo orden y en el capítulo anterior, punto 2.1, se definió una topología de la red neuronal artificial de cuatro neuronas de entrada, tres en la segunda capa y dos de salida; sin embargo como solo se utiliza una neurona de salida; la capa de salida solo tendrá una neurona.

Para poder entrenar una primer red neuronal artificial se deben formar los patrones de entrenamiento; para lo cual se van a utilizar los datos de las respuestas obtenidas al simular el sistema en Simnon. Los

datos que se requieren son: las salidas del controlador y las de salida del sistema, que son las entradas al controlador, V_0 y V_i en la figura 3.5. Simon entrega tanto las gráficas como la lista numérica de los valores para tiempos iguales en cada punto que se desee del sistema simulado.

Para la primer red neuronal artificial se utilizan los datos de la figura 3.9; como entradas a la red se toman los datos de salida del sistema y como salida de la red para un patrón se toma su correspondiente valor de salida del controlador. Un patrón de entrada en cualquier tiempo consiste de cuatro valores numéricos, recordemos que son cuatro neuronas de entrada, los cuales sirven de entrada a la red y si esta ya ha sido entrenada tendrá en la neurona de salida el correspondiente valor a dichas entradas. Tal vez cuando se entrene la red no se le muestren todos los patrones posibles que se le puedan presentar; sin embargo si ha tenido un buen "aprendizaje" tendrá la capacidad de generalizar y entregar una salida correspondiente a dichas entradas. Sería poco práctico tener que formar patrones para cada escalón de entrada que se le puede dar al sistema y para cada valor posible de entradas que se pudieran presentar; además de que se podría crear un conflicto; pues podría haber patrones de entrada repetidos con valores de salida diferentes, caso en el cual la red no tendría un buen aprendizaje. En la siguiente figura se muestra como se utilizan las gráficas de la figura 3.9 para formar los patrones de entrenamiento.

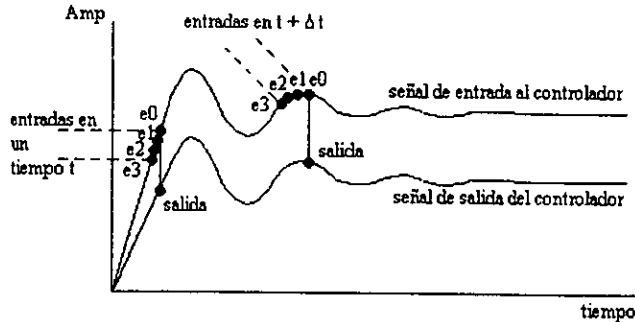


Figura 3.11. Formación de patrones de entrenamiento.

En la figura 3.11 se puede ver como las entradas a la red, e_0 e_1 e_2 e_3 , van recorriendo todos y cada uno de los valores de la señal de entrada y se les asigna su correspondiente valor de salida. Para un instante "t" las entradas tienen un determinado valor: $e_0 = w$, $e_1 = x$, $e_2 = y$, $e_3 = z$, salida = s; para el tiempo próximo siguiente los valores de las entradas se recorren un valor teniendo e_0 el nuevo valor de la señal: $e_0 = v$, $e_1 = w$, $e_2 = x$, $e_3 = y$, salida = r; y así sucesivamente hasta recorrer completamente todos y cada uno de los datos con los que se

cuente en la gráfica. Hay que considerar que el primer patrón que se debe formar contiene ceros (o su valor de inicio) en todas las entradas y cero (o el correspondiente valor de inicio) en su salida, ya que el servomotor debe tener una posición inicial antes de recibir cualquier entrada. Entonces formados todos los patrones de la señal se puede realizar el entrenamiento de la red neuronal artificial como se explicó en el capítulo anterior utilizando el programa Dynamind.

Para implementar el control del servomotor con una red neuronal artificial se utiliza el diagrama de la figura 3.12 donde se muestra como se utiliza la pc conectada al sistema; para esto se le agrega a la computadora una tarjeta adicional de adquisición de datos que es la PCLAB812, la cual nos permite llevar los datos del sistema directamente a la red neuronal mediante un programa de software y cuyas especificaciones se mencionan posteriormente en el apéndice B de esta tesis.

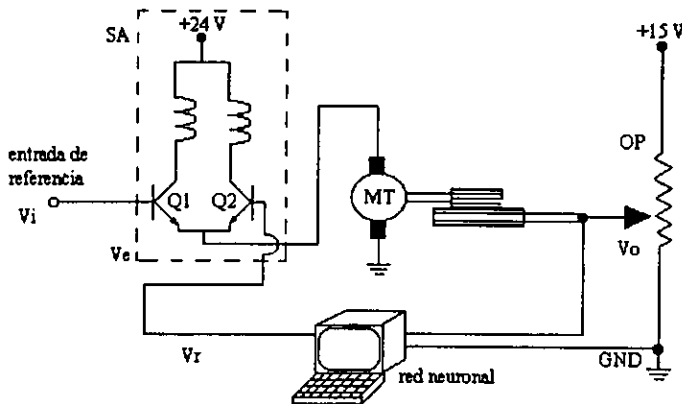


Figura 3.12. Control del servomotor en la PC.

La red neuronal artificial esta implementada dentro del programa servo.exe; el cual contiene el software necesario para trabajar también con la tarjeta de adquisición de datos PCLAB812. El programa servo.exe se encuentra en el apéndice A de esta tesis. La entrada al sistema puede ser dada ya sea con un potenciómetro, parecido al de salida (OP), o también desde la PC (la tarjeta de adquisición de datos tiene la posibilidad de manejar dos canales de entrada y dos de salida); es usada la PC para tener una señal de referencia más precisa tanto en amplitud como en tiempo.

El programa para introducir los datos de las entradas a la PC, procesar los datos en la red neuronal y enviar las salidas hacia el sistema se describe en la figura 3.13; en forma de bloques y de una manera simplificada:

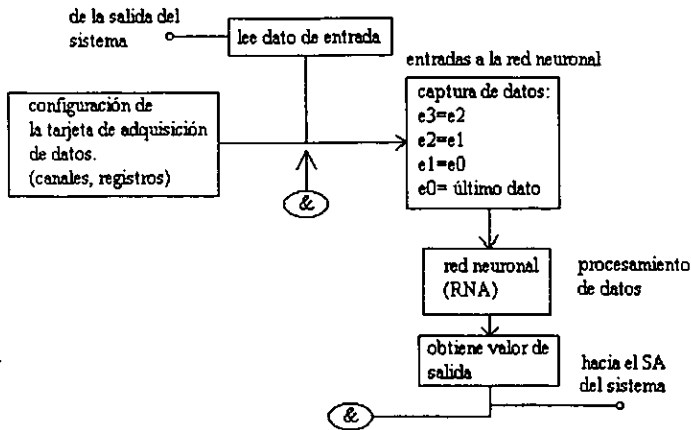


Figura 3.13. Diagrama del programa para la red neuronal artificial y la tarjeta de adquisición de datos.

Lo primero que se realiza en el programa es fijar la configuración de la tarjeta de adquisición de datos; después, ya en el programa principal, se hace la captura de un dato directamente de la salida del sistema para formar el patrón de entrada a la RNA; se procesa el patrón actual y una vez obtenida la salida de la RNA se envía hacia el sistema para poder leer un nuevo dato de la salida del sistema; formar el siguiente patrón de entrada y continuar con el proceso. Una vez que la salida de la RNA es prácticamente igual a la señal de referencia el motor deja de moverse; sin embargo el programa sigue corriendo, lo que sucede es que los patrones de entrada son iguales en ese momento y por lo tanto la salida de la RNA también. Puede entonces darse otra referencia de entrada para mover el motor a otra posición.

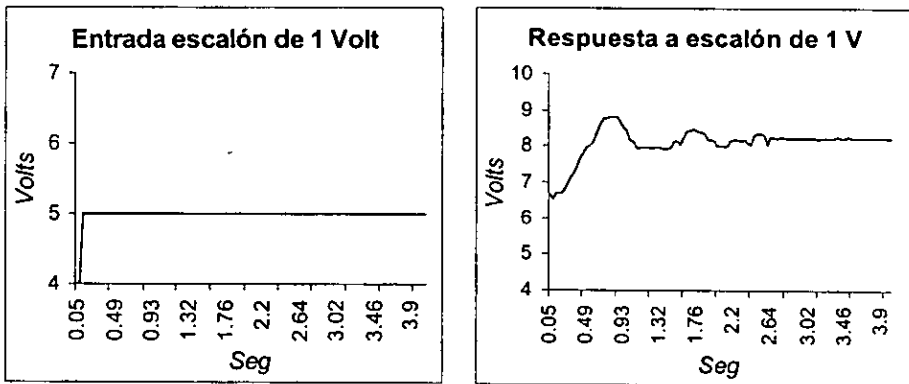
CAPITULO IV.

COMPARACION DE CONTROL
LINEAL Y CONTROL MEDIANTE
REDES NEURONALES.

4.1 COMPARACION DE CONTROL LINEAL Y CONTROL MEDIANTE REDES NEURONALES.

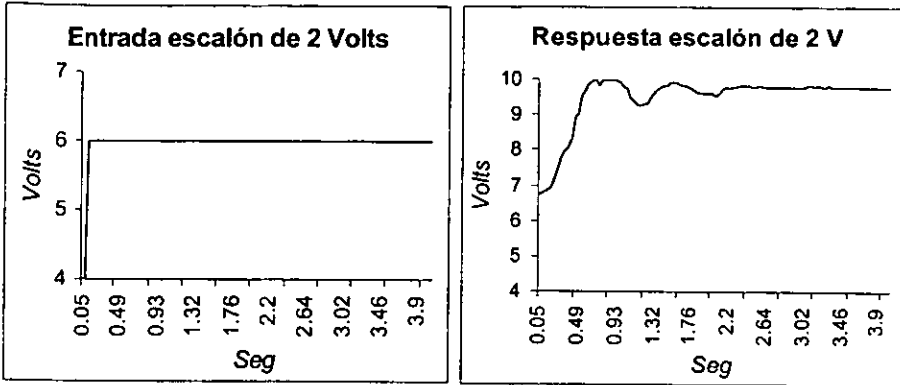
Como se explicó en el capítulo 3 el servomotor presenta una respuesta de tipo de sistema de segundo orden con un porcentaje de sobrepaso y un tiempo de asentamiento determinados. Las curvas de respuesta del sistema real, V_o en la figura 3.4, se obtienen para el valor de k del control proporcional usado en el capítulo anterior con el que se obtuvo la función de transferencia de la planta; este valor es de $k = 1.4303$. Con este valor de k el sistema con control proporcional responde de la siguiente forma para diferentes entradas de tipo escalón, figura 4.1.

Figura 4.1 A. Sistema con control lineal.



Del lado izquierdo se tiene la entrada al sistema o referencia; del lado derecho la respuesta del sistema a dicha entrada; se tiene un sobrepaso para un el escalón de 1 volt (4-5 V) de 30%; y un tiempo de asentamiento de 2.2 segundos. El eje vertical es la amplitud de la señal en volts mientras que el eje horizontal indica el tiempo en segundos.

Figura 4.1 B. Sistema con control lineal.



De igual forma el eje horizontal representa el tiempo en segundos y el eje vertical la amplitud en volts; con un escalón de entrada de 2 volts (4-6 V), gráfica de la derecha, el sistema presenta en la salida voltajes mayores a 10 volts, gráfica de la izquierda; es importante recordar que la tarjeta de adquisición de datos de la PC maneja un voltaje máximo de entrada de 10 volts, razón por la que los datos en la salida mayores a 10 son registrados como un voltaje máximo de 10 volts; es por ello que no puede medirse el sobrepaso en esta gráfica; en cuanto al tiempo de asentamiento este si se puede medir y es de 2.2 segundos similar al obtenido con la entrada anterior de 1 volt.

Físicamente en el sistema con el control lineal real no existe este límite; pero cuando se implemente algún tipo de control dentro de la PC mientras más se pase el voltaje en la salida del sistema de 10 volts éste se irá saliendo de control y perdiendo estabilidad, ya que la tarjeta de adquisición de datos registrara solamente voltajes de 10 volts como máximo y generando salidas del controlador erróneas. Para poder medir el sobrepaso cuando se le da un escalón de entrada de 2 volts veamos la gráfica a la salida del controlador, Figura 4.2.

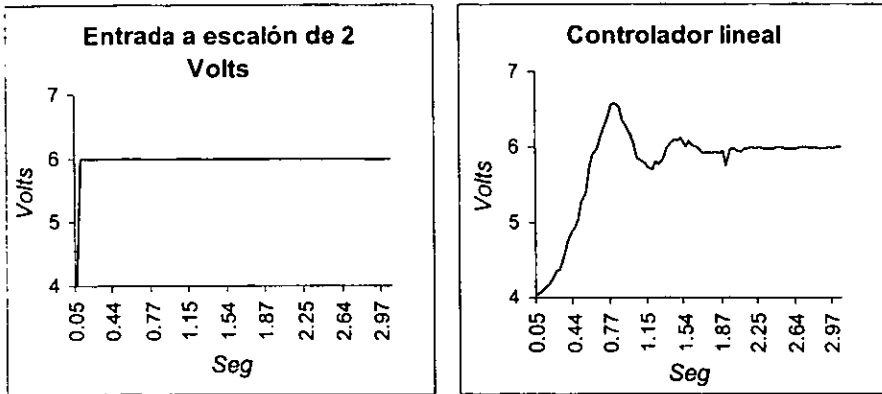


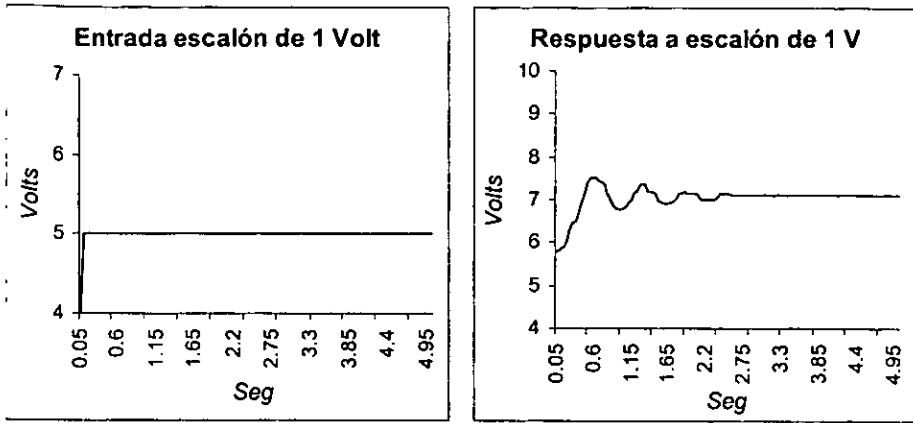
Figura 4.2. Curva de salida del control lineal.

Del lado derecho se tiene una gráfica del sistema en la salida del control lineal; en ella se tiene un sobrepaso del 27% y un tiempo de asentamiento similar al de la figura 4.1 de 2.2 segundos. A la izquierda se ve representada la entrada aplicada al sistema para este caso.

Al utilizar una PC para implementar un controlador este puede cambiar y de ser un control de tipo continuo pasa a ser un control de tipo discreto. Si conectamos la PC al sistema para que lleve a cabo la acción de control, ver figura 3.11, obtenemos respuestas del sistema como las mostradas en la figura 4.3; donde se puede observar el cambio de la respuesta del sistema, aumentando su tiempo de asentamiento y variando el porcentaje de sobrepaso según el tamaño de la entrada y dependiendo de la frecuencia de muestreo; es decir de la velocidad de adquisición de los datos de la tarjeta a la PC, que se fije en el controlador implementado.

La figura 4.3 presenta gráficas de respuesta del sistema con control discreto en la PC y con una frecuencia de muestreo óptima para la mejor acción de control posible y diferentes escalones de entrada.

Figura 4.3 A. Respuesta del sistema con control discreto en PC.



Con una entrada escalón de 1 volt (4-5 V) el sistema con control discreto aumenta el tiempo de asentamiento de su respuesta en comparación con el sistema con control lineal a 2.5 segundos y el sobrepaso a un 24%. En la gráfica del lado derecho se ve como la salida del sistema tiene mayor número de oscilaciones para la misma entrada.

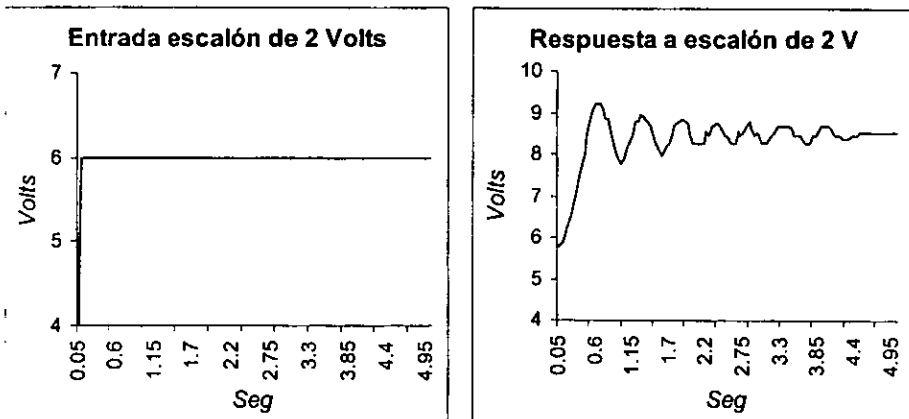


Figura 4.3 B. Respuesta del sistema con control discreto en PC.

Para una entrada escalón de 2 volts el tiempo de asentamiento aumenta hasta 4.28 segundos, obsérvese el número de oscilaciones en la gráfica del lado derecho, en algunos casos continúa oscilando aún más;

mientras que el sobrepaso continua en 24%. En general la diferencia más considerable con respecto al sistema lineal es el tiempo de asentamiento; ya que el sobrepaso presenta variaciones pequeñas. Además de que este sistema con control discreto tiene mayor inestabilidad.

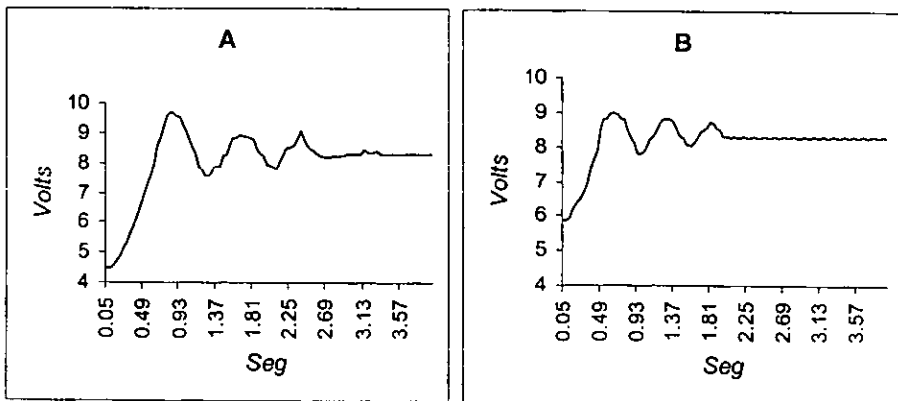
Para obtener las respuestas de la figura 4.3 A y B se fijó una frecuencia de muestreo (Fs) en el programa de la computadora de 100 Hz; debido a que con frecuencias menores el sistema presenta un tiempo de asentamiento cada vez mayor y varía mas el porcentaje de sobrepaso entre cada una de las respuestas; frecuencias de muestreo mayores no presentan cambios en la respuesta del sistema.

El programa dentro la computadora realiza la acción de control de acuerdo a la ecuación (XX): $u = y/k_p$; donde y es la salida del sistema y/o entrada al controlador, u es la salida del controlador y k_p tiene el valor de 1.4303.

Las curvas de respuesta del sistema con control lineal y con control discreto no pueden ser modificadas para cambiar el tipo de repuesta; es decir una vez fijados los parámetros e implementado el controlador el sistema responde siempre con el mismo tipo de respuesta, en este caso como sistema de segundo orden. Al implementar una red neuronal el diseñador del controlador tiene la posibilidad de ajustar la curva de respuesta del sistema o seguir con la misma, según la aplicación.

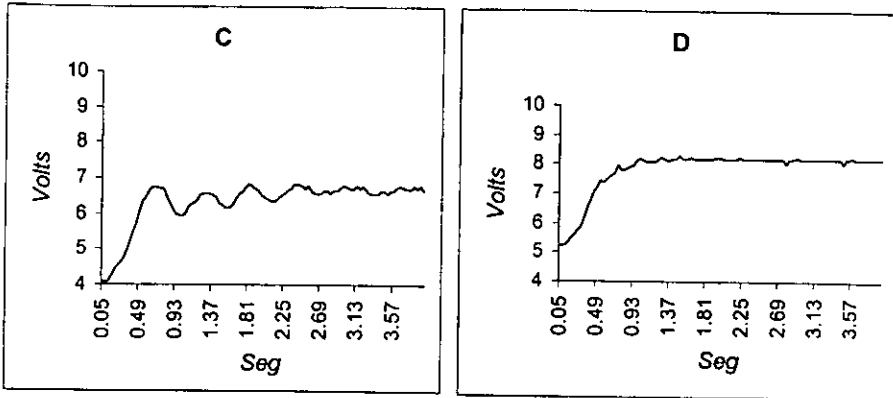
En la figura 4.4 se muestran curvas de respuesta del sistema obtenidas con diferentes redes neuronales; se puede ver como cambia la respuesta del sistema dependiendo de la red neuronal que este realizando la acción de control y del grado de entrenamiento de ella.

Figura 4.4. Respuestas del sistema con diferentes redes neuronales.



Durante el proceso de diseño del controlador con red neuronal la primera red, figura 4.4 A, fue entrenada con patrones formados de datos obtenidos del sistema lineal, modificados, para lograr que el sistema respondiera casi igual que el sistema con control lineal.

La figura 4.4 B presenta al sistema con otra red neuronal como controlador y la cual se obtuvo de patrones formados y modificados del sistema funcionando con la red de la figura 4.4 A logrando así disminuir el sobrepaso.



En la figura 4.4 C se muestra una respuesta del sistema con otra red neuronal como control, obtenida del sistema de la figura 4.4 B y modificando sus patrones de entrenamiento para disminuir ahora el tiempo de asentamiento en la respuesta del sistema.

La figura 4.4 D muestra la respuesta del sistema funcionando con una nueva red neuronal, entrenada con patrones formados de datos obtenidos de la figura 4.4 C y modificados para tener un sobrepaso y tiempo de asentamiento aún menores.

Las redes se fueron entrenando de una forma recursiva; colocando valores que se ajustaran al tipo de salida buscada. Para la realización de esta tesis lo que se buscó fue una respuesta más rápida, con el mínimo sobrepaso y con el menor número de oscilaciones posibles.

En la figura 4.5 se muestra como se ajustan las salidas de la red neuronal para cada patrón de entrada y obtener así la salida buscada.

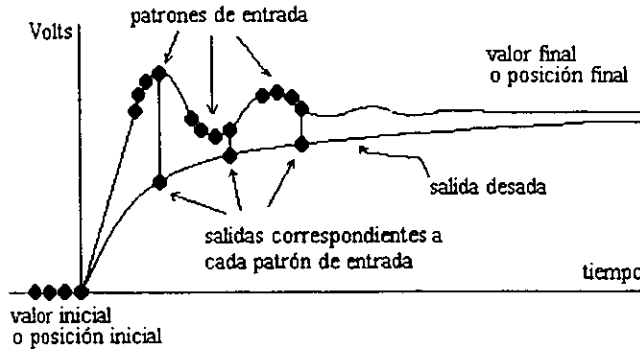
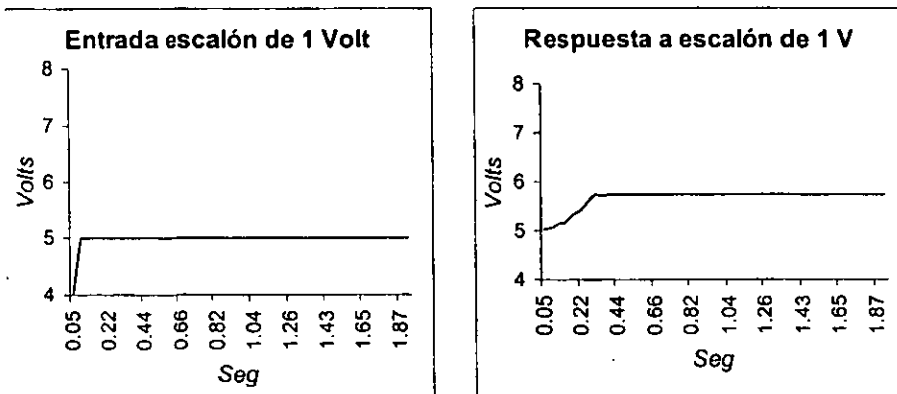


Figura 4.5. Modificación de los patrones de entrenamiento para obtener una salida deseada

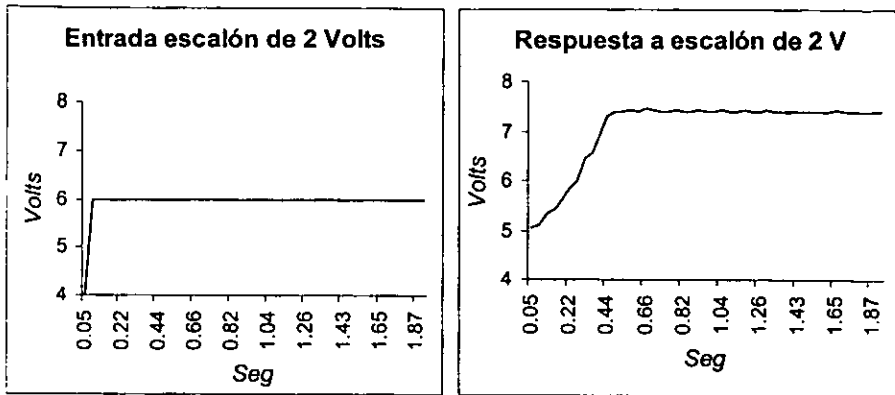
Después de haber realizado una serie de entrenamientos para diferentes redes neuronales, incluyendo los de la figura 4.4, y modificando las salidas para obtener una mejor respuesta del sistema con menor sobrepaso y tiempo de asentamiento se obtiene una red con poco error en su entrenamiento y también buena respuesta en cuanto a los parámetros que nos interesan. La figura 4.6 presenta respuestas del sistema con esta red neuronal y para diferentes valores del escalón de entrada.

Figura 4.6 A. Respuestas del sistema con la red neuronal artificial.



La figura 4.6 A muestra la respuesta del sistema con una red neuronal ya definida y entrenada para tener un tiempo de asentamiento pequeño, 0.4 segundos y un sobrepaso mucho muy bajo; la entrada es un escalón de 1 volt (4-5 V).

Figura 4.6 B Respuestas del sistema con la red neuronal artificial.

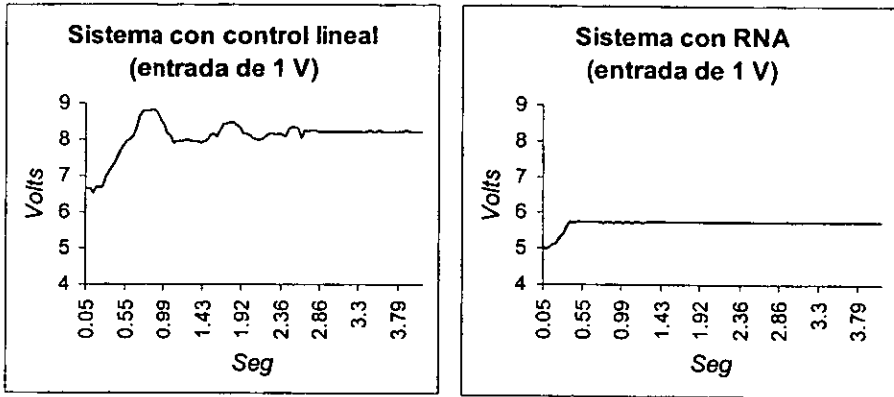


En la figura 4.6 B la entrada aplicada al sistema es de 2 volts (4-6 V) con la misma red neuronal como controlador y presenta un tiempo de asentamiento de 0.5 segundos y un sobrepaso prácticamente de 0%. El sistema se comporta bastante estable a las dos entradas mostradas.

Un punto importante de resaltar es que con esta red neuronal aunque el escalón de entrada sea de 2 volts la salida en el sistema no sobrepasa el límite superior de 10 volts de la tarjeta de adquisición de datos; por lo cual aumenta el límite máximo de entrada, y por lo tanto el rango de control de posición del eje del motor, hasta una entrada cercana a los 2.5 volts. Esto se logra al modificar los patrones de entrenamiento de la RNA. Sin embargo al igual que con el control lineal al sobrepasar este límite el sistema se puede ir desestabilizando.

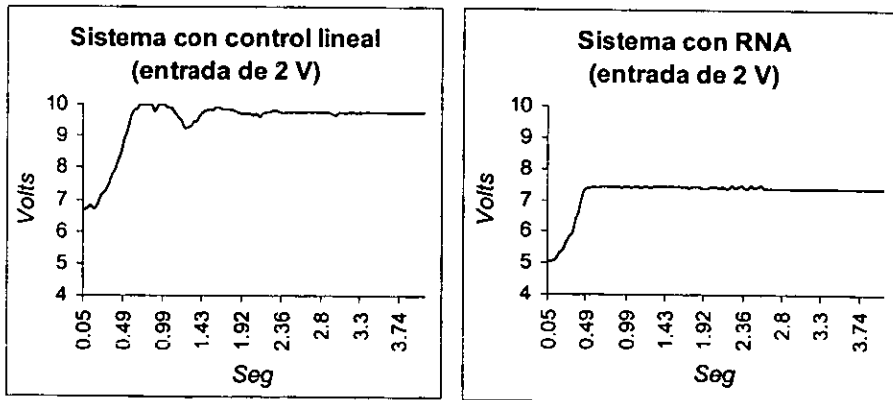
Al comparar la respuesta del sistema, para una misma entrada determinada, tanto con red neuronal como con control lineal, figura 4.7, observaremos mejor las diferencias de parámetros como el tiempo de asentamiento y porcentaje de sobrepaso.

Figura 4.7 A. Respuestas del sistema con control lineal y con RNA.



Del lado izquierdo se muestra la respuesta del sistema con control lineal y del lado derecho se muestra la respuesta del sistema con RNA como control para una entrada de 1 volt.

Figura 4.7 B. Respuestas del sistema con control lineal y con RNA.

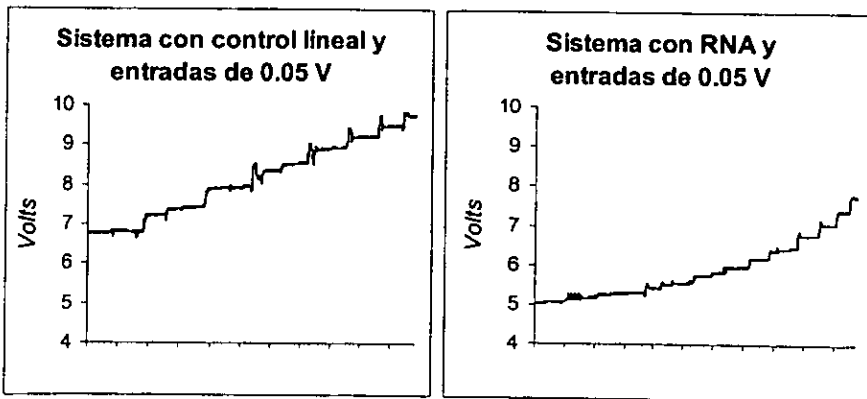


De igual forma la figura 4.7 B muestra las diferencias entre las respuestas del sistema funcionando con dos diferentes tipos de controladores, lineal y RNA, pero para una entrada de 2 volts; nótese que una diferencia es el tiempo para llegar a la posición final y el sobrepaso en las respuestas.

Una vez que la red neuronal hace que el sistema tenga una respuesta como la deseada podemos ahora analizar que tan bien logra generalizar sus salidas a entradas relativamente pequeñas y consecutivas no mostradas en los patrones de entrenamiento y sin tener que regresar a la posición de inicial.

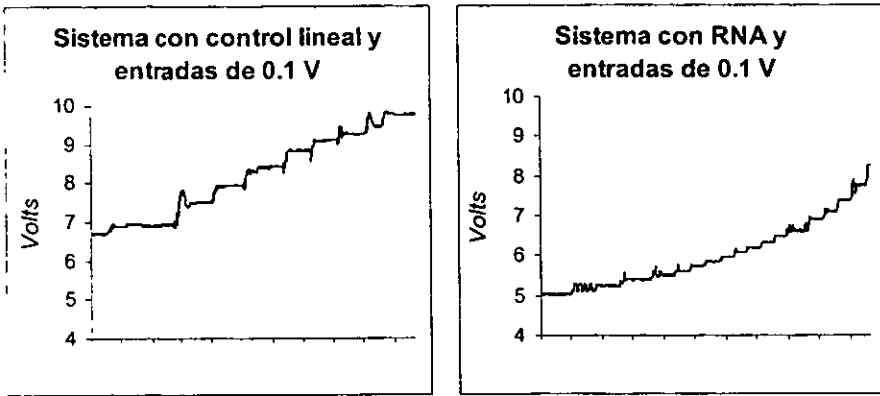
Las gráficas de la figura 4.8 muestran como la red neuronal no solo funciona para la entrada con la cual se hicieron los patrones de entrenamiento; sino que generaliza para diferentes escalones de entrada de amplitudes pequeñas. El tiempo de asentamiento es muy pequeño para los dos tipos de controladores, lineal y red neuronal, la diferencia en el porcentaje de sobrepaso es más notable; el sistema con control lineal presenta un sobrepaso para éstas entradas relativamente pequeñas mientras que el sistema funcionando con la red neuronal como control responde con un sobrepaso muy bajo, prácticamente nulo.

Figura 4.8 A. Respuestas del sistema a entradas de 0.05 volts.



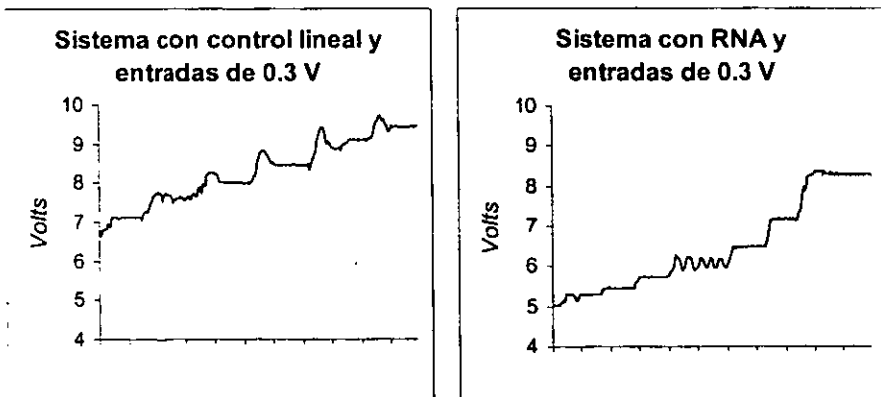
En las gráficas anteriores tenemos respuestas del sistema con control lineal y con la RNA seleccionada para pasos de 0.05 volts de entrada. Estas entradas de 0.05 volts llegan a 5 volts totales de entrada. Aquí lo importante es notar como responde el sistema con cada tipo de control, sin poner demasiado cuidado en la escala tiempo, principalmente con la RNA para determinar que tanto logra generalizar a entradas "desconocidas" y si es estable.

Figura 4.8 B. Respuestas del sistema a entradas de 0.1 volts.



En la figura 4.8 B se observan las respuestas del sistema nuevamente con los dos tipos de controladores pero ahora con pasos de entradas de 0.1 volts, de igual forma hasta 5 volts totales, y tenemos que el sistema con la RNA continua realizando una buena generalización de sus entradas para determinar la salida su correspondiente. Solo presenta algunas pequeñas oscilaciones alrededor de 5.1 volts de salida pero logra finalmente realizar el control. Obsérvese que en la figura 4.8 A en ese mismo valor de salida se presenta esta pequeña variación.

Figura 4.8 C. Respuestas del sistema a entradas de 0.3 volts.

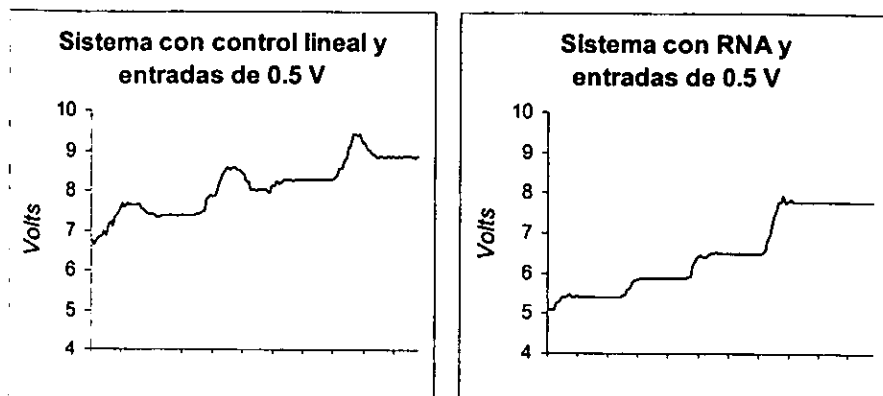


Ahora en la figura 4.8 C se muestran las respuestas de los sistemas a pasos de entrada más grandes, de 0.3 volts, y a pesar de que el sistema

con RNA responde de una buena manera existe un punto donde presenta algunas oscilaciones, además de un pequeño salto en el último paso; sin embargo para las demás entradas generaliza en una forma adecuada.

Hasta el momento las gráficas de la figura 4.8 C son las que presentan una variación mayor de la salida del sistema con respecto a los pasos de entrada.

Figura 4.8 D. Respuestas del sistema a entradas de 0.5 volts.

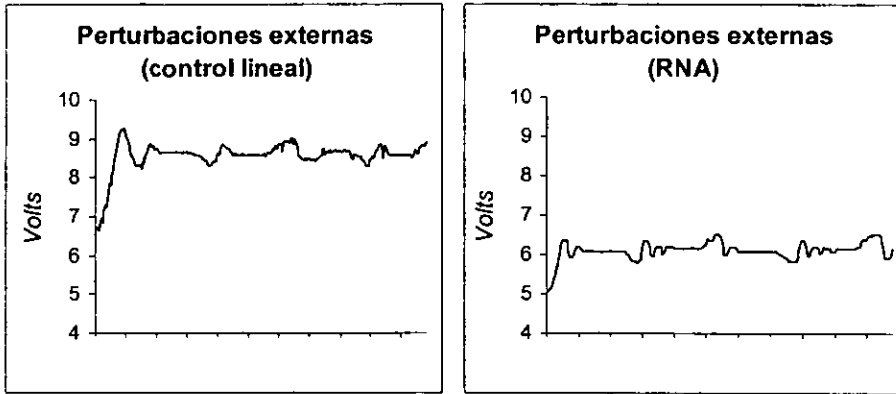


En la figura 4.8 D se observan las gráficas de respuesta a pasos de entrada 0.5 volts; la RNA generaliza bastante bien sus salidas a entradas "desconocidas" menores de 6.5 volts a la salida; sin embargo de forma similar a las anteriores cuando se rebasan los 7 volts presenta un salto.

Recordemos que con la RNA se aumenta el intervalo de control del sistema en comparación al control lineal y es al acercarnos a éste límite superior y con escalones de entrada mayores de 0.3 volts y menores de 1 volt cuando el sistema con RNA presenta este tipo de variaciones, que se pueden evitar no llegando a dicho límite o evitando éstos escalones de entrada en valores cercanos al límite superior en la entrada del sistema.

Analicemos ahora como responde el sistema con control lineal y con red neuronal a perturbaciones externas al sistema; esto por que puede ocurrir que con el eje del motor puesto en una posición determinada se presente alguna perturbación externa que haga que se mueva aleatoriamente en un sentido o en otro cambiando su posición sin haberle indicado nada a la entrada. Esta perturbación puede hacerse manualmente alterando el potenciómetro de salida de la figura 3.3. La figura 4.9 muestra el comportamiento del sistema, con los dos controladores, a algunas diferentes perturbaciones externas.

Figura 4.9. Respuestas del sistema a perturbaciones externas.



En la figura 4.9 se tiene que los dos sistemas siguen estables aún después de las perturbaciones externas; sin embargo una diferencia es que el control con RNA presenta mayor estabilidad en relación con el control lineal. Al simular una perturbación manualmente al sistema, con la RNA el eje del motor está más firme en su posición y por lo cual se altera con mayor dificultad que cuando se conecta el sistema con un control lineal; en las gráficas tenemos que con la RNA la perturbación no rebasa los 0.5 volts. Puede haber perturbaciones externas y el sistema no se desestabiliza por mucho tiempo; siempre y cuando no se rebase el límite de 10 volts, por la tarjeta de adquisición de datos de la PC, de lo contrario puede salirse de control el sistema.

Después de esta serie de pruebas para analizar al control lineal y el control con RNA nos falta probar al sistema con carga. La carga para el motor es simulada mediante un freno magnético (imán) puesto en el disco de eje rápido del motor. Figura 4.10.

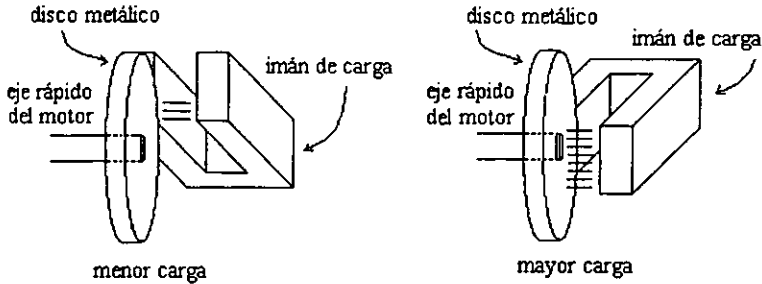
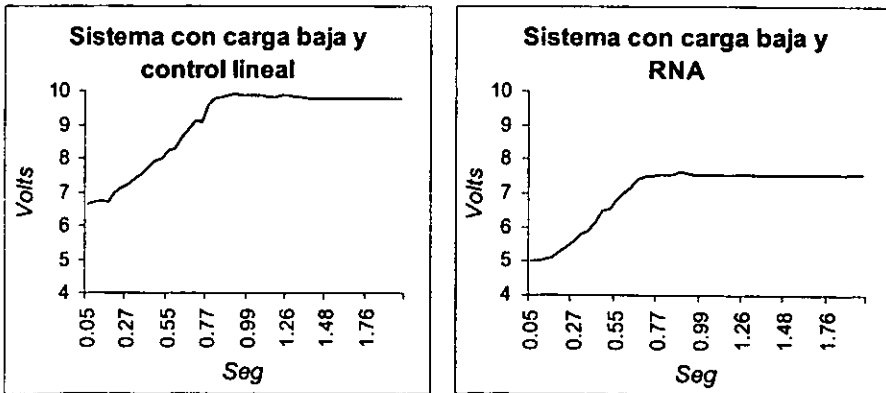


Figura 4.10. Freno magnético como carga en el motor.

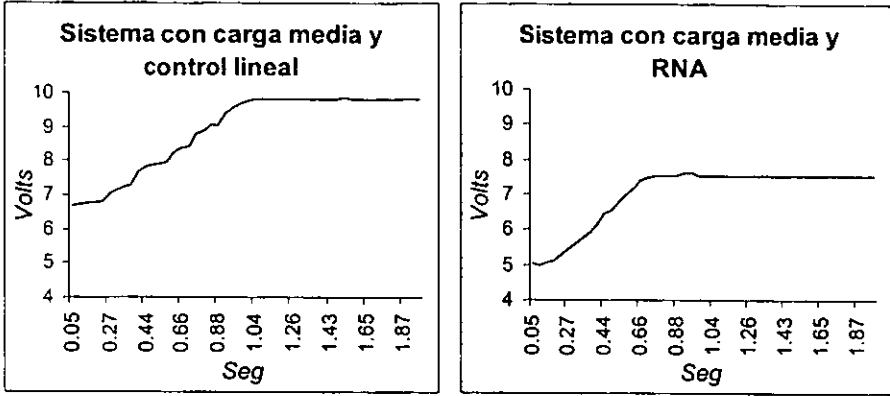
En las siguientes gráficas se muestran las curvas de respuesta del sistema con control lineal y con red neuronal a tres diferentes cargas. La carga es clasificada dependiendo de la posición en que se coloque el imán en el eje lento del motor; el campo magnético aumenta según la posición de este imán.

Figura 4.11 A. Sistema con carga.



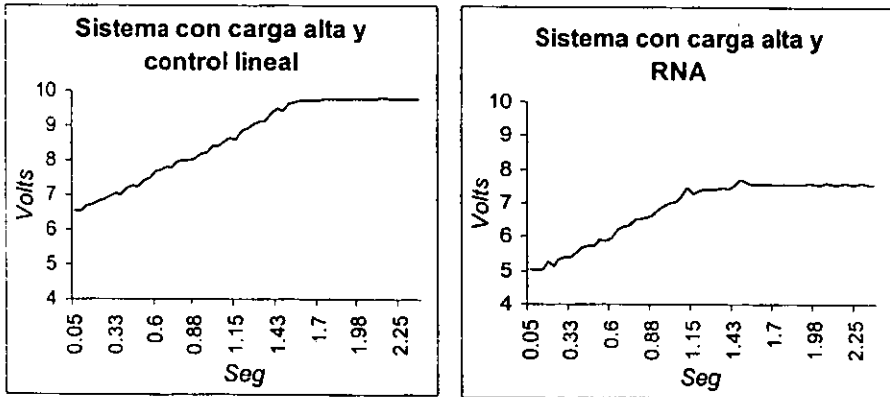
La intensidad de campo magnético sobre el disco metálico del eje lento del motor depende de la posición del imán; para simular una carga baja se coloca el imán en la posición tres de su escala; cuyo fin es solo dividir la colocación del imán en el disco en diez posibles posiciones. Para esta carga relativamente baja, figura 4.11 A, el sistema responde satisfactoriamente ya que llega a la posición indicada pero en un mayor tiempo, debido a la carga aplicada.

Figura 4.11 B. Sistema con carga.



En la figura 4.11 B se aplica al sistema una carga media, con el imán colocado en la posición cinco de su escala (a la mitad), y continua llegando a su posición indicada aunque debido a la carga aumenta el tiempo de asentamiento; sin embargo para el sistema con el control lineal esto le sirve para evitar tener un sobrepaso.

Figura 4.11 C. Sistema con carga.



En la figura 4.11 C tenemos al sistema con una carga denominada alta, con la posición del imán en el numero ocho de su escala. Hasta este tipo de carga el sistema sigue respondiendo adecuadamente ya que aunque aumenta mas su tiempo de asentamiento, por la carga simulada, si llega a

la posición a la que debe llegar. Cabe destacar que el sistema con los dos tipos de control se comporta de una forma similar cuando se aplica una carga al motor.

Al aplicar una carga al motor mayor de la denominada carga alta; este ya no llega a la posición deseada; es decir es demasiada carga y detiene al motor antes de llegar a su posición.

Si no se pasa del límite de carga del motor podemos aprovechar este análisis y si la aplicación lo requiere se puede usar cuando se requiera de un frenado suave del motor o de un movimiento que no necesite de una velocidad rápida o en su defecto si se desea el control lineal sin porcentaje de sobrepaso en la respuesta.

CAPITULO V.

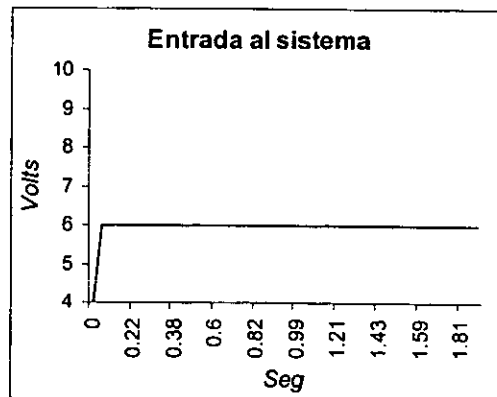
DISCUCION.

5. DISCUSION

Una vez analizadas las curvas de respuesta del sistema con control lineal y con la red neuronal para los mismos valores de entrada y mismas cargas se pueden apreciar algunas diferencias significativas entre ellas. La primera es el tipo de respuesta, ver figuras 4.1 y 4.6; el control tiene una respuesta de segundo orden, como ya se había mencionado anteriormente, con un porcentaje de sobrepaso y un tiempo de asentamiento determinado; al colocar la red neuronal como controlador del sistema éste puede tener una respuesta de segundo orden; pero también puede irse modificando hasta alcanzar una respuesta de algún otro tipo, por ejemplo de un sistema de primer orden con un sobrepaso nulo.

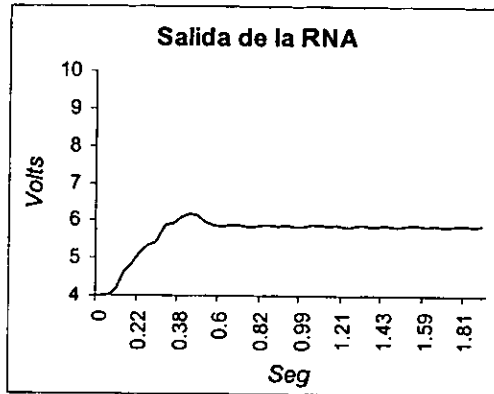
Es importante mencionar aquí que en el sistema hay un sobrepaso nulo debido a que la diferencia de potencial en los transistores de entrada al servoamplificador, salida del controlador (red neuronal), origina una corriente muy pequeña como para causar un movimiento del motor en un sentido o en otro y esto es aprovechado para tener una mayor estabilidad del sistema; al observar la gráfica de salida de la red neuronal tenemos que presenta un pequeño sobrepaso muy rápido pero no se refleja en la salida del sistema por que no es tan grande como para causar un movimiento del motor; este sobrepaso a la salida de la red neuronal se hace rápidamente pequeño hasta ser también nulo; podríamos considerar esta diferencia de potencial como un umbral el cual tiene que ser rebasado para que el eje del motor gire; figura 5.1.

Figura 5.1 A. Entrada al sistema funcionando con la RNA como controlador.



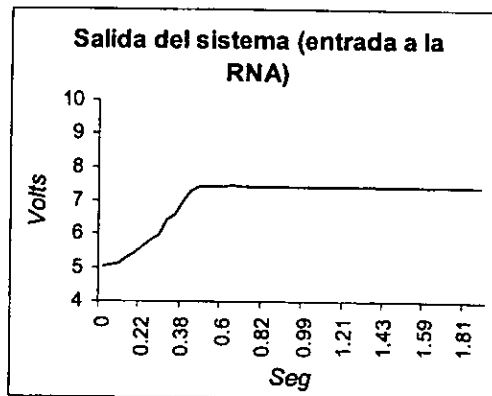
La gráfica anterior indica una entrada escalón aplicada al sistema con la RNA realizando la acción de control.

Figura 5.1 B. Salida de la Red neuronal.



Nótese como a la salida del control (RNA) si existe un pequeño sobrepaso muy "corto"; pero esto no implica que el sistema en sí tenga un sobrepaso. Esta situación se presenta solo con entradas grandes al sistema, entradas escalón de 2 volts o más.

Figura 5.1 C. Salida del sistema con la RNA como control.



En la figura 5.1 C se observa como para la entrada dada, a la salida de la RNA hay un pequeño sobrepaso, pero éste no se ve reflejado a

la salida del sistema por que es relativamente pequeño y en un tiempo corto como para causar un movimiento extra en el motor.

Este hecho representa una ventaja en el entrenamiento de la red neuronal ya que permite mejorar el tiempo de asentamiento del sistema. También ayuda a poder ir cambiando la curva de salida del sistema y lograr que éste responda como el diseñador lo desea. Esto no sucede con los controladores lineales; por ejemplo, con el control de tipo proporcional la salida del controlador es proporcional a su entrada y por lo tanto no puede ser cambiada a algún otro valor pues depende de lo que haya a la entrada y la ganancia del control (k). De igual forma se va deteniendo paulatinamente conforme disminuye la diferencia de potencial en los transistores de entrada del servoamplificador y aquí tanto en la salida del sistema como a la salida del controlador se tiene un sobrepaso y por ende un mayor tiempo de asentamiento. En la figura 4.7 se notan éstas diferencias ya que se presentaron las salidas para los mismos valores de entrada al sistema.

Por otra parte existe diferencia en el valor inicial de las curvas de respuesta de los dos controladores por que así se construyeron los patrones de entrenamiento para la red neuronal y diferenciar así a los controladores; obsérvese que la diferencia es la misma tanto en el valor inicial como en el valor final de las curvas; sin embargo de la misma forma que se cambio éste valor puede hacerse un entrenamiento con patrones de entrenamiento contruoidos para valores de inicio y final iguales a los del control lineal.

Los dos tipos de controladores responden aceptablemente a entradas mínimas de 0.05 volts, que físicamente equivalen a mover el eje lento del motor aproximadamente 10 grados; figura 4.8.

A entradas pequeñas el sistema con RNA presenta buena resolución, ya que solo presenta oscilaciones en algunos pasos, y responde a la mayoría de escalones de 0.05 y 0.1 volts que se le presentan a la entrada. En ocasiones los dos tipos de controladores ignoran alguna entrada pequeña y hasta que se acumula la siguiente entrada responden, esto solo en entradas menores a 0.1 volts. Conforme aumenta el valor de escalones de entrada aumenta la diferencia en el tiempo de asentamiento y sobrepaso y ya en 0.5 volts la diferencia es notable físicamente; si se observan las gráficas de salida se visualizarán estas diferencias.

En cuanto a perturbaciones externas, figura 4.8, los dos controladores responden bien ya que no se salen de estabilidad después de llegar a la posición indicada; cuando se presenta la perturbación y si esta hace que cambie la posición en la que se encuentran tienen la capacidad de regresar a su posición original sin mayor problema. La diferencia básicamente es la misma: el sobrepaso; aunque cuando hay perturbaciones con red neuronal también se presentan algunas oscilaciones inevitables. Por otro lado tenemos que el sistema es más susceptible a perturbaciones externas con el control lineal.

De las gráficas de la figura 4.11 tenemos que las respuestas del sistema con carga son muy parecidas; lo que sucede es que al colocar la carga; esta sirve al controlador lineal como un freno que disminuye el sobrepaso en gran medida, casi haciéndolo nulo; pero aumentando el tiempo de asentamiento. El sistema con la red tiene la misma variación básicamente y es que también presenta un tiempo de asentamiento mayor respecto al sistema sin carga; lo cual puede decirse que es normal en un sistema con carga. Lo que tenemos es que las respuestas del sistema con los dos controladores, lineal y red neuronal, se igualan en gran medida sobre todo al aumentar la carga; la respuesta del sistema con la red neuronal con carga aumenta el tiempo de asentamiento y el control lineal cambia su respuesta con carga disminuyendo al mínimo su sobrepaso y aumentando también su tiempo de asentamiento. Con una carga considerada como alta las dos respuestas son casi iguales hasta en el tiempo de asentamiento.

Es importante mencionar que al aplicar una carga mayor de la soportada al sistema, mayor a la de la figura 4.11 C, tanto con un control como con el otro, no se puede saber hasta donde llegará el motor; si alcanzará la posición indicada, se detiene antes, no se mueve, etc.

CAPITULO VI.

CONCLUSIONES.

6. CONCLUSIONES

Después de la realización de la presente tesis se puede llegar a la conclusión de que el uso de las redes neuronales (RNA) representa otra opción para llevar a cabo la acción de control de sistemas; considerando un menor tiempo de asentamiento y sobrepaso en la respuesta del servomecanismo de corriente directa. Aunque dependerá de la aplicación con la que se trabaje. Con la RNA se logró una respuesta de la planta más rápida y sin sobrepaso.

Ahora bien también se puede concluir que el controlador con RNA es bastante flexible en su diseño; pues una vez entrenada una RNA la respuesta que da la planta con ella puede ser modificada entrenando otra RNA con los patrones anteriores, un poco modificados, y así sucesivamente hasta llegar a una respuesta particular de la planta.

También se comprobó que tanto una RNA puede generalizar las entradas posibles a la planta. En ocasiones no es posible entrenar a una RNA con todos los patrones de entrada que se le pueden presentar; sin embargo la RNA responde de manera satisfactoria a todas las entradas que se le presentan aunque no se le hubieran mostrado en el entrenamiento. Aún a perturbaciones externas y trabajando con carga la RNA logra generalizar bastante bien y la planta muestra una buena respuesta.

Otro aspecto importante es que una RNA no solo puede ser aplicada como un controlador de una planta; pues algo interesante es que también puede hacerse una RNA que actúe como simulador de la misma planta; una RNA sería el controlador y otra RNA sería la planta. Entonces se pueden usar RNA's para uno o varios módulos en algún sistema, con diferentes configuraciones de diagramas de bloques y colocando redes neuronales en los bloques haciendo funciones determinadas.

En general una RNA nos representa otra opción de control que puede no ser la mejor o la peor; esto dependerá completamente de la aplicación que se tenga; por ejemplo si se requiere rapidez de respuesta o algún tipo de respuesta suave, un frenado rápido o lento, con oscilaciones, etc; sin embargo existe la posibilidad de usar la RNA; consideremos la ventaja de que las RNA nos brindan la flexibilidad de ir adaptándolas al sistema para que funcione lo más apegado a lo que el diseñador requiera.

APENDICES.

APENDICE A. TARJETA DE ADQUISICION DE DATOS PCLAB812.

INFORMACION GENERAL

La PCLAB812 es una tarjeta de adquisición de datos de alta velocidad y multifuncional para computadoras IBM y compatibles. Las funciones más importantes con las que cuenta son:

- 16 canales de entradas analógicas.
- Convertidor de entradas analógicas a digitales (HADC574Z); con un máximo rango de muestreo de 30 KHz.
- Dos canales D/A de salida multiplexados de 12 bits.
- Switch de selección de rango de entradas analógicas bipolares: +/-1V, +/-2V, +/-5V, +/-10V.
- Conector de entrada/salida de 20 pines para puertos A/D y I/O.

Convertidor A/D HADC574Z (entradas analógicas)

- Convertidor de aproximaciones sucesivas de dos canales, resolución de 12 bits y diferentes rangos bipolares: +/-1V, +/-2V, +/-5V, +/-10V.
- Velocidad de conversión máxima de 30 KHz y aproximación de 0.015% de lectura +/- 1 bit.
- Disparo por software.
- Transferencia de datos por control de programa.

Convertidor D/A (salidas analógicas)

- Convertidor de dos canales y 12 bits multiplexado.
- Rango de salida de 0 a +5V con referencia fija de -5V. +/-10V con referencia externa de DC o AC.
- +/-5mA de salida máx.

Asignación de pines en los conectores

La PCLAB812 esta equipada con dos conectores de 20 pines que pueden ser conectados al mismo tipo de cable plano en la parte posterior y tres conectores en la tarjeta. Los siguientes diagramas ilustran el asignamiento de pines para cada conector:

Conector 1 (CN1) - Entradas analógicas

A/D	0	1	2	A.GND
A/D	1	3	4	A.GND
A/D	2	5	6	A.GND
A/D	3	7	8	A.GND
A/D	4	9	10	A.GND
A/D	5	11	12	A.GND
A/D	6	13	14	A.GND
A/D	7	15	16	A.GND
A/D	8	17	18	A.GND
A/D	9	19	20	A.GND

Conector 2 (CN2) * Salidas analógicas

A/D *	10	1	2	A.GND
A/D *	11	3	4	A.GND
A/D	12	5	6	A.GND
A/D	13	7	8	A.GND
A/D	14	9	10	A.GND
A/D	15	11	12	A.GND
D/A *	1	13	14	A.GND
D/A *	2	15	16	A.GND
V.REF	1	17	18	A.GND
V.REF	2	19	20	A.GND

El conector 3 (CN3) asigna del pin 0 al pin 15 las salidas digitales (D/O), el 17 y 18 D. GND, el 19 +5V y el pin 20 +12 V.

El conector 4 (CN4) asigna del pin 0 al pin 15 a las entradas digitales (D/I), el 17 y 18 D. GND, el 19 +5V y el pin 20 +12 V.

Conector 5 (CN5) - Contador

EX. TRG	1	2	
DRDY	3	4	
2 Mhz	5	6	CTR1 GATE
	7	8	CTR0 CLK
	9	10	CTR0 OUT
	11	12	CTR0 GATE
	13	14	CTR1 OUT
	15	16	
D.GND	17	18	D.GND
+ 5V	19	20	

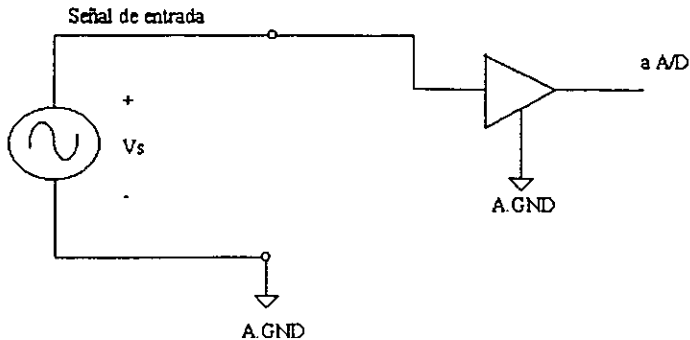
Simbología:

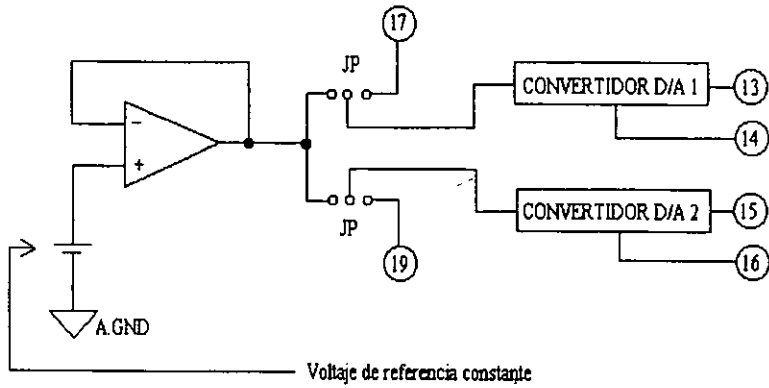
A/D - Entrada analógica
 A.GND - Tierra analógica
 D/A - Salida analógica
 D.GND - Tierra y fuente de energía digital
 CLK - Entrada de reloj
 GATE - Entrada de compuerta
 OUT - Señal de salida
 VREF - Voltaje de referencia

Los pines marcados con * son los utilizados como entradas y salidas en esta tesis.

Conexión de señales

Alambrado estándar para la señal de entrada.





Conexión de la salida analógica.

Simbología:

- 13 - D/A 1 salidas D/A
- 15 - D/A 2 salidas D/A
- 14 - Tierra analógica
- 16 - Tierra analógica
- 17 - V. REF. 1
- 18 - V. REF. 2

Estructura y formato de registros

La PCLAB812 requiere de 16 direcciones consecutivas en espacio para entradas/salidas. Algo importante para la programación de la tarjeta es el significado de éstos registros direccionables desde la dirección del puerto base de entrada/salida seleccionada. El mapa de funciones de cada localidad para cada registro y dirección relativa a la dirección base es:

Localidad	Lectura	Escritura
Base +0	Contador 0	Contador 0
Base +1	Contador 1	Contador 1
Base +2	Contador 2	Contador 2
Base +3	No usado	Control del contador
Base +4	byte menor A/D	CH1 D/A byte menor
Base +5	byte mayor A/D	CH1 D/A byte mayor
Base +6	byte menor D/I	CH2 D/A byte menor

Localidad	Lectura	Escritura
Base +7	byte mayor D/I	CH2 D/A byte mayor
Base +8	No usado	Interrupción CLEAR
Base +9	No usado	No usado
Base +10	No usado	Canal MUX scan
Base +11	No usado	Control de PCLAB812
Base +12	No usado	Disparo A/D software
Base +13	No usado	D/O byte menor
Base +14	No usado	D/O byte mayor
Base +15	No usado	

De los cuales son utilizados:

Registro de datos A/D

1. Datos A/D, byte menor

BASE +4	D7...	D0	
	AD7...		AD0

2. A/D byte mayor

BASE +5	D7...	D5	D4	D3	D2	D1	D0
	0 ...	0	DRDY	AD11	AD10	AD9	AD8

AD11 - AD0: Datos analógicos convertidos a digitales. AD0-LSB, AD11-MSB.
 DRDY: Señal de datos listos. 0 dato listo, 1 dato aún no listo. Una lectura del registro pone el bit en 1 y es puesto a cero cuando una conversión A/D es completada.

Registro multiplexor (MUX scan)

Es solo de escritura y da el número de canal a switchear por el multiplexor. Al escribir este en el registro el nuevo canal es switchado.

BASE +10	D7	D6	D5	D4	D3	D2	D1	D0
Canal	x	x	x	x	CL3	CL2	CL1	CL0

CL3 - CL0: Número de canal.

Registros de salida D/A

Son registros de escritura.

BASE +4	D7	D6	D5	D4	D3	D2	D1	D0
D/A 1 menor	DA7	DA6	DA5	DA4	DA3	DA2	DA1	DA0
BASE +5	D7	D6	D5	D4	D3	D2	D1	D0
D/A 1 mayor	x	x	x	x	DA11	DA10	DA9	DA8
BASE +6	D7	D6	D5	D4	D3	D2	D1	D0
D/A 2 menor	DA7	DA6	DA5	DA4	DA3	DA2	DA1	DA0
BASE +7	D7	D6	D5	D4	D3	D2	D1	D0
D/A 2 mayor	x	x	x	x	DA11	DA10	DA9	DA8

DA11 - DA0: Datos digitales convertidos a analógicos. DA0-LSB, DA11-MSB.

Registro de control

Es un registro solo de escritura y provee la información del modo de operación de la PCLAB812.

BASE +11	D7	D6	D5	D4	D3	D2	D1	D0
Ctrl. Status	x	x	x	x	x	S2	S1	S0

A. Condición de disparo interno

S2	S1	S0	
0	0	0	Software deshabilitado y disparo por pulso.
0	0	1	Solo disparo por software habilitado.
0	1	0	Disparo por pulso usando transferencia DMA.
1	1	0	Disparo por pulso con transferencia por programa o por interrupción.

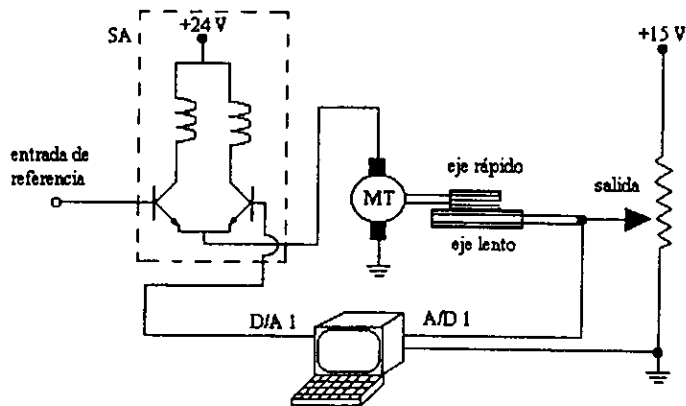
B. Condición de fuente de disparo externa

S2	S1	S0	
0	0	x	Solamente disparo externo habilitado.
0	1	0	Disparo externo con transferencia DMA.
1	1	0	Transferencia de datos por programa o interrupción usando fuente de disparo externa.

APENDICE B. PROGRAMA SERVO

Para poder controlar el servomotor de c.d. con la pc se usa el programa servo; esto independientemente del controlador que se quiera utilizar, ya que en el programa se puede usar un controlador proporcional discreto o una red neuronal artificial (RNA); aún mas, puede utilizarse el programa solo para obtener lecturas o archivos de algún punto de la planta teniendo ésta un control lineal fuera de la pc.

Antes de correr el programa se debe tener la conexión del sistema según la aplicación requerida. Para llevar el control con la pc se hace la siguiente conexión de módulos:



donde el canal A/D1 de la tarjeta pclab812 transforma la salida de la planta para usarla como entrada al controlador y el canal D/A1 de la tarjeta envía la salida del controlador hacia el módulo servoamplificador. Realizando la acción de control en la pc. Como la tarjeta de adquisición de datos pclab812 conectada en la pc tiene dos canales A/D y dos D/A puede usarse el canal D/A2 para dar una entrada de referencia más exacta o mejor conocida solo con conectar esa entrada al canal de salida de la tarjeta D/A2; entonces la entrada puede darse desde la pc.

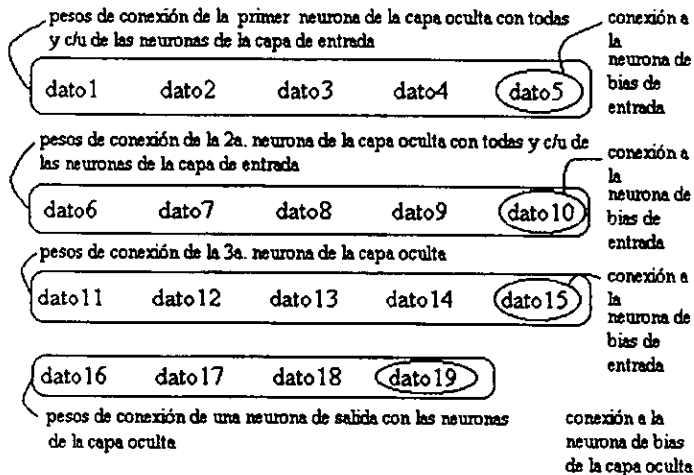
Al tener ya la planta lista se puede correr el programa servo; el cual despliega un menú principal con las siguientes opciones:

- Controlar la planta con la RNA.
- Controlar la planta con un control discreto.

Ver gráficas después de una acción de control.
 Guardar datos en un archivo.
 Especificar archivo de pesos para la RNA.
 Ayuda rápida.

Control con RNA

Para controlar la planta con una RNA primero se debe de especificar al programa el nombre del archivo de pesos de conexión que la RNA utilizará. El archivo de pesos debe estar ya construido con el siguiente formato:



La RNA tiene 4 neuronas de entrada, 3 neuronas en la capa oculta y dos de salida; sin embargo solo se utiliza el valor de una neurona de salida, es por eso que solo se tienen 19 pesos de conexión para la RNA.

Ya especificado el archivo de pesos se elige la opción "R" y aparece la pantalla para el control con RNA.

Control discreto

Este controlador discreto es un control proporcional que se efectúa mediante la ecuación $u=y/k$; donde u es la salida del control, y es la entrada e éste y k es una constante de proporcionalidad de 1.4303.

Para entrar a la pantalla de control solo se elige la opción "D" en el menú principal y no requiere de ningún archivo adicional.

Gráficas de control

En esta opción se puede visualizar en pantalla la gráfica de respuesta del sistema o de salida del controlador después de una acción de control.

Para ver las gráficas se tienen que guardar los datos en un archivo, (con F1). Este archivo contiene datos de tiempo, entrada al control y de salida de él.

Cuando se desea ver una gráfica se elige la opción "V" y aparece un submenú para escoger cual curva visualizar: la de salida de la planta (entrada al controlador) o la de salida del control y enseguida aparece la pantalla con la gráfica correspondiente.

Guardar datos en un archivo

Si se desea guardar los datos de alguna acción de control se elige la opción "G" y posteriormente se da el nombre del archivo en el cual se desean guardar los datos (nombre de archivo con un máximo de ocho caracteres). Previamente estos datos se guardaron en un archivo temporal (con F1).

Especificar archivo de pesos para la RNA

Para poder entrar al control con RNA se debe primero especificar el archivo de pesos de conexión entre las neuronas; este archivo debe estar ya hecho para cada RNA que se desee utilizar como control (cambiándolo las veces que se quieran) y se da con la opción "P".

Ayuda rápida

La opción de ayuda rápida es una guía dentro del programa para las teclas claves que se utilizan en el programa servo. Se entra con la opción "A" y enseguida aparece la pantalla de ayuda. Las teclas importantes para el uso del programa son:

Esc - Regresa al menú principal desde la pantalla en que se encuentre. También se usa para detener la escritura de datos en la opción de control con RNA o con control discreto.

Teclas de números 1 a 6 - Aumentan a la entrada actual de la planta un escalón equivalente al número oprimido.

Inicio - Envía a la planta el voltaje correspondiente para poner al motor en su posición de inicio.

Número 0 - En los controles discreto y con RNA hay posibilidad de dar entradas a la planta de voltajes diferentes a números enteros, mediante incrementos o decrementos. Cuando se desea dar entrada de éstos voltajes solo se ajusta el valor y éste valor se dará a la planta con la tecla del número 0.

Re pág - Inicialmente el decremento de entrada a la planta esta en cero; para disminuir en intervalos de 0.05 volts la entrada se oprime la tecla Re pág y este valor cambiara hasta ajustar al valor deseado.

Av pág - Para incrementar la entrada a la planta, inicialmente también esta en cero pero va aumentando en intervalos de 0.05 volts y puede ajustarse a un valor determinado.

Uarrow - El programa trabaja con una frecuencia de muestreo en la tarjeta de adquisición de datos por defecto con la cual tiene un funcionamiento óptimo; sin embargo ésta puede cambiarse y se aumenta con ésta tecla. No es recomendable cambiarla a menos que la aplicación lo requiera.

Darrow - Para disminuir la frecuencia de muestreo con la cual el programa trabaja por defecto. Se cambia según sea la aplicación pero se recomienda no cambiar el valor de inicio.

F1 - Cuando se esta haciendo una acción de control y se quieren guardar los datos se usa la tecla F1; esta tecla prepara al programa para guardar los datos en el archivo temporal "archdat.s" de donde son leídos para la graficación o para guardar los datos en un archivo especificado por el usuario. Una vez apretada la tecla F1 el programa solo espera el escalón de entrada a la planta para iniciar el guardado de datos. Pueden darse varias entradas y los datos son guardados en "archdat.s".

Cada vez que se teclee F1 y se da la entrada al sistema el archivo archdat.s es sobrescrito con los nuevos datos; por lo cual es importante que si se desean conservar los datos anteriores se deben guardar en un archivo especificado por el usuario. Para detener la escritura de los datos después de oprimir F1 y las entradas correspondientes se oprime ESC y el control del programa regresara al menú principal.

BIBLIOGRAFIA.

- *Control Systems Theory.*
Elgerd. Olle I.
McGraw Hill.

- *Neural Networks. A comprehensive Foundation.*
Haykin, Simon.
Macmillan Publishing Company.

- *Neural Networks. Algorithms, Applications and Programming Techniques.*
Freeman, James A., Skapura David M.
Addison-Wesley Publishing Company.

- Astorga de Riquer, Agustin de Jesús.
Cano García, José Luis.
Control de un mobil mecánico mediante RN empleando el algoritmo de retropropagación del error.
Tesis de Licenciatura. UNAM. México 1997.

- *Dynamind User's Developers Guide.*
NDX Manual.