

26
25/Jan

**UNIVERSIDAD NACIONAL
AUTÓNOMA DE MÉXICO**

FACULTAD DE QUÍMICA



**ALGORITMOS NO LINEALES DE CONTROL
DE PROCESOS VÍA REDES NEURALES**

T E S I S

**QUE PARA OBTENER EL TÍTULO DE
INGENIERO QUÍMICO PRESENTA**

JORGE RUBÉN DURÁN LIZARRAGA

MÉXICO D.F.

1994

**TESIS CON
FALLA DE ORIGEN**



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 está 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.



Jurado asignado según el tema:

EXAMENES PROFESIONALES.
FAC. DE QUÍMICA

PRESIDENTE	PROF. CÉSAR ALEJANDRO RINCÓN ORTA
VOCAL	PROF. EUGENIO LEON FAUTSCH TAPIA
SECRETARIO	PROF. ALBERTO BREMAUNTZ MICHAVILA
1 ^{ER} SUPLENTE	PROF. MANUEL JESÚS AGUILAR GÓMEZ
2 ^{DO} SUPLENTE	PROF. CELESTINO MONTIEL MALDONADO

SITIO DESARROLLO EL TEMA:

SEN DEN S.A. DE C.V.
Calle Privada el Relox No. 30
Chimalistac 01070
México D.F.

ASESOR:


I.F. ALBERTO BREMAUNTZ MICHAVILA

SUSTENTANTE:


JORGE RUBÉN DURÁN LIZARRAGA

*a mi madre y a mi padre,
que me han dado todo y mas*

*a María Elena,
David,
Juan Carlos y Andrea
y Marina*

a mi Universidad

*a Angélica,
Luis
y Mauricio,
que bien saben porqué*

AGRADECIMIENTOS

**Agradezco a Alberto Bremauntz Michavila por
su amistad y apoyo para la realización de la
presente tesis**

**Gracias también a todos mis amigos, porque en
esto hay mucho de ellos**

Y muchas gracias a Elena Gottdiener

INDICE

	<u>Página</u>
CAPÍTULO 1	
EL PROBLEMA FUNDAMENTAL DEL CONTROL	
1.1. La importancia del control	1-1
1.2. El problema fundamental del control	1-2
1.2.1. Criterios de desempeño	1-4
1.2.2. Representación del proceso	1-5
1.3. Soluciones al problema fundamental del control. Redes Neuronales	1-7
CAPÍTULO 2	
CONTROL ALGORÍTMICO DE PROCESOS	
2.1. Control proporcional, integral y derivativo PID	2-1
2.1.1. Control proporcional	2-1
2.1.2. Control integral	2-4
2.1.3. Control derivativo	2-6
2.1.4. Deducción matemática de la ecuación del PID	2-7
2.1.5. Utilidad del PID	2-11
2.2. Control dinámico matricial DMC	2-12
2.2.1. Modelos de respuesta a entradas escalón	2-12
2.2.2. Algoritmo del control dinámico matricial	2-16
CAPÍTULO 3	
REDES NEURALES	
3.1. Introducción	3-1
3.1.1. Analogía con el cerebro	3-1
3.1.2. Redes Neuronales	3-2
3.2. Operación de Redes Neuronales	3-4
3.2.1. Aprendizaje	3-4
3.2.2. Recapitulación	3-4
3.2.3. Operación en estratos	3-5
3.3. Características de la neurocomputación	3-6

3.3.1	Aprendizaje por ejemplos	3-6
3.3.2	Memoria distribuida asociativa	3-7
3.3.3	Tolerancia a fallas	3-8
3.3.4	Reconocimiento de patrones	3-8
3.3.5	Síntesis	3-9
3.4.	La técnica de Back-Propagation o "propagación hacia atrás"	3-10
3.4.1.	Redes con Back-Propagation	3-10
3.4.2.	Propagación hacia atrás del error local	3-12
3.4.3.	Minimizando el error global	3-13
3.4.4.	La función de error global	3-14
3.4.5.	Algoritmo de Back-Propagation	3-16

CAPITULO 4

REDES NEURALES COMO ALTERNATIVA DE CONTROL DE PROCESOS

4.1.	Introducción	4-1
4.2.	Modelos para la identificación de sistemas dinámicos no lineales	4-2
4.2.1.	Modelo I	4-2
4.2.2.	Modelo II	4-3
4.2.3.	Modelo III	4-3
4.2.4.	Modelo IV	4-5
4.3.	Procedimiento de identificación	4-6
4.4.	Excitación persistente	4-7
4.5.	Control adaptivo	4-8
4.5.1.	Control directo	4-8
4.5.2.	Control indirecto	4-9
4.5.3.	Control adaptivo en sistemas dinámicos no lineales	4-10
4.6	Redes Neuronales como controladores adaptivos	4-12

CAPITULO 5

DESARROLLO EXPERIMENTAL

5.1.	Introducción	5-1
5.1.1.	Planteamiento	5-1
5.1.2.	Objetivos	5-1
5.1.3.	Metodología general	5-2
5.2.	Preliminares	5-4
5.2.1.	Procesos dinámicos	5-4
5.2.1.1.	Procesos de primer orden	5-4
5.2.1.2	Procesos de segundo orden	5-7
5.2.1.3.	Procesos con tiempo muerto	5-9
5.2.2.	Controlador PID	5-11
5.2.3.	Conversión de funciones continuas en el dominio de la frecuencia a funciones discretas en el dominio del tiempo:	5-12
5.2.3.1.	Procesos de primer orden	5-13
5.2.3.2	Procesos de segundo orden	5-13
5.2.3.3.	Procesos con tiempo muerto	5-14
5.2.3.4.	Controlador PID	5-14
5.2.4.	Sintonización del controlador y variaciones efectuadas a la entrada	5-15
5.2.4.1.	Sintonización	5-15
5.2.4.2.	Cambios efectuados en la entrada	5-16
5.2.5.	Red Neural	5-23
5.3.	Entrenamiento	5-25
5.4.	Pruebas	5-29

5.4.1.	Recapitulación y análisis de robustez	5-29
5.4.1.1.	Red 1	5-29
5.4.1.2.	Red 2	5-30
5.4.1.3.	Red 3	5-30
5.4.1.4.	Red 4	5-30
5.4.2.	Actuación de las redes neurales como controladores	5-38
5.5.	Análisis de resultados	5-44
5.5.1.	Pruebas de recapitulación	5-44
5.5.2.	Redes neurales como controladores	5-50

CAPITULO 6
CONCLUSIONES

CAPÍTULO 1

EL PROBLEMA FUNDAMENTAL DEL CONTROL

1. EL PROBLEMA FUNDAMENTAL DEL CONTROL

1.1. LA IMPORTANCIA DEL CONTROL

La industria de proceso siempre se ha caracterizado por sus dinámicas y a menudo impredecibles condiciones de mercado. Estas condiciones conllevan a la necesidad de que los procesos trabajen con mayores rangos de operación. En el pasado, estas fluctuaciones en los mercados internacionales frecuentemente daban lugar a nuevos diseños de plantas de proceso, o por lo menos a importantes modificaciones en plantas existentes.

En la actualidad, estas variaciones son más grandes que nunca. Por ello resulta ahora menos atractiva la alternativa arriba presentada, ya que debido a las dimensiones y velocidades de los cambios en los mercados, la rentabilidad de los nuevos diseños no es segura.

Debido a esto, resulta mucho más económico aprovechar los diseños existentes, desarrollando nuevas técnicas que permitan extraer las mayores utilidades con el mínimo de inversión.

La manera más efectiva y económica de responder a los cambios mencionados es desarrollando un sistema tecnológico que integre todos los aspectos de la automatización del proceso, automatización que permite que empresas y compañías reaccionen ante la diversidad de cambios en los mercados. Estos aspectos pueden ser presentados de la siguiente manera: Medición, Control, Optimización y Logística.

Elemento fundamental de esta integración de tecnologías constituyen los sistemas de control.

Anteriormente, el único objetivo de los sistemas de control era el mantener a los procesos en condiciones de operación estables. Sin embargo, los requerimientos de la industria han cambiado, al igual que los mismos procesos. Éstos, como se menciona arriba, se han vuelto cada vez más complicados, aumentando en diversidad, en capacidad y en número de equipos involucrados.

Hoy en día, el mantener una producción constante y estable ha dejado de ser lo único importante. La actual evolución del mercado, la creciente competencia, la situación económica mundial y las cada vez más estrictas reglamentaciones en materia de protección ambiental han provocado que los objetivos de producción cambien, de manera tal que ahora no solo interesa el producir, sino también el producir óptimamente, con la mayor calidad posible, con la mínima desviación de especificaciones, con el menor desperdicio de materia y energía y contaminando lo menos posible.

Este nuevo enfoque impone condiciones adicionales: Los sistemas de control ya no deben de responder únicamente a los requerimientos normales del proceso, sino que ahora deben actuar en función de decisiones logísticas tomadas en niveles gerenciales. Es decir, los objetivos corporativos deben de ser traducidos a los objetivos de los sistemas de control.

1.2 EL PROBLEMA FUNDAMENTAL DEL CONTROL

En el área del control de procesos existe la necesidad de minimizar los costos tanto de diseño como de mantenimiento. Debido a que cada proceso posee características que lo hacen único, no se pueden aplicar las mismas técnicas y criterios a todos, sino que cada uno requiere de atención y estudios especiales.

Lo anterior significa que el diseñar un sistema de control que aplique a todos los casos tendría un costo excesivo, debido a que tendría que cubrir una inmensa cantidad de posibilidades. Por otro lado, el hecho de diseñarlo de una manera individual hasta sus últimos detalles no resultaría rentable, ya que sería demasiada inversión y únicamente aplicaría al proceso para el que fue diseñado.

De acuerdo con Prett y García¹, es necesario proporcionar un enfoque unificado y sistemático al control de procesos, con el fin de obtener esquemas generales para el tratamiento de cada problema, resultando de esto una reducción significativa en los costos. Un enfoque de este tipo proporcionaría un marco teórico general no solo para el control de procesos, sino para cualquier tipo de problema de control.

Este enfoque unificado es conveniente también en el sentido de que a través de él se logran diseños eficientes de sistemas de control que permiten llevar a cabo la operación de una planta de proceso con poco personal, por lo que los costos se reducen de una manera considerable.

Ahora bien, es conveniente señalar que un enfoque unificado como el presente no significa un enfoque simplificado. En control de procesos es común pensar que mientras más sencillo es un sistema de control será más fácil su aplicación. En realidad, simplificaciones de este tipo dan como resultado, a la postre, mayores costos de mantenimiento y diseño, debido a que los esquemas simples generalmente no resuelven el problema, por lo que hay que efectuar correcciones adicionales.

En este contexto, el Problema Fundamental del Control puede ser definido, de acuerdo nuevamente a Prett y García, de la siguiente manera:

*"Diseñar un sistema de control que manipule el proceso con el fin de que se cumplan los objetivos corporativos, los cuales se cumplen a través de los objetivos de operación, de los cuales a su vez se desprenden los objetivos de control, bajo las restricciones particulares de cada proceso, utilizando para ello una descripción del proceso a controlar y considerando la incertidumbre intrínseca de esta descripción"*².

Elementos

El problema fundamental del control consta de cinco elementos, los cuales son presentados de la siguiente manera:

(1,2) Prett, D.M., García, C. FUNDAMENTAL PROCESS CONTROL
Butterworths Series in Chemical Engineering, E.U.A., 1988

Crterios de Desempeño

- Objetivos
- Restricciones de igualdad
- Restricciones de desigualdad.

Representación del Proceso

- Modelos matemáticos
- Descripción de incertidumbre

1.2.1 CRITERIOS DE DESEMPEÑO

Un sistema de control deberá de satisfacer ciertos criterios de desempeño, como son:

- Económicos
- Seguridad y protección ambiental
- Limitaciones en cuestiones de equipo
- Calidad de producción

Para que el sistema de control tenga éxito, todos estos criterios pueden ser traducidos a las expresiones matemáticas que permitirán su diseño. Para ello, se requiere que toda variable necesaria para evaluar los criterios anteriores pueda ser o bien medida directamente, o bien inferida a través de otras mediciones. Así, cualquiera de los cuatro criterios anteriores puede ser expresado matemáticamente de alguna de estas tres formas:

- **Objetivos**

Optimización dinámica de las funciones de variables cuyo estado óptimo significa el satisfacer el criterio.

- **Restricciones de Igualdad**

Manipular la variable hasta que ésta alcance el valor deseado.

- **Restricciones de Desigualdad**

Manipular la variable a fin de evitar que ésta llegue a valores no deseados.

Las diferentes metodologías en control de procesos se encuentran basadas en suposiciones acerca de la funcionalidad de objetivos y restricciones, a fin de simplificar la solución del problema, logrando así dos cosas: simplificar las matemáticas del modelado del sistema y la implementación del hardware existente.

1.2.2 REPRESENTACIÓN DEL PROCESO

De manera particular, los procesos químicos e industriales son invariablemente descritos por modelos construidos a partir de ecuaciones basadas en termodinámica, fisicoquímica, cinética química, etc. En realidad, estos modelos se han ido complicando a medida que los procesos mismos han ido evolucionado, por lo que resultan difíciles de ser formulados. Esta creciente complejidad para proponer modelos confiables se debe principalmente a que los parámetros físicos (que constituyen la base de todo modelo de este tipo) no se identifican fácilmente, y a que las materias primas con frecuencia no se encuentran bien caracterizadas. Así, encontramos que los modelos no solo no son lineales, sino que además sus parámetros no son del todo confiables.

El control de un proceso se basa precisamente en el modelo de dicho proceso. Tradicionalmente, los modelos empleados han sido lineales, ya que por su misma naturaleza dan lugar a controladores más fáciles de desarrollar, a pesar de que siempre existe una desviación entre la predicción del modelo y el comportamiento real del proceso, desviación acentuada por el hecho de utilizar modelos lineales. Sin embargo, no obstante el modelo utilizado, siempre se presentará esta diferencia, que puede deberse a cualquiera de estas causas:

- El modelo no describe al proceso cuando el punto de operación cambia y se mueve fuera de los límites del rango para el cual los parámetros fueron identificados.
- El equipo cambia o se degrada (incrustaciones en intercambiadores de calor, degradación del catalizador, etc.)
- Las características de las perturbaciones no se conocen, por lo que en la etapa de diseño se realizan consideraciones incorrectas respecto al comportamiento dinámico real del proceso.

- Las técnicas utilizadas para identificar los parámetros no se aplican correctamente, o no son lo suficientemente precisas.

Los errores en el modelado constituyen un punto muy importante, debido a que por inexactitudes significativas frecuentemente el sistema de control no puede cumplir con sus objetivos, teniendo entonces un desempeño deficiente.

Por otro lado, siempre existirán parámetros no conocidos del proceso que deben de ser incluidos en el diseño del modelo. A esto se le conoce como "descripción de la incertidumbre", que puede ser implementada mediante parámetros de rangos tales que representen a todos los valores posibles en las regiones de operación, o bien mediante la inclusión de datos estadísticos sobre las perturbaciones estocásticas del proceso.

El incluir descripciones de incertidumbre en el modelo permite el análisis y diseño de sistemas de control que trabajen en rangos de operación más amplios, resultando de éstos diseños más robustos.

Cada uno de estos elementos debe poder cambiar durante la actividad del sistema de control: por un lado, los criterios de desempeño pueden cambiar debido a decisiones provenientes de niveles gerenciales; por otro lado, las condiciones de operación pueden cambiar hasta el punto de caer en regiones de comportamiento no contempladas por el modelo del proceso.

Si esto último ocurre, puede ser necesario modificar el diseño original en línea. La rama del control que se ocupa de estos cambios es el Control Adaptivo.

Aunque no necesariamente se llega a algo tan complicado. Siguiendo un procedimiento orientado hacia la resolución del problema fundamental del control, puede ser posible llegar a un controlador sencillo. Pero entonces, se habrá llegado a esa conclusión considerando todos los criterios de desempeño y los factores de incertidumbre. Lo realmente importante es el grado en el cual todos los elementos del problema fundamental fueron considerados en el diseño de su solución.

1.3 SOLUCIONES AL PROBLEMA FUNDAMENTAL DEL CONTROL. REDES NEURALES

Todas las actuales técnicas de control de procesos constituyen alternativas de solución al problema fundamental del control. En este punto pueden ser mencionadas dos de las técnicas importantes: El controlador PID, controlador de acción proporcional, integral y derivativa, y el Control Dinámico Matricial.

El controlador PID, si bien no constituye una solución directa al Problema Fundamental del Control, ha demostrado ser de mucha utilidad en un gran número de aplicaciones, debido a su funcionalidad y relativa sencillez, hasta el punto de ser el algoritmo más popular durante varias décadas. Por otro lado, el Control Dinámico Matricial DMC (Dynamic Matrix Control) es una técnica del llamado Control Predictivo, basada en la resolución del problema de optimización del error de los controladores. Así, el DMC, como parte del Control Predictivo, constituye una solución al Problema Fundamental del Control, ya que no solo aplica tanto a procesos univariados como a procesos multivariados, sino porque incluye la formulación de una función de optimización que permite minimizar funciones económicas de costos. Tanto el PID como el DMC se presentarán con mayor detalle en el siguiente capítulo.

Continuamente se llevan a cabo nuevos avances en los diferentes sistemas de control, aplicándose a ellos nuevas tecnologías. Un ejemplo de ello es la cada vez más frecuente utilización de Sistemas Expertos, técnica de Inteligencia Artificial que con éxito ha resuelto algunos de los problemas del control de procesos. Ahora bien, en los últimos años se ha presentado un desarrollo importante en un área de la Inteligencia Artificial llamada Redes Neuronales, cuyas técnicas han probado ser efectivas en la resolución de diversos problemas, tales como reconocimiento de imágenes, procesamiento de lenguajes, etc. Las Redes Neuronales constan de un modelo matemático singular, que las hace idóneas para ciertas áreas en donde el reconocimiento de patrones es importante.

En la presente tesis se pretende demostrar que una solución alternativa al Problema Fundamental del Control puede ser lograda a través del uso de sistemas de control basados en Redes Neuronales.

CAPÍTULO 2

CONTROL ALGORÍTMICO DE PROCESOS

2. CONTROL ALGORÍTMICO DE PROCESOS

En el presente capítulo se presentarán dos soluciones alternativas al Problema Fundamental del Control: el control PID y el control DMC.

2.1. CONTROL PROPORCIONAL, INTEGRAL Y DERIVATIVO PID

El control PID constituye una de las técnicas más utilizadas en el control de procesos. En realidad es una conjunción de tres algoritmos matemáticos: el proporcional, el integral y el derivativo, cuyos porcentajes de actuación varían dependiendo de las condiciones particulares del proceso a controlar.

Para cada proceso pueden existir diferentes combinaciones de control PID. Siempre se requiere algo de control básico proporcional, aunque para algunos casos sencillos resulta suficiente. Otros controladores combinan ya sea la acción integral o derivativa con la proporcional. Y existen también controladores con las tres acciones, por lo que en general son descritos como controladores P, PI, PD y PID. Las contribuciones relativas de cada acción son determinadas por los coeficientes de la llamada ecuación universal para un controlador ideal de tres términos. A continuación se describirán cada una de las acciones mencionadas, y la derivación matemática del algoritmo completo.

2.1.1. CONTROL PROPORCIONAL

El control proporcional surgió de la necesidad de contar con un mecanismo de control que permitiera ejecutar acciones destinadas a efectuar correcciones continuas para que la variable por controlar o variable de proceso alcanzara el valor deseado, llamado punto de ajuste. Esta acción se calcula a partir del error existente entre el punto de ajuste y la variable de proceso, lográndose así que en cualquier instante la acción del controlador sea proporcional al error. (A lo largo de la presente tesis se designará a la variable de proceso como PV y al punto de ajuste como SP por las siglas en inglés de *process variable* y *set point*.)

Por ejemplo, un controlador con algoritmo de acción proporcional puede manipular una válvula a fin de que ésta regule el flujo de combustible a una caldera para que la temperatura de salida sea la deseada (ver figura 2.1).

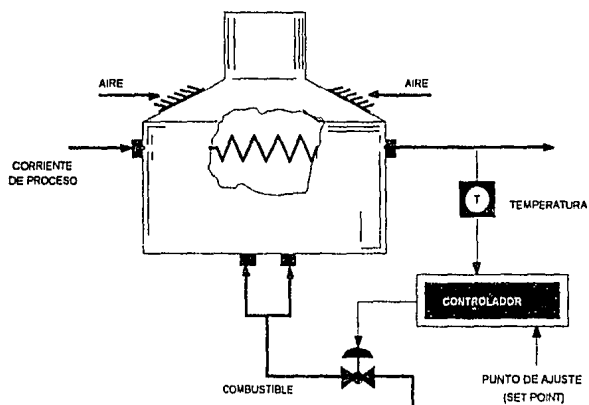


FIGURA 2.1
CONTROL DE CALDERA

Un controlador proporcional pretende mantener estable la temperatura de salida de la caldera, pese a las perturbaciones del proceso. Por cada grado que la temperatura sobrepase el punto de ajuste SP, el controlador reduce la abertura de la válvula en una cantidad específica; igualmente, cuando la temperatura es demasiado baja, el controlador aumentará la abertura de la válvula un porcentaje proporcional a la magnitud del error. La constante de proporcionalidad, que es la que establece que tanto se mueve la válvula en respuesta a una determinada desviación de temperatura, es llamada la ganancia del controlador (figura 2.2).

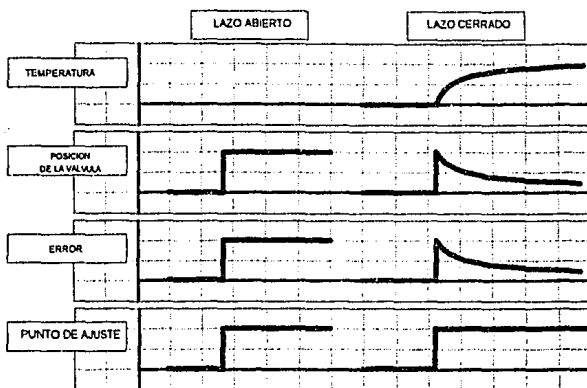


FIGURA 2.2
CONTROL PROPORCIONAL

El algoritmo de acción proporcional permite que el controlador proporcione una respuesta gradual ante una perturbación. Los elementos reguladores de las válvulas son ajustadas por medio de rangos continuos de valores. Así, si la temperatura difiere solo ligeramente del punto de ajuste, la corrección a la posición inicial de la válvula será mínima, mientras que para errores más grandes la corrección será mayor.

No obstante su utilidad, el control proporcional presenta un serio problema: cada valor posible de error corresponde a una posición específica de la válvula, por lo que solo una posición de ésta corresponde a un valor de error de cero. Por lo tanto, un controlador proporcional mantendrá la variable de proceso en el punto de ajuste solo bajo un conjunto determinado y específico de circunstancias, como pueden ser aquellas bajo las cuales se determinó la posición inicial de la válvula. En cualquier otro caso, bajo condiciones diferentes, el control proporcional tiende a reducir el error, pero no lo elimina.

El error residual en un controlador de este tipo puede ser disminuido incrementando la ganancia. Entonces, si el controlador efectúa una respuesta más drástica ante cada pequeña desviación de temperatura, la variable de proceso se mantendrá más cerca del punto de ajuste.

Por otro lado, existen límites que definen que tanto puede ser incrementada la ganancia. Ganancias muy altas pueden conducir a inestabilidad, debido a que cuando el controlador sobrecompensa el error, da origen a otro error en el sentido opuesto, que también produce una sobrecompensación, y así sucesivamente.

Debido a esta limitación, un proceso controlado únicamente por un algoritmo proporcional debe de ser ajustado manualmente cada vez que una perturbación desvía de manera considerable la variable de proceso del punto de ajuste.

2.1.2. CONTROL INTEGRAL

Otra falla importante del control proporcional es que carece del sentido del tiempo. El controlador basa sus cálculos en el estado presente de la variable de proceso, pero no "recuerda" ningún valor anterior y no realiza ningún esfuerzo por anticipar valores futuros. De manera similar, ninguna parte del sistema retiene archivos de los diferentes puntos de ajuste utilizados.

Es claro que el registro de eventos es muy importante en la operación de un sistema de control. Ningún evento en el proceso o en el controlador ocurre instantáneamente. Por ejemplo, en el caso mencionado de la caldera, cuando la temperatura de salida no es la deseada, existe un breve retraso antes de que el sensor registre el cambio; otro retraso se presenta debido al tiempo que tarda el controlador en determinar la respuesta apropiada y modifique la posición de la válvula. Por otro lado, los retrasos del proceso son típicamente más largos que los provocados por el controlador: pueden pasar varios minutos antes de que el incremento de flujo de combustible produzca un cambio apreciable en la temperatura de la corriente de proceso.

La presencia de retrasos de este tipo complican de manera importante el desempeño del controlador. Un controlador proporcional puede muy bien detectar una desviación del punto de ajuste y calcular la acción correctiva, pero para el momento en que esta acción se aplique, el proceso muy probablemente habrá cambiado. Si los cambios en la variable de proceso son mucho más rápidos que las respuestas del controlador, las acciones correctivas resultarán completamente inapropiadas, pudiendo llevar al sistema aún más lejos del punto de ajuste.

De lo anterior se deduce que es necesario incorporar un factor que tome en cuenta el paso del tiempo y que permita efectuar correcciones basadas en valores pasados y futuros. La forma de realizar esto es calculando dos funciones del error: su integral en el tiempo y su derivada con respecto al tiempo. Así el controlador puede combinar las tres acciones de control: proporcional, integral y derivativa.

La principal función del control integral es asegurar que el valor de la variable de proceso sea igual al del punto de ajuste. Su naturaleza puede ser demostrada considerando un cambio súbito en el punto de ajuste, como es una entrada escalón (ver figura 2.3), lo cual da lugar a la correspondiente variación escalón en el error. Si el controlador únicamente cuenta con acción proporcional, producirá un cambio inmediato en la posición de la válvula, pero este cambio no será suficiente para eliminar el error.

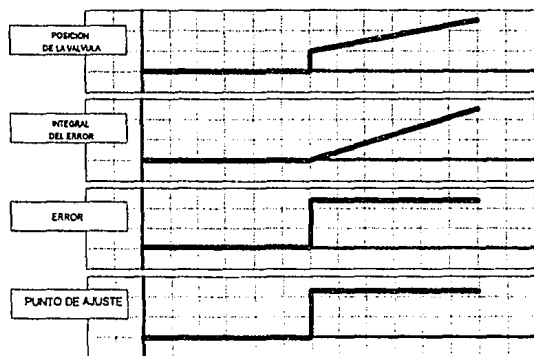


FIGURA 2.3
CONTROL INTEGRAL

El control integral permite una corrección aún mejor: la integral en el tiempo de una variable es la suma continua de sus valores a lo largo de un intervalo de tiempo dado, por lo que la integral del error del sistema de control aumenta a medida que el error instantáneo difiere de cero. Incluso una pequeña desviación del punto de ajuste, que puede no provocar una respuesta del control proporcional, dará como consecuencia una señal de error significativa cuando es integrada sobre algún intervalo de tiempo lo suficientemente largo.

Es decir, mientras que con el control proporcional es necesario que se presente un error para que el controlador responda, con el control integral un pequeño error positivo siempre provocará un incremento en la señal de control, mientras que un error negativo originará un decremento en la señal de control, sin importar que tan pequeño sea dicho error.

2.1.3. CONTROL DERIVATIVO

El objetivo de la acción derivativa es mantener e incrementar la estabilidad del sistema de control. Si se aplica una entrada tipo rampa al controlador, la acción proporcional del mismo responderá de igual manera, como ya se ha mencionado, modificando la posición de la válvula a fin de compensar la variación de la entrada.

Si el error es constante, su derivada con respecto al tiempo es cero, y no se realiza ningún cambio en la posición de la válvula. Una función tipo rampa produce un cambio en el error con rapidez constante, con lo cual el parámetro derivativo tiene una constante y una salida diferente de cero. Ver figura 2.4.

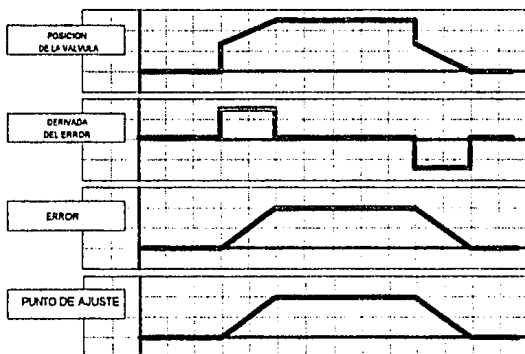


FIGURA 2.4
CONTROL DERIVATIVO

Una característica importante de esta acción es que alcanza su amplitud completa tan pronto como la entrada al controlador empieza a cambiar, por lo que esta parte del algoritmo tiende a anticipar tendencias en la variable de proceso (ver figura 2.5), acelerando por lo tanto la respuesta del controlador ante diferentes perturbaciones.

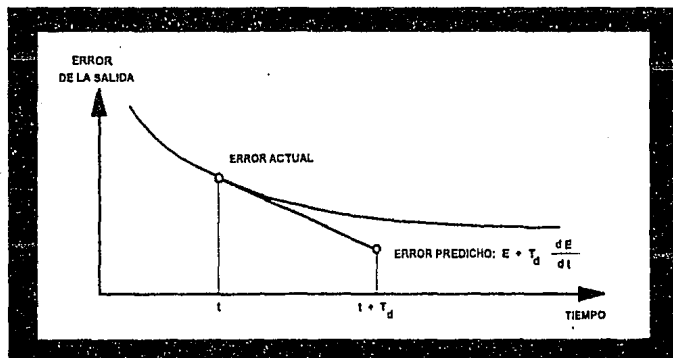


FIGURA 2.5
INTERPRETACIÓN DE LA ACCIÓN DERIVATIVA COMO CONTROL PREDICTIVO

2.1.4. DEDUCCIÓN MATEMÁTICA DE LA ECUACIÓN DEL PID

Es común analizar un sistema de control en el "dominio de la frecuencia", donde todas las cantidades se encuentran expresadas como funciones de la frecuencia de oscilación de los sistemas.

Los puntos de vista de frecuencia y tiempo se encuentran muy relacionados entre sí, ya que una señal de alta frecuencia varía rápidamente con respecto al tiempo, mientras que una de baja frecuencia lo hace lentamente. La operación matemática llamada Transformada de Laplace convierte una ecuación en el dominio del tiempo en una ecuación en el dominio de la frecuencia y viceversa.

En el dominio de la frecuencia, la operación de un lazo cerrado de control similar al mostrado en la figura 2.6 puede ser descrita como:

$$PV(s) = E(s)G_C(s)G_P(s) \quad \dots 2.1$$

donde:

- PV = variable de proceso
- E = error
- GC = función de transferencia del controlador
- GP = función de transferencia del proceso

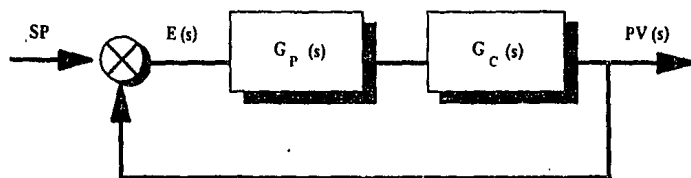


FIGURA 2.6
DIAGRAMA DE BLOQUES PARA UN SISTEMA DE CONTROL DE LAZO CERRADO

La función de transferencia es la expresión matemática que describe lo que sucede con la señal a medida que ésta se propaga a través del sistema de control. Las cuatro cantidades son funciones de la variable s de Laplace.

Por definición, el error es igual a la diferencia entre la variable de proceso y el punto de ajuste. Realizando esta sustitución y reacomodando la ecuación obtenemos la función de transferencia del sistema:

$$M(s) = \frac{PV(s)}{SP(s)} = \frac{G_C(s)G_P(s)}{1 + G_C(s)G_P(s)} \quad \dots 2.2$$

$M(s)$ es llamada la función "objeto" del controlador. Describe como cambia la PV en respuesta a alteraciones en o con respecto al punto de ajuste. La función $M(s)$ ideal es aquella en donde $M(s) = 1$, ya que esto significa que la variable de proceso es modificada ante cada perturbación de manera exacta e inmediata. Esto último es impráctico, debido a la ya mencionada existencia de inevitables retrasos en el proceso y en el controlador.

El considerar este retraso o tiempo muerto permite entonces obtener una función de transferencia mucho más objetiva. En notación de Laplace, la respuesta retrasada toma la forma de una función exponencial:

$$M(s) = EXP[-T_d s] \quad \dots 2.3$$

donde T_d es el tiempo muerto del sistema de control. A medida que T_d ó s aumentan, la función objeto tiende a cero.

La función de transferencia puede entonces rearrreglarse, resolviéndola para G_C :

$$G_C(s) = \frac{M(s)}{G_P(s)(1 - M(s))} \quad \dots 2.4$$

Reemplazando la función objeto $M(s)$ con el término exponencial que representa el tiempo muerto se obtiene:

$$G_C(s) = \frac{EXP[-T_d(s)]}{G_P(s)(EXP[-T_d(s)])} \quad \dots 2.5$$

El siguiente paso en el análisis del desempeño del sistema de control es asignar una forma determinada a la función de transferencia. Es común representar a la mayoría de los procesos industriales por medio de una función de transferencia que incluya un término de tiempo muerto (de manera similar a la función objeto) y dos términos de retraso, que representan diferencias de fase en la respuesta en frecuencia del proceso. Los términos combinados quedan como sigue:

$$G_C(s) = K_P \frac{\text{EXP}[-T_d(s)]}{[T_1s + 1][T_2s + 1]} \quad \dots 2.6$$

Los cuatro coeficientes de esta ecuación K_P , T_d , T_1 y T_2 deben calcularse empíricamente.

Una vez definida $G_P(s)$, ésta se sustituye en la expresión para $G_C(s)$, resultando:

$$G_C(s) = \frac{1}{K_P} \frac{[T_1s + 1][T_2s + 1]}{T_d(s)} \quad \dots 2.7$$

Posteriores manipulaciones algebraicas dan lugar a la ecuación universal de tres términos en donde los coeficientes K , T_r y T_i son funciones de K_P , T_d , T_1 y T_2 :

$$G_C(s) = K \left[1 + \frac{1}{T_i s} + T_r s \right] \quad \dots 2.8$$

Finalmente, la transformada inversa de Laplace regresa esta ecuación al dominio del tiempo, dando como resultado la ecuación general del PID:

$$C(t) = K \left[E(t) + \frac{1}{T_i} \int_0^t E(t) dt + T_r \frac{dE(t)}{dt} \right] \quad \dots 2.9$$

donde:

- $C(t)$ = variable de control
- $E(t)$ = error
- T_i = tiempo integral
- T_d = tiempo derivativo

Los tres términos corresponden a las acciones proporcional, integral y derivativa: el primer término es proporcional al error, del segundo término se obtiene la integral del error, y el tercer término representa la derivada del error respecto al tiempo.

2.1.5. UTILIDAD DEL PID

Considerando todos los aspectos que, como se menciona en el capítulo 1, deben de ser incluidos en el diseño de los sistemas de control, puede parecer sorprendente que un algoritmo de control tan sencillo como el PID haya tenido tanto éxito. Si el proceso es univariable, o al menos de segundo orden, y no presenta tiempos muertos, un controlador PID puede minimizar la suma de las desviaciones de la variable de proceso con respecto al punto de ajuste, resultando de ello un control que, dependiendo del punto de vista, puede resultar aceptable e incluso satisfactorio.

De cualquier manera, debido a la naturaleza del algoritmo, en la mayoría de los casos la utilización del PID ha implicado la inclusión automática de un error de modelado, por lo que aún en el mejor de los casos se presentan inestabilidad y pérdida gradual del desempeño. En algunas ocasiones los controladores se han degradado tanto que resulta mejor dejar los lazos de control abiertos.

Es importante señalar que el PID se utiliza comúnmente en sistemas multivariables de alto grado de interacción. Esto se debe a que muchas veces se prefiere la simplificación del tener muchos controladores univariables en lugar de pocos controladores multivariables, con el fin de reducir costos tanto en diseño como en operación y mantenimiento. Otro factor que favorece esta tendencia es la comparación entre la disponibilidad y costos de adquisición del equipo de control necesario para la implementación de lazos de control sencillos y el necesario para la implementación de técnicas avanzadas de control multivariable.

En realidad, este tipo de consideraciones resulta a la postre en mayores gastos que los provocados por las técnicas de control multivariable. Esto se acentúa en el caso de procesos con restricciones, ya que debido a que tanto las restricciones como las interacciones del proceso son ignoradas en la formulación inicial del algoritmo, es necesario realizar correcciones posteriores al sistema de control.

2.2. CONTROL DINÁMICO MATRICIAL DMC

El Control Dinámico Matricial DMC (*Dynamic Matrix Control*) es una de las técnicas más eficientes del Control Predictivo. El Control Predictivo, a partir del modelo del proceso, ejerce acciones preventivas de control basándose en las predicciones acerca del comportamiento de la variable por controlar. El DMC, basado en un modelo de dominio en el tiempo, fue desarrollado en la compañía petrolera Shell Oil Company (C.R. Cutler y B.L. Ramaker, "*Dynamic Matrix Control: A Computer Control Algorithm*", artículo presentado en 86ª Convención del AIChE, 1979).

La idea básica consiste en utilizar modelos de procesos con respuesta a entradas tipo escalón con dominio en el tiempo, para calcular cambios futuros en las variables manipuladas, con el fin de minimizar algún tipo de índice de desempeño.

2.2.1. MODELOS DE RESPUESTA A ENTRADAS ESCALÓN

El control DMC, como se menciona arriba, utiliza modelos de procesos con respuesta a entradas tipo escalón con dominio en el tiempo, llamados *modelos de convolución*. Como se presenta en la figura 2.7, la respuesta que un proceso presenta ante un cambio en la entrada del tipo escalón unitario ($\Delta m_j = 1$) realizado en el tiempo cero puede ser descrita por los valores de x en diferentes puntos discretos de tiempo (los b_i 's mostrados en la figura 2.7).

En el tiempo $t = nT_s$, el valor de x es $b_n T_s$. Si Δm_j no es igual a uno, el valor de x en el tiempo $t = nT_s$ es $b_n T_s \Delta m_j$. La respuesta completa puede ser descrita utilizando un número finito β de valores de los coeficientes b_i . β se escoge de manera tal que la respuesta haya alcanzado entre el 90 y 95 % de su valor final.

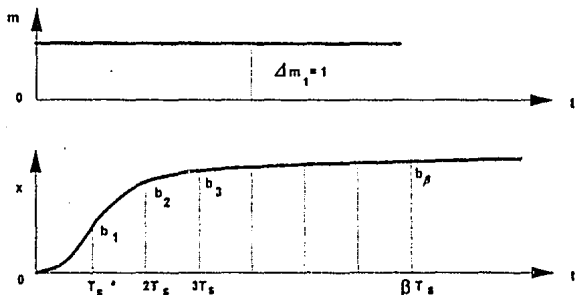


FIGURA 2.7
RESPUESTA A ENTRADAS ESCALÓN

Si ahora se presentan dos cambios escalón en la entrada, Δm_1 en $t = 0$ y Δm_2 en $t = T_s$, de acuerdo al principio de superposición (la salida total es la suma de los efectos de las entradas individuales), x puede ser calculada en cada punto discreto del intervalo de tiempo.

Para $t = 0$:	$x_0 = 0$
Para $t = T_s$:	$x_1 = b_1 \Delta m_1$
Para $t = 2T_s$:	$x_2 = b_2 \Delta m_1 + b_3 \Delta m_2$
Para $t = 3T_s$:	$x_3 = b_3 \Delta m_1 + b_4 \Delta m_2$

Si los cambios ocurren en la entrada sobre cuatro escalones en tiempo, la salida de los siguientes seis escalones será:

$$\begin{aligned}
 x_1 &= b_1 \Delta m_1 \\
 x_2 &= b_2 \Delta m_1 + b_1 \Delta m_2 \\
 x_3 &= b_3 \Delta m_1 + b_2 \Delta m_1 + b_1 \Delta m_3 \\
 x_4 &= b_4 \Delta m_1 + b_3 \Delta m_2 + b_2 \Delta m_3 + b_1 \Delta m_4 \\
 x_5 &= b_5 \Delta m_1 + b_4 \Delta m_2 + b_3 \Delta m_3 + b_2 \Delta m_4 \\
 x_6 &= b_6 \Delta m_1 + b_5 \Delta m_2 + b_4 \Delta m_3 + b_3 \Delta m_4
 \end{aligned}$$

Las ecuaciones presentadas pueden ser generalizadas para dar β valores de salida para a cambios en la entrada, utilizando notación matricial:

$$\begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ \dots \\ x_\beta \end{bmatrix} = \begin{bmatrix} b_1 & 0 & \dots & 0 \\ b_2 & b_1 & 0 & \dots \\ \dots & \dots & \dots & \dots \\ \dots & \dots & \dots & \dots \\ b_\beta & b_{\beta-1} & \dots & b_{\beta+1-\alpha} \end{bmatrix} \begin{bmatrix} \Delta m_1 \\ \Delta m_1 \\ \Delta m_1 \\ \dots \\ \Delta m_\alpha \end{bmatrix} \quad \dots 2.10$$

$$\bar{x} = \bar{B} x \Delta m \quad \dots 2.11$$

donde \bar{B} es la matriz definida en la ecuación 2.10, y que contiene β renglones y α columnas. Puede ahora ser definida una nueva matriz \bar{A} $\alpha \times \beta$, e igualarla a la matriz \bar{B} :

$$\bar{A} = \begin{bmatrix} a_{11} & a_{12} & \dots & a_{1,\alpha} \\ a_{21} & a_{22} & \dots & a_{2,\alpha} \\ \dots & \dots & \dots & \dots \\ a_{\beta,1} & a_{\beta,2} & \dots & a_{\beta,\alpha} \end{bmatrix} \quad \dots 2.12$$

Los elementos de \bar{A} (a_{ik}) se relacionan con los elementos de \bar{B} (b_{ij}) a través de la siguiente relación:

$$a_{ik} = b_{i+1-k} \quad \dots 2.13$$

Por lo tanto, el i -ésimo valor de la salida puede ser escrito como:

$$x_i = \sum_{k=1}^{\alpha} b_{i+1-k} \Delta m_k = \sum_{k=1}^{\alpha} a_{ik} \Delta m_k \quad \dots 2.14$$

La ecuación 2.14 describe como Δm_k afecta la i -ésima salida x_i utilizando el coeficiente de respuesta b_{i+1-k} . La suma de los subíndices en Δm y b siempre es $i + 1$.

Desde un punto de vista de control, es conveniente distinguir entre cambios pasados y futuros de la variable manipulada. Si han existido β cambios en la variable manipulada en los β escalones previos, y no se han presentado otros cambios, la salida cambiará en el futuro debido a los cambios en la entrada. A estos cambios en el pasado se les puede denominar $(\Delta m)^P$, y a la respuesta del sistema a estos cambios x_{LA} o respuesta de lazo abierto. Utilizando el modelo de respuesta a escalones, se denomina $\bar{x}_{LA,i}$ al valor estimado de la salida en el i -ésimo escalón en el futuro:

$$\bar{x}_{LA,i} = \sum_{k=0}^{-\beta+1} b_{i+1-k} (\Delta m_k)^P \quad \dots 2.15$$

La medición de la salida actual del proceso se denomina x_A . Si el modelo fuese perfecto, los valores predichos serían igual a los medidos. Pero en realidad siempre existe alguna diferencia. El valor predicho en el instante actual (instante 0), se calcula a partir de la ecuación anterior:

$$\bar{x}_{LA,0} = \sum_{k=0}^{-\beta+1} b_{1-k} (\Delta m_k)^P \quad \dots 2.16$$

La diferencia entre el valor de la medición actual x_A y el valor predicho actual $\bar{x}_{LA,0}$ se adiciona al valor estimado para el instante i -ésimo, a fin de conseguir una predicción más confiable de la respuesta de lazo abierto:

$$\begin{aligned} \bar{x}_{LA,i} &= \sum_{k=0}^{-\beta+1} b_{i+1-k} (\Delta m_k)^P + x_A - \bar{x}_{LA,0} \\ \bar{x}_{LA,i} &= \sum_{k=0}^{-\beta+1} b_{i+1-k} (\Delta m_k)^P + x_A - \sum_{k=0}^{-\beta+1} b_{1-k} (\Delta m_k)^P \\ \bar{x}_{LA,i} &= x_A + \sum_{k=0}^{-\beta+1} [b_{i+1-k} - b_{1-k}] (\Delta m_k)^P \end{aligned} \quad \dots 2.17$$

La respuesta real será la suma de la respuesta de lazo abierto y los efectos de los cambios futuros en las variables manipuladas. Este término se simboliza como $(\Delta m)^F$. A esta respuesta se le denomina respuesta de lazo cerrado y se simboliza como $x_{LC,i}$ para el i -ésimo escalón en el futuro. La ecuación 2.14 puede usarse para predecir la salida cuando α cambios en la variable manipulada han sido realizados en el futuro.

$$x_{LC,i} = x_{LA,0} + \sum_{k=1}^{\alpha} a_{ik} (\Delta m_k)^F \quad \dots 2.18$$

2.2.2. ALGORITMO DEL CONTROL DINÁMICO MATRICIAL

El objetivo básico del DMC es el encontrar los mejores valores de los cambios futuros de $(\Delta m)^F$'s (las variables manipuladas) que minimizen algún índice de desempeño J . Este índice de desempeño puede definirse como la suma de dos términos:

1. El cuadrado de los errores $(x^s - x_{LC,i})$ sumados en todos los intervalos β suficientes para cubrir de un 90 a un 95 % de la respuesta del sistema.
2. Los cuadrados de las variables manipuladas sumadas en los α intervalos. Generalmente, α es la mitad de β . La razón de adicionar variables manipuladas al índice de desempeño es el prevenir grandes oscilaciones que pudiesen presentarse en las entradas manipuladas. Se define también un factor ϕ para ponderar los diferentes parámetros; a mayor ϕ , menores cambios en Δm 's. Al aumentar ϕ aumenta el coeficiente de amortiguamiento del sistema.

$$J = \sum_{i=1}^{\beta} [x^s - x_{LC,i}]^2 + \phi^2 \sum_{i=1}^{\alpha} [(\Delta m_k)^F]^2 \quad \dots 2.19$$

$$J = \sum_{i=1}^{\beta} \left[x^s - x_{LA,i} - \sum_{k=1}^{\alpha} [(\Delta m_k)^F]^2 \right]^2 + \phi^2 \sum_{i=1}^{\alpha} [(\Delta m_k)^F]^2 \quad \dots 2.20$$

La matriz \bar{A} se encuentra definida en las ecuaciones 2.11 y 2.12. Por lo tanto la solución es:

$$(\bar{\Delta m})^F = \left[\bar{A}^{-T} \bar{A} + f^2 \bar{I} \right]^{-1} \bar{A}^{-T} x \quad \dots 2.21$$

donde el vector $(\Delta m)^F$ está formado por los α valores de los cambios futuros en las variables manipuladas que minimizan el índice de desempeño, siendo \bar{I} la matriz identidad.

El algoritmo del DMC es el siguiente:

PASO 1

Calcular los β valores de $x_{L,i}$ de la ecuación 2.17. Estos valores dependen únicamente de los valores pasados de las variables manipuladas y del valor actual de la variable controlada x_A .

PASO 2

Calcular los α valores de los cambios futuros en las variables manipuladas a partir de la ecuación 2.21 utilizando la matriz "dinámica" de la ecuación 2.12.

PASO 3

Implementar el primer cambio en $(\Delta m)^F$.

PASO 4

En el siguiente período de muestreo, medir la variable controlada para obtener un nuevo valor de x_A , y repetir los pasos anteriores.

El miembro derecho de la ecuación 2.21 es una matriz de elementos constantes que no cambian, por lo que únicamente deben ser calculadas una vez, y no en cada paso. El factor ϕ es un parámetro que previene cambios bruscos en las variables manipuladas. Mientras mayor sea el valor de ϕ , más robusto será el lazo de control.

El desarrollo del DMC hasta este momento presentado es aplicable para sistemas de una entrada y una salida (SISO: Single Input-Single Output). El procedimiento para extender el desarrollo a sistemas multivariables (MIMO: Multiple Inputs-Multiple Outputs) es relativamente sencillo: La matriz \bar{A} se expande para incluir todas las matrices \bar{A} que relacionen todas las entradas con todas las salidas. De la misma manera, el índice de desempeño se expande para incluir los errores en todas las variables controladas, ponderando además los movimientos de todas las variables manipuladas.

CAPITULO 3

REDES NEURALES

3. REDES NEURALES

En los últimos años, las Redes Neuronales han surgido como uno de los más poderosos paradigmas computacionales derivados de las técnicas de Inteligencia Artificial. En este capítulo se expondrán los fundamentos y principales atributos de las Redes Neuronales, así como los algoritmos más utilizados como métodos de optimización.

3.1. INTRODUCCIÓN

3.1.1. ANALOGÍA CON EL CEREBRO

La neurona es la unidad celular fundamental del sistema nervioso y, particularmente, del cerebro. Cada neurona es simplemente una unidad microprocesadora que recibe y combina las señales de otras muchas neuronas a través de estructuras de entrada llamadas dendritas. Las señales provenientes de las dendritas son comunicadas a través del cuerpo de la neurona por medio de la sinapsis. Si la señal combinada es lo suficientemente fuerte, activa el mecanismo de disparo que produce la señal de salida de la neurona; esta señal se transfiere a lo largo de otro componente neuronal llamado axón. Esta simple transferencia de información es de naturaleza química, pero produce efectos eléctricos cuantificables.

El cerebro humano está formado por millones de neuronas densamente interconectadas entre sí. El axón (salida) de cada neurona se ramifica y se conecta con las dendritas (entradas) de otras neuronas a través de uniones llamadas sinapsis. La transmisión a través de estas uniones es de naturaleza química, y la cantidad de señales transmitidas depende del número de sustancias químicas (neurotransmisores) liberadas por el axón y recibidas por las dendritas. La eficiencia sináptica es el parámetro que se modifica cuando el cerebro aprende. La sinapsis, combinada con el procesamiento de información en la neurona, conforma el mecanismo básico de memoria del cerebro.

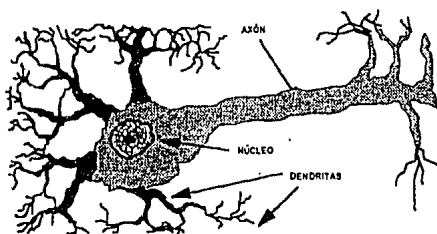
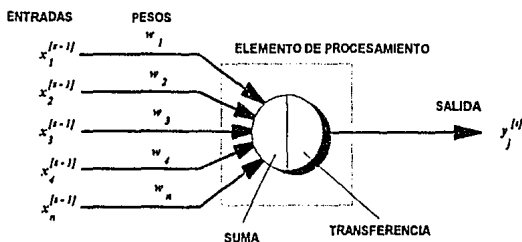


FIGURA 3.1
NEURONA BIOLÓGICA

3.1.2. REDES NEURALES

En una Red Neural artificial (también llamada red neuronal), la unidad análoga a la neurona biológica es denominada "Elemento de Procesamiento". Un Elemento de Procesamiento EP tiene muchas entradas, cuyos valores son combinados, normalmente por simple suma. El resultado es un determinado nivel de actividad interna en el EP. El resultado de esta combinación es modificado después por una función de transferencia. Esta función de transferencia puede ser una función umbral, que únicamente transmite información cuando el nivel de actividad alcanza un valor determinado, o bien una función continua del valor combinado de las entradas. El valor de la salida de la función de transferencia generalmente se transmite directamente a la salida del EP.

Esta salida del EP es conectada a las entradas de otros Elementos de Procesamiento a través de conexiones con pesos asociados (conexiones ponderadas) que corresponden a la fuerza sináptica de las conexiones neurales. Debido a que cada conexión tiene su propio peso asociado, las entradas a cada EP son modificadas por estos pesos antes de la suma. Debido a ello, la función de suma es una suma ponderada.



$$I_j^s = \sum_{i=1}^n w_{ji}^s x_i^{s-1} \quad y_j^s = f(I_j^s)$$

FIGURA 3.2
NEURONA ARTIFICIAL

Una Red Neural consiste en muchos Elementos de Procesamiento conectados entre sí, como se describe en el párrafo anterior. Normalmente, estos elementos se encuentran organizados en grupos llamados estratos o niveles. Una red típica consiste en una secuencia de estratos o niveles con conexiones totales o aleatorias entre estratos sucesivos. Para el mundo exterior, existen dos niveles típicos: el estrato de entrada, en donde el conjunto de datos se presenta a la Red, y el estrato de salida, el cual presenta la respuesta de la Red ante un determinado patrón de entrada. Los demás estratos se denominan estratos internos u ocultos.

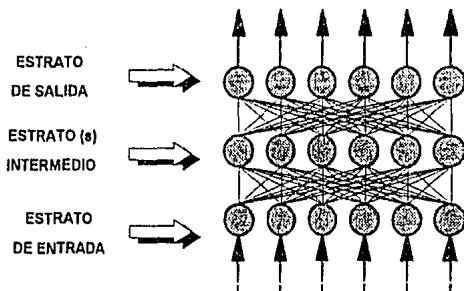


FIGURA 3.3
ARQUITECTURA DE UNA RED NEURAL

3.2. OPERACIÓN DE REDES NEURALES

Existen dos fases principales en la operación de una Red Neuronal: Aprendizaje y Recapitulación.

3.2.1. APRENDIZAJE

El aprendizaje es el proceso de adaptar o modificar los pesos asociados a las conexiones en respuesta a los datos presentados en las capas de entrada y salida de la Red. Un patrón de datos presentado en el estrato de salida corresponde a la respuesta esperada de un determinado conjunto de datos de entrada. Este caso es conocido como "Aprendizaje Supervisado".

Si la salida deseada es diferente de la entrada, la Red Neural se denomina red hetero-asociativa. En el caso en el que para todos los ejemplos el vector de entrada sea igual al de salida, la Red se conoce como red auto-asociativa. Si no existe una salida deseada, el aprendizaje se denomina "Aprendizaje No Supervisado"

Independientemente del tipo de aprendizaje utilizado, una característica esencial de cualquier Red es su regla de aprendizaje, la cual especifica la forma en la que los pesos se adaptan en respuesta a un ejemplo dado. El aprendizaje puede requerir el presentar a la Red Neural muchos ejemplos un gran número de veces. Los parámetros que gobiernan la regla de aprendizaje pueden cambiar a medida que la Red avanza en el período de aprendizaje. El control a largo plazo de estos parámetros se conoce como programa de aprendizaje.

3.2.2. RECAPITULACIÓN

El término de recapitulación se refiere al proceso de respuesta de la Red Neuronal ante diferentes estímulos presentados en el estrato de entrada y la generación de la respuesta en la capa de salida. Frecuentemente, la recapitulación es una parte integral del proceso de aprendizaje, en el sentido de que la respuesta deseada se compara con la salida real de la Red, con el fin de crear una señal de error.

La forma más simple de una Red es aquella en la que no existen conexiones de retroalimentación de una a otra. En este caso, la información es transmitida desde el estrato de entrada hacia los estratos intermedios, utilizando las funciones de transferencia y de suma particulares del tipo de Red.

Muchos tipos de Redes Neuronales tienen una función de energía asociada. Cada estado de la Red, definido por un conjunto particular de salidas de los EP's, tiene un valor de energía. El proceso de recapitulación modifica, de manera iterativa, los estados de la Red de forma tal que la energía disminuya, dando lugar a un mínimo en la superficie energética. Para Redes Neuronales con retroalimentación, puede utilizarse un programa de recapitulación para controlar los parámetros a medida que el proceso de aprendizaje progresa.

Sincronía y Asincronía

Una Red puede operar en modo sincronizado o desincronizado. El comportamiento sincronizado significa que todos los Elementos de Procesamiento disparan simultáneamente en cada paso. Por otra parte, en una Red de comportamiento asincrónico todos los EP's emiten su salida de manera aleatoria.

3.2.3. OPERACIÓN EN ESTRATOS

Las dos operaciones llevadas a cabo en una Red Neural que afectan de manera total a los estratos son Normalización y Competencia.

Normalización

La función de normalización actúa sobre el vector de valores correspondiente a la salida de un estrato, escalándolo de manera tal que la salida total constituye un valor fijo. En los sistemas biológicos, esto se realiza conectando cada uno de los EP's a todos los EP's del estrato siguiente. Estas conexiones permiten que los Elementos de Procesamiento, de manera individual, perciban la salida del estrato inferior y modifiquen en consecuencia sus propios valores. El resultado de la normalización es que la actividad total en cada estrato permanece aproximadamente constante.

Competencia

Se refiere a la interacción que un Elemento de Procesamiento tiene con los demás EP's en el mismo estrato. De manera diferente a la normalización, en donde todos los EP's ajustan su salida para crear un nivel de actividad fijo, en la operación de competencia solo algunos Elementos disparan. Una forma común de competencia es la competencia por niveles de actividad: son los EP's con los niveles más altos aquellos que emiten su salida.

3.3. CARACTERÍSTICAS DE LA NEUROCOMPUTACIÓN

A continuación se presentan algunas de las características que diferencian a las Redes Neuronales de la computación tradicional.

3.3.1. APRENDIZAJE POR EJEMPLOS

A diferencia de los sistemas expertos tradicionales, en donde el conocimiento se representa explícitamente, en forma de reglas, las Redes Neuronales generan sus propias reglas aprendiendo de los ejemplos que se les presentan. El aprendizaje se lleva a cabo a través de una regla que modifica o adapta los pesos de las conexiones de la Red, en respuesta a la comparación entre las salidas producidas con base en los ejemplos de entrada y las salidas deseadas.

En el aprendizaje no supervisado, únicamente se le presentan a la Red conjuntos de entrada, y la red se organiza internamente, de manera tal que cada uno de los EP's responde de manera individual a los diferentes estímulos. Los parámetros generados en los EP's constituyen entonces valores que representan distintos conceptos del mundo real.

En el aprendizaje supervisado, por cada conjunto de valores presentado como entradas, se presenta también una salida esperada, producida por dicha entrada, y la Red Neural gradualmente se modifica a fin de imitar el patrón de entradas/salidas presentado. En general, este aprendizaje se lleva a cabo con alguna variación de cualquiera de las siguientes técnicas:

Aprendizaje Hebbian

En esta técnica, el peso de una conexión hacia la entrada de un EP aumenta tanto si el valor de la entrada es alto como si el valor de la salida es alto. En términos biológicos, esto significa que una conexión entre dos EP's es reforzada cada vez que se lleva a cabo la sinapsis entre ellas.

Aprendizaje por Regla Delta

Las variaciones de esta técnica se basan en la reducción del error entre la salida deseada y la salida real de la Red. Esto se realiza modificando los pesos de las conexiones.

Aprendizaje Competitivo

Los Elementos de Procesamiento compiten entre sí, y aquél que proporcione la respuesta más fuerte a una determinada entrada, se modifica a sí mismo para igualar lo más posible al valor de la entrada.

3.3.2. MEMORIA DISTRIBUIDA ASOCIATIVA

Una característica muy importante de las Redes Neuronales es que almacenan información. La memoria en neurocomputación es distribuida. Los pesos de las conexiones constituyen las unidades de memoria de las Redes Neuronales. Los valores de los pesos representan el estado actual del conocimiento de la Red. Un elemento de conocimiento, representado por ejemplo por un par de entrada/salida-deseada, se distribuye a través de muchas de las unidades de memoria en la red y comparte estas unidades con otros elementos de conocimiento almacenados en la Red.

La memoria neurócomputacional es asociativa en el sentido de que si a la Red entrenada se le presenta un conjunto parcial a la entrada, la Red elegirá los parámetros más acordes con dicha entrada, generando una salida que corresponde a una entrada completa. Si la red es auto-asociativa (es decir, que la entrada sea igual a la salida para todos los ejemplos usados en el entrenamiento), la presentación de vectores de entrada parciales resultará en su complementación.

La naturaleza de la memoria en neurocomputación permite respuestas razonables ante entradas incompletas, con ruido o completamente diferentes a las presentadas durante el entrenamiento. Esta propiedad se conoce como generalización. La calidad y el significado de una generalización depende de cada aplicación en particular y del tipo y complejidad de la Red. Redes estratificadas no lineales aprenden acerca de los conceptos en los estratos intermedios, y los combinan para generar su salida.

3.3.3. TOLERANCIA A FALLAS

Mientras que los sistemas de cómputo tradicionales son muy susceptibles incluso a daños menores en la memoria, los sistemas neurocomputarizados son altamente tolerantes a fallas. Este término se refiere al hecho de que, en la mayoría de las Redes Neuronales, si alguno de los EP's falla, es destruido o sus conexiones son dañadas, el comportamiento global de la Red se degrada tan solo ligeramente.

A medida que los EP's fallan, el comportamiento de la Red va degradándose un poco más. El desempeño de la Red se altera, pero el sistema no se detiene de manera total. Estos sistemas son tolerantes a fallas debido a que la información se encuentra distribuida a través del sistema. Esta característica de las Redes Neuronales las hacen extremadamente adecuadas para aplicaciones en las que fallas en el equipo de control pueden significar un desastre, tales como: operaciones de plantas nucleares, guía de misiles, etc.

3.3.4. RECONOCIMIENTO DE PATRONES

Los sistemas neurocomputacionales realizan las actividades relacionadas con la identificación de parámetros de una manera más eficiente que los sistemas estadísticos tradicionales o los sistemas expertos. El reconocimiento de patrones requiere la caracterización de grandes cantidades de información de manera simultánea, generando después una salida generalizada o categórica. También requiere de una respuesta razonable ante entradas con ruido aleatorio o incompletas.

Los sistemas basados en Redes Neuronales poseen todas estas características, así como también la capacidad de aprender y construir estructuras específicas para un problema particular.

3.3.5. SÍNTESIS

Algunos tipos de Redes son capaces de aprender mapeos continuos y complejos de una o más salidas. Esta capacidad para sintetizar funciones complejas continuas es análoga a la capacidad de los sistemas biológicos para aprender movimientos coordinados.

Estas características especiales de las Redes Neuronales las hacen aplicables a un gran número de problemas del mundo real, como son:

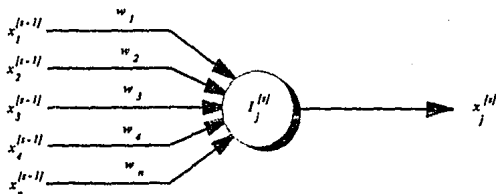
- Procesamiento de lenguaje.
- Comprensión de imágenes.
- Reconocimiento de caracteres.
- Problemas combinatorios.
- Reconocimiento de patrones en imágenes.
- Procesamiento de señales: simulación y modelado.
- Modelados financieros y económicos.
- Servo-control.
- Síntesis funcional.

3.4. LA TÉCNICA DE BACK-PROPAGATION O "PROPAGACIÓN HACIA ATRÁS"

Una de las técnicas más utilizadas dentro de los tipos de Reglas de Delta (de hecho la más popular) es la llamada "Propagación hacia Atrás" o "Back-Propagation". En lo sucesivo, se utilizará el término de Back-Propagation o BP.

3.4.1 REDES CON BACK-PROPAGATION

Una red típica de Back-Propagation siempre presenta un estrato de entrada, uno de salida, y al menos un estrato intermedio. No existe un límite teórico en cuanto al número de estratos internos, aunque normalmente son uno o dos. Algunos trabajos³ indican que para resolver problemas complejos de clasificación se requieren redes con un máximo de tres estratos intermedios. Cada uno de los estratos se encuentra totalmente conectado a la capa siguiente. Durante el aprendizaje la información se propaga hacia atrás en la Red, y se utiliza para actualizar los pesos de las conexiones. La Red puede ser autoasociativa o heteroasociativa. En la figura 3.4 se presenta un elemento típico de una red con BP:



$$x_j^{[s]} = f\left(\sum_{i=1}^n w_{ji}^{[s]} x_i^{[s-1]}\right) \quad x_j^{[s]} = f(I_j^{[s]})$$

FIGURA 3.4
ELEMENTO DE PROCESAMIENTO DE RED NEURAL CON BACK-PROPAGATION.

(3) Jones, W.P.; Hoskins, J. "BACK-PROPAGATION: A GENERALIZED DELTA LEARNING RULE"
BYTE Magazine, E.U.A., octubre 1987

donde:

$x_j^{[s]}$: Estado actual de la salida de la neurona j -ésima en el estrato s .

$w_{ji}^{[s]}$: Peso de la conexión entre la neurona i -ésima en el estrato $s-1$ y la neurona j -ésima en el estrato s .

$I_j^{[s]}$: Suma ponderada de las entradas de la neurona j -ésima en el estrato s .

Un elemento de Back-Propagation transfiere sus entradas como sigue:

$$x_j^{[s]} = f\left(\sum_{i=1}^n w_{ji}^{[s]} x_i^{[s-1]}\right) = f(I_j^{[s]}) \quad \dots 3.1$$

donde f es tradicionalmente una función sigmoide pero que en realidad puede ser cualquier función diferenciable. La función sigmoide se define como:

$$f(z) = [1.0 + \text{EXP}(-z)]^{-1} \quad \dots 3.2$$

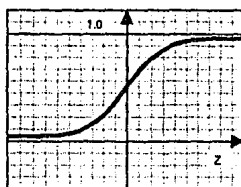
Sin embargo, otras funciones de transferencia pueden utilizarse, tales como la tangente hiperbólica, definida por:

$$f(z) = \frac{\text{EXP}(z) - \text{EXP}(-z)}{\text{EXP}(z) + \text{EXP}(-z)} \quad \dots 3.3$$

Al igual que la derivada de la función sigmoide, la derivada de la tangente hiperbólica puede ser expresada en términos de sí misma:

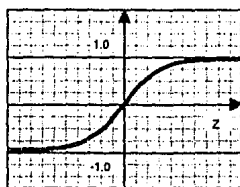
$$f'(z) = [1.0 + f(z)][1.0 - f(z)] \quad \dots 3.4$$

En la siguiente figura se presentan de manera gráfica las funciones de transferencia típicas de una Red con Back-Propagation.



SIGMOIDE

$$f(z) = [1.0 + \text{EXP}(-z)]^{-1}$$



TANGENTE
HIPERBÓLICA

$$f(z) = \frac{\text{EXP}(z) - \text{EXP}(-z)}{\text{EXP}(z) + \text{EXP}(-z)}$$

FIGURA 3.5
FUNCIONES DE TRANSFERENCIA PARA BACK-PROPAGATION

En el desarrollo siguiente se utiliza la sigmoide como función de transferencia.

3.4.2. PROPAGACIÓN HACIA ATRÁS DEL ERROR LOCAL

Continuando con la descripción de la técnica de BP, es posible asumir la existencia de una función de error global E asociada con la Red, siendo E una función diferenciable de todos los pesos de las conexiones de la Red. El parámetro crítico que es transferido hacia atrás a través de los estratos se define como:

$$e_j^{[s]} = - \frac{\partial E}{\partial I_j^{[s]}}$$

...3.5

Aplicando la regla de la cadena dos veces sucesivamente se obtiene una relación entre el error local en un EP dado en el nivel s y todos los errores locales en el nivel $s+1$:

$$e_j^{[s]} = f' \left(I_j^{[s]} \right) \sum_k \left[e_k^{[s+1]} w_{kj}^{[s+1]} \right]$$

...3.6

Nótese que la ecuación anterior considera la existencia de un estrato superior; por ello, la ecuación 3.6 no puede utilizarse para estratos de salida.

Si f es la función sigmoide definida en la ecuación 3.2, entonces su derivada puede ser expresada como:

$$f'(z) = f(z)[1.0 - f(z)] \quad \dots 3.7$$

Por lo tanto, de (3.1) la ecuación 3.6 puede ser reescrita como:

$$e_j^{[s]} = x_j^{[s]} \left(1.0 - x_j^{[s]} \right) \sum_k \left[e_k^{[s+1]} w_{kj}^{[s+1]} \right] \quad \dots 3.8$$

El término de suma de la ecuación 3.8, utilizado para propagar hacia atrás el error, es análogo al término de suma en (3.1), el cual se utiliza para propagar hacia adelante las señales de entrada a través de la Red. Así, el mecanismo principal en una Red de BP es propagar hacia adelante las señales de entrada a través de los estratos de la red hacia el estrato de salida, calcular el error en la salida y después transferir hacia atrás el error utilizando (3.8) o de manera más general (3.6). La multiplicación del error por la derivada de la función de transferencia escala el error.

3.4.3. MINIMIZANDO EL ERROR GLOBAL

El objetivo principal del proceso de aprendizaje es el minimizar la función de error global E del sistema, por medio de la modificación de los pesos de las conexiones. A continuación se presenta como lograr lo anterior basándose en los errores locales de cada EP.

Dado un conjunto de pesos $w_{ij}^{[s]}$, se debe determinar la forma de incrementarlos o disminuirlos, a fin de disminuir el valor de la función de error global. Esto se realiza utilizando la siguiente regla de gradiente descendente (gradiente negativo):

$$\Delta w_{ji}^{[s]} = -\xi \frac{\partial E}{\partial w_{ji}^{[s]}}$$

...3.9

donde ξ se conoce como coeficiente de aprendizaje. Esta función modifica cada uno de los pesos tomando en cuenta tanto el tamaño como el gradiente descendente en la superficie del error.

Las derivadas parciales de (3.9) pueden ser calculadas directamente de los valores del error local. Aplicando la regla de la cadena y la ecuación 3.1 se obtiene:

$$\frac{\partial E}{\partial w_{ji}^{[s]}} = \left[\frac{\partial E}{\partial I_j^{[s]}} \right] \left[\frac{\partial I_j^{[s]}}{\partial w_{ji}^{[s]}} \right]$$

$$\frac{\partial E}{\partial w_{ji}^{[s]}} = e_j^{[s]} x_i^{[s-1]}$$

...3.10

Combinando (3.10) con (3.9) da como resultado:

$$\Delta w_{ji}^{[s]} = \xi e_j^{[s]} x_i^{[s-1]}$$

...3.11

3.4.4. LA FUNCIÓN DE ERROR GLOBAL

En el desarrollo anterior se asumió la existencia de una función global del error. En esta sección se describirá dicha función, necesaria para definir los errores locales en el estrato de salida, para que puedan ser propagados hacia atrás a través de la Red. Para ello, es conveniente definir tres vectores:

- ρ : vector de entrada en el estrato inicial.
- σ : vector de salida real del estrato final.
- τ : vector de salida esperado.

Así, puede definirse una función de error global como sigue:

$$E = 0.5 \sum_k [\tau_k - \sigma_k]^2 \quad \dots 3.12$$

donde el subíndice k indica los componentes de τ y σ . Aquí el error local de cada estrato se define como $\tau - \sigma$. De la ecuación 3.5, el error local escalado en cada EP del estrato de salida es:

$$\begin{aligned} e_k^\sigma &= -\frac{\partial E}{\partial I_k} \\ e_k^\sigma &= -\left[\frac{\partial E}{\partial I_k} \right] \left[\frac{\partial \sigma_k}{\partial I_k} \right] \\ e_k^\sigma &= -[\tau_k - \sigma_k] f'(I_k) \end{aligned} \quad \dots 3.13$$

La ecuación 3.12 define el error global de la Red para un par de p y τ . Tal y como se encuentra especificada, resulta proporcional al cuadrado de la distancia euclidiana entre la salida esperada y la salida real para un caso particular de entradas. En realidad, cualquier función de error cuyas derivadas puedan calcularse en el estrato de salida pueden sustituir a la función presentada en la ecuación 3.12.

3.4.5. ALGORITMO DE BACK-PROPAGATION

Dados los vectores p y t :

1. Presentar p en el estrato inicial y propagarlo a través de la red hasta el estrato final, obteniendo σ . A medida que la información se transmite a través de la Red Neural, se determinan los valores de las sumas de entradas I_j y los estados de salida x_j para cada EP de la Red.
2. Para cada uno de los EP's, calcular el error local escalado tal y como se especifica en (3.13), calculando también el gradiente de los pesos como se muestra en (3.11).
3. Para cada estrato intermedio, comenzando con el del penúltimo nivel y finalizando con el segundo, calcular tanto el error escalado local (3.8) como el gradiente para modificar los pesos (3.11).
4. Modificar todos los pesos de la Red mediante la suma de cada gradiente a los pesos correspondientes.

CAPITULO 4
REDES NEURALES COMO ALTERNATIVA
DE CONTROL DE PROCESOS

4. REDES NEURALES COMO ALTERNATIVA DE CONTROL DE PROCESOS

4.1 INTRODUCCIÓN

La teoría de sistemas matemáticos ha tenido importantes avances en las cuatro últimas décadas, transformándose en toda una disciplina científica que se extiende a muy diferentes áreas del conocimiento, desde el diseño y desarrollo por un lado hasta las matemáticas por el otro. El área en la cuál esta teoría ha tenido el mayor desarrollo concierne a los sistemas lineales, introduciendo importantes conceptos y resultados teóricos en campos tales como estabilidad, optimización, teoría multivariable y control adaptivo.

Las poderosas técnicas desarrolladas en el área del control adaptivo complementan las tecnologías de cómputo, presentando un enorme potencial en el universo de las aplicaciones, donde los sistemas deben de ser controlados en presencia de incertidumbre. A pesar de que la inmensa mayoría de sistemas son no lineales, casi todos las teorías de dichos sistemas están fuertemente cimentadas en la teoría de los sistemas lineales.

En años recientes, las Redes Neuronales se han perfilado como importantes herramientas en los problemas de optimización y de reconocimiento de patrones. Desde un punto de vista de teoría de sistemas, estas Redes pueden ser consideradas como componentes eficientes de sistemas complejos y no lineales.

4.2. MODELOS PARA LA IDENTIFICACIÓN DE SISTEMAS DINÁMICOS NO LINEALES

En esta sección se presentarán cuatro modelos desarrollados por Narendra y Parthasarathy⁴ en 1989, para la representación de modelos SISO no lineales (SISO: Single Input Single Output). De acuerdo a los autores, estos modelos fueron escogidos tanto por su generalidad como por su seguimiento analítico.

Los cuatro modelos surgen de los modelos que han sido utilizados tradicionalmente en la literatura de sistemas adaptivos para la identificación de sistemas lineales, y pueden ser considerados como generalizaciones para sistemas no lineales.

CARACTERIZACIÓN DE PROCESOS

En las siguientes ecuaciones, $[u(k), x(k)]$ representan el par E/S del proceso en el tiempo k . Las funciones f y g son funciones diferenciables de sus argumentos.

4.2.1. MODELO I

La salida del proceso desconocido y no lineal se considera linealmente dependiente de sus valores en el pasado, y no linealmente dependiente de los valores pasados de la entrada. La estructura del sistema se presenta en la figura 4.1.

$$x(k+1) = \sum_{i=0}^{n-1} \alpha_i u(k-i) + f[u(k), u(k-1), \dots, u(k-n+1)] \quad \dots 4.1$$

(4) Narendra, K.S.; Parthasarathy, K. "BACK-PROPAGATION IN DYNAMICAL SYSTEMS CONTAINING NEURAL NETWORKS"
 Technical report 880, Center for systems science, Department of Electric Engineering, Yale University, New Haven, CT, E.U.A., 1987

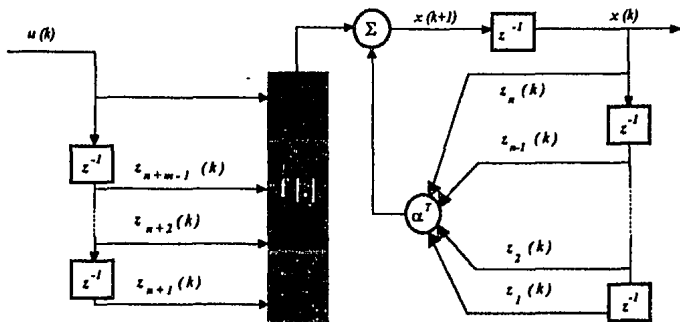


FIGURA 4.1
MODELO I: ESTRUCTURA DEL SISTEMA

4.2.2. MODELO II

En este caso, la salida depende linealmente de la entrada $u(k)$ y de sus valores pasados, pero de manera no lineal de sus propios valores del pasado. La estructura del sistema se presenta en la figura 4.2.

$$x(k+1) = f[x(k), x(k-1), \dots, x(k-n+1)] + \sum_{i=0}^{m-1} \beta_i x(k-i) \quad \dots 4.2$$

4.2.3. MODELO III

La ecuación de este modelo es

$$x(k+1) = f[x(k), x(k-1), \dots, x(k-n+1)] + g[u(k), u(k-1), \dots, u(k-m+1)] \quad \dots 4.3$$

por lo que depende de manera no lineal tanto de valores pasados en la entrada como en la salida. Estos efectos de las entradas y salidas pasadas son aditivos. La estructura del sistema se presenta en la figura 4.3.

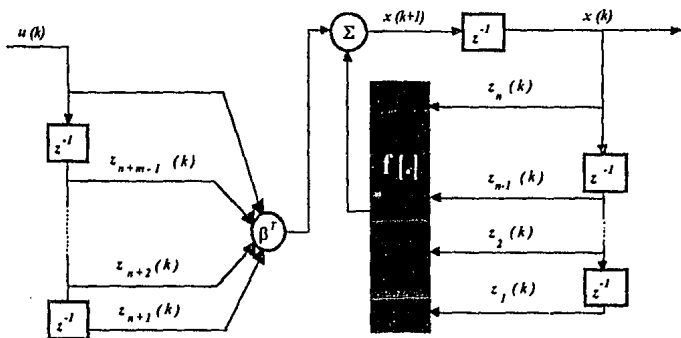


FIGURA 4.2
MODELO II: ESTRUCTURA DEL SISTEMA

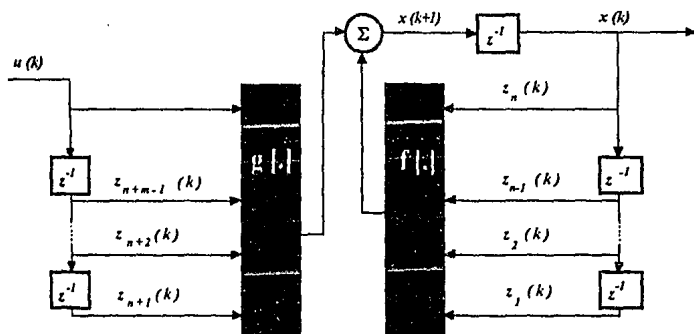


FIGURA 4.3
MODELO III: ESTRUCTURA DEL SISTEMA

4.2.4. MODELO IV

Este es el más general de los modelos aquí presentados. Su salida en cada instante es una función no lineal de los valores pasados de entradas y salidas. La estructura del sistema se presenta en la figura 4.4.

$$x(k+1) = f[x(k), x(k-1), \dots, x(k-n+1); u(k), u(k-1), \dots, x(k-m+1)] \quad \dots 4.4$$

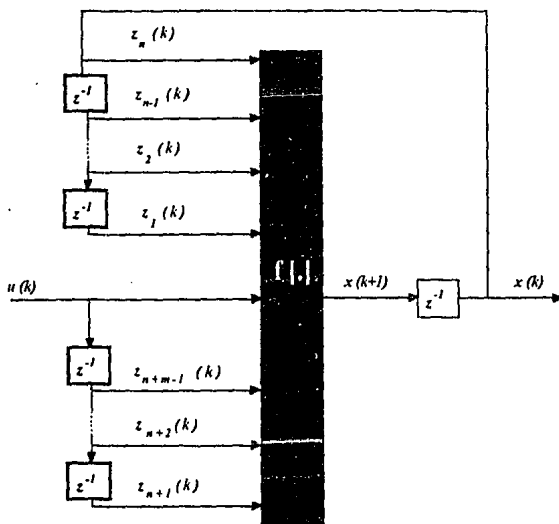


FIGURA 4.4
MODELO IV: ESTRUCTURA DEL SISTEMA

4.3. PROCEDIMIENTO DE IDENTIFICACIÓN

Con base en lo anterior, Narendra desarrolló un modelo de identificación de parámetros constituido por Redes Neuronales. En cada caso, se considera que la Red contiene el suficiente número de estratos y nodos en cada estrato, a fin de ajustar con el conjunto de datos del proceso. Desde el punto de vista de análisis matemático, esto implica que las funciones no lineales de las ecuaciones diferenciales que describen el proceso pueden ser reemplazadas por Redes Neuronales con matrices de pesos desconocidos W_i^* .

Para identificar el proceso, se escoge un modelo de identificación basándose en la información existente. Por ejemplo, si el proceso posee una estructura similar a la del Modelo III, el modelo tendrá la arquitectura presentada en la figura 4.5:

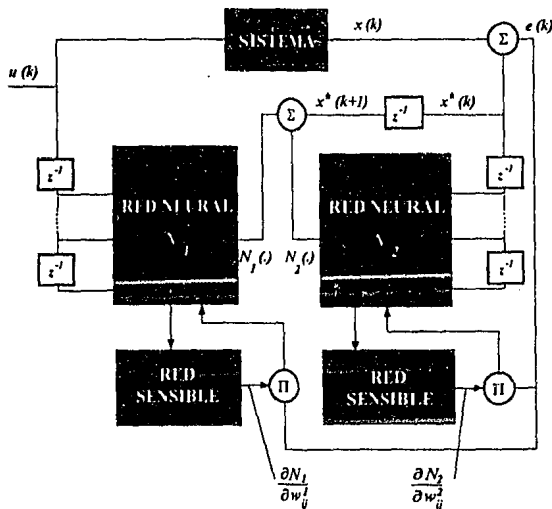


FIGURA 4.5
MODELO III: ESTRUCTURA DEL MODELO DE IDENTIFICACIÓN

El problema consiste entonces en determinar los pesos de las dos Redes Neurales N_1 y N_2 , a fin de que el mapeo de N_1 sea igual a $g[f, j]$, y que el mapeo de N_2 sea igual al de $f[j]$. Si $x(k+1)$ y $x^*(k+1)$ son respectivamente las salidas en la etapa $k+1$ del proceso y del modelo de identificación, el error $e = x^*(k+1) - x(k+1)$ se utiliza para ajustar los pesos de N_1 y N_2 .

4.4. EXCITACIÓN PERSISTENTE

El tipo de entradas utilizadas para entrenar sistemas adaptivos es de gran importancia. El conjunto de entradas debe ser representativo de la clase completa de entradas a las que el sistema estará sujeto. Ésto asegurará que el sistema responda del modo deseado cuando se presente una entrada no incluida de manera específica en el conjunto de entrenamiento. A ésto se le conoce como *excitación persistente*.

Sea θ un vector de parámetros ajustables en un sistema y θ^* un vector constante, de tal manera que cuando $\theta = \theta^*$, el sistema tenga la misma función de transferencia que el modelo. El vector θ puede ser único o bien pertenecer a un conjunto S . En este último caso, el sistema tiene el mismo comportamiento E/S que el modelo, para todos los valores constantes de $\theta \in S$. Si entradas específicas son aplicadas tanto al sistema como al modelo, el error de la salida, con el tiempo, tenderá asintóticamente a cero, incluso cuando $\theta \notin S$. No obstante, si la entrada es persistentemente excitada, $\theta(k) \rightarrow \theta^* \in S$. Si θ^* es único, ello implica que los parámetros del sistema pueden ser determinados de manera exacta.

Este concepto de excitación persistente es muy importante al trabajar en la identificación y el control de sistemas dinámicos no lineales utilizando Redes Neurales. Sea U el conjunto de entradas al cual pertenece la entrada u . Si se escoge una entrada específica $u \in U$, el error de la salida se aproxima rápidamente a cero cuando se utiliza la técnica de Back-Propagation para ajustar los pesos de la Red. Sin embargo, si Ψ y Ψ^* representan respectivamente a los operadores del sistema y del modelo de identificación, Ψ^* no se "acercará" a Ψ en ninguna forma: examinando, por ejemplo, la norma $\|\Psi u - \Psi^* u\|$, se puede observar que no es pequeña para toda $u \in U$. Esto resulta evidente al considerar una entrada u no perteneciente al conjunto de entrenamiento.

Si la entrada al sistema (y al modelo de identificación) es lo suficientemente general y los pesos de la Red son ajustados durante un tiempo lo suficientemente grande, el error de la salida puede ser reducido para cualquier entrada en U . La entrada mencionada debe entonces ser considerada persistentemente excitada.

4.5 CONTROL ADAPTIVO

El control adaptivo se refiere al problema de controlar la salida de un sistema de estructura conocida pero de parámetros no conocidos. Para resolver este problema analíticamente, en la teoría de sistemas adaptivos el proceso a controlar se considera lineal e invariable en el tiempo, con parámetros desconocidos. Estos parámetros pueden ser considerados como elementos de un vector p . Si p es conocido, el vector de parametrización θ de un controlador puede ser escogido como θ^* , de manera tal que tanto el proceso como el controlador se comporten como un modelo de referencia descrito por ecuaciones diferenciales lineales. Si no se conoce p , el vector $\theta(t)$ debe ser ajustado en línea, utilizando toda la información disponible acerca del sistema.

Existen dos formas de aplicación del control adaptivo en un proceso no conocido: Control Directo y Control Indirecto.

4.5.1. CONTROL DIRECTO

En el control directo, los parámetros del controlador son ajustados directamente con el fin de reducir alguna función del error de la salida.

Por el momento, no se encuentran disponibles métodos para ajustar de forma directa los parámetros del controlador (la Red N_C del figura 4.6). Esto se debe a que, tanto el proceso como el controlador son de naturaleza no lineal. Inclusive, no puede utilizarse el método de Back-Propagation, ya que al ser el proceso desconocido, no puede utilizarse para generar las derivadas parciales necesarias. Debido a ello, y hasta que se desarrollen métodos efectivos de control directo, deben de utilizarse métodos indirectos de control.

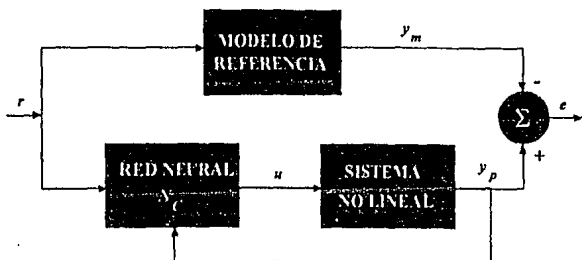


FIGURA 4.6
CONTROL ADAPTIVO DIRECTO CON REDES NEURALES

4.5.2. CONTROL INDIRECTO

En el control indirecto, los parámetros del proceso son estimados como $p^*(t)$ en cualquier instante, escogiéndose el vector de parametrización $\theta(t)$, asumiendo que $p^*(t)$ representa el valor real del vector de parametrización.

De esta manera, se encuentran los parámetros del proceso utilizando el procedimiento descrito en la sección 4.3., y los parámetros del modelo se actualizan utilizando el error de identificación. Los parámetros del controlador son ajustados por medio de Back-Propagation. Un diagrama de bloques de este tipo de control se presenta en la figura 4.7.

Tanto la identificación como el control pueden llevarse a cabo en cada instante o después de procesar la información a través de intervalos finitos de tiempo. Si no se presentan perturbaciones externas y/o ruido, es razonable ejecutar ambas acciones simultáneamente.

Por otro lado, si se presentan dichas perturbaciones, lo conveniente es realizar la identificación de manera continua, actualizando los parámetros de control siguiendo una escala de tiempo más lenta, a fin de asegurar robustez.

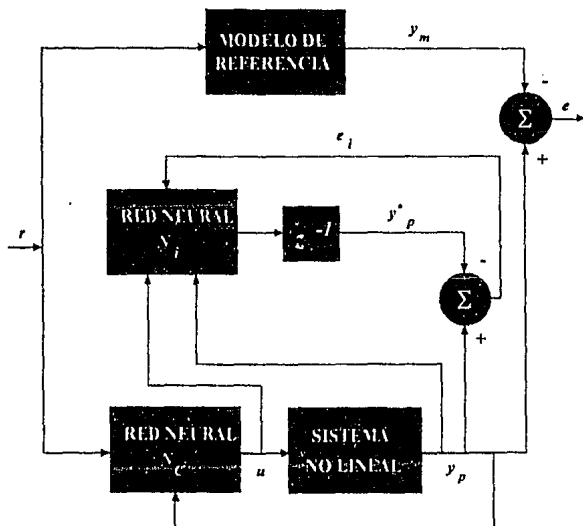


FIGURA 4.7
CONTROL ADAPTIVO INDIRECTO UTILIZANDO REDES NEURALES

4.5.3. CONTROL ADAPTIVO EN SISTEMAS DINÁMICOS NO LINEALES

A continuación se presentará un ejemplo de control adaptativo en un sistema similar al presentado en la figura 4.7, y que resulta independiente del modelo utilizado para la identificación. Este último depende de la información disponible referente al proceso. Los valores retrasados de la entrada y salida del proceso constituyen la entrada de la Red Neuronal N_c , la cual genera la señal de control que retroalimenta al proceso. Los parámetros de la Red Neuronal N_i son ajustados mediante la propagación hacia atrás del error de identificación e_i , mientras que los de la Red N_c son ajustados mediante la propagación hacia atrás del error de control (entre la salida del modelo de referencia y el modelo de identificación) a través del modelo de identificación.

Se debe elegir entonces un modelo de identificación que permita determinar una señal de control que genere un error de salida que tienda asintóticamente a cero. Si se consideran los modelos presentados en la sección 4.2, este problema puede ser muy grande, debido a que todavía no existen métodos generales para controlar procesos no lineales, aún cuando sus características de E/S sean completamente conocidas. De los cuatro modelos presentados, únicamente el II se presta por sí solo a propósitos de control.

Supóngase que el sistema pueda ser descrito por la ecuación diferencial

$$x(k+1) = f[x(k), x(k-1), \dots, x(k-n+1)] + u(k) \quad \dots 4.5$$

y el modelo de referencia descrito por:

$$x_m(k+1) = \sum_{i=0}^{n-1} \alpha_{mi} x_m(k-i) + \sum_{j=0}^{m-1} \beta_{mj} r(k-j) \quad \dots 4.6$$

donde $\alpha_{mi}, \beta_{mj} \in \mathfrak{R}$, y los polos y ceros de la función de transferencia del modelo se encuentran dentro del círculo unitario. Asimismo, f^* representa el valor de f en la etapa k , basado en las entradas y salidas observadas hasta el instante $k-1$. f^* se determina actualizando los pesos de una Red Neural utilizando el error de identificación

$$\begin{aligned} e_i(k) &= x(k) - x^*(k) \\ e_i(k) &= f[x(k-1), x(k-2), \dots, x(k-n)] - f_{k-1}^*[x(k-1), x(k-2), \dots, x(k-n)] \end{aligned} \quad \dots 4.7$$

en el método de Back-Propagation. La entrada de control $u(k)$ es modificada de la siguiente manera:

$$u(k) = \sum_{i=0}^{n-1} \alpha_{mi} x_m(k-i) + \sum_{j=0}^{m-1} \beta_{mj} r(k-j) - f_k^*[x(k), x(k-1), \dots, x(k-n+1)] \quad \dots 4.8$$

El problema analítico es entonces demostrar que la salida $x(k)$ del proceso tiende asintóticamente hacia la salida del modelo de referencia. Si el Modelo I es utilizado para representar el proceso de tal manera que

$$x(k+1) = f[x(k), x(k-1), \dots, x(k-n+1)] + g[u(k)] \quad \dots 4.9$$

un procedimiento similar puede ser adoptado si, para todo el rango de interés, existe la inversa derecha de g (denotada g^+), de tal manera que $gg^+ + u(k) = u(k)$. En tal caso, g^+ debe de ser estimada en línea, para su aplicación en el controlador. Por otro lado, los modelos III y IV, en donde las ecuaciones que describen el proceso son, respectivamente,

$$x(k+1) = f[x(k), x(k-1), \dots, x(k-n+1)] + g[u(k), u(k-1), \dots, u(k-m+1)] \quad \dots 4.10$$

$$x(k+1) = f[x(k), x(k-1), \dots, x(k-n+1); u(k), u(k-1), \dots, u(k-m+1)] \quad \dots 4.11$$

son considerablemente más difíciles de controlar de manera teórica.

4.6. REDES NEURALES COMO CONTROLADORES ADAPTIVOS.

Las aplicaciones de las Redes Neuronales, por su naturaleza como paradigma computacional, y por lo expuesto en las secciones anteriores, se pueden enfocar al control de sistemas lineales y no lineales. A continuación se presentan cuatro clases de problemas en donde se requiere de control no lineal, y en donde las Redes Neuronales pueden ser de gran aplicabilidad; en cada caso se asume que el proceso está descrito por una ecuación diferencial compleja y no lineal.

La primera clase incluye sistemas cuyas ecuaciones de comportamiento se encuentran completamente especificadas. En este caso, el controlador puede ser diseñado cuando menos en teoría. Pero, debido a que la determinación de los parámetros del controlador puede ser demasiado compleja para propósitos prácticos, puede ser preferible el utilizar una Red Neural que realice el procedimiento de identificación.

En otros casos, el sistema puede ser estable, pero puede presentar respuestas de bajo nivel o simplemente no responder ante determinadas entradas. Un controlador lineal puede no cumplir con los objetivos, para condiciones iniciales muy largas. En este caso, la utilización de una Red Neural como Identificador, seguida de un controlador no lineal puede ser efectivo.

En otros casos, como los procesos químicos, el sistema debe de operar de manera eficiente a lo largo de diferentes etapas de equilibrio. En tales ocasiones, el propósito de un controlador adaptivo es el estabilizar el sistema alrededor de uno de dichos estados de equilibrio, dependiendo de la información disponible en ese momento. Aunque existen pocos desarrollos teóricos en esta área, algunos estudios de simulación indican que controladores basados en Redes Neuronales tienen amplias perspectivas de éxito.

En muchas situaciones, el control de un sistema debe ser llevado a cabo con base en los patrones de entrada. En tales circunstancias, el espacio-estado se particiona en regiones separadas, que resultan equivalentes para propósitos de control. Las Redes Neuronales pueden ser utilizadas para generar la entrada óptima a cada región.

CAPÍTULO 5

DESARROLLO EXPERIMENTAL

ESTA TESIS NO DEBE
SALIR DE LA BIBLIOTECA

5. DESARROLLO EXPERIMENTAL

5.1. INTRODUCCIÓN

5.1.1. PLANTEAMIENTO

Se puede plantear una solución alternativa al *Problema Fundamental del Control* (PFC) mediante la utilización de sistemas de control basados en Redes Neuronales.

5.1.2. OBJETIVOS

Los objetivos fundamentales de la presente tesis se derivan del plantear una solución alternativa al *Problema Fundamental del Control*. Estos objetivos son los siguientes:

1. Que una Red Neural emule el comportamiento de un método de control ya probado. Para fines de la presente tesis, se eligió como controlador modelo el algoritmo PID, debido a la gran importancia que ha adquirido en el control de procesos industriales como solución del PFC.
2. Que el comportamiento de la Red Neural resulte satisfactorio, bajo los criterios de estabilidad, repetibilidad y versatilidad.
3. Que tanto el desempeño de la Red Neural como su naturaleza como algoritmo presente ventajas sobre el algoritmo utilizado como modelo.

El cumplimiento de los objetivos arriba presentados permitirá discernir si la utilización de una Red Neural puede servir como una solución alternativa del *Problema Fundamental del Control*.

De manera general, lo que se pretende es modificar el arreglo típico del diagrama de bloques para el controlador PID (figura 5.1)



FIGURA 5.1
DIAGRAMA DE BLOQUES PARA EL CONTROLADOR PID

a un esquema en el que se sustituya el controlador por una Red Neural (figura 5.2).



FIGURA 5.2
DIAGRAMA DE BLOQUES PARA UNA RED NEURAL ACTUANDO COMO CONTROLADOR

5.1.3. METODOLOGÍA GENERAL

En general, la metodología utilizada fue la siguiente:

ENTRENAMIENTO

Se presentó a la red el comportamiento de un controlador PID y su actuación sobre un proceso determinado (figura 5.3). La red fue entrenada de esta manera para emular el desempeño del PID.

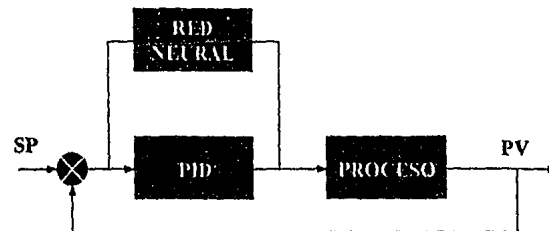


FIGURA 5.3
PRESENTACIÓN DEL COMPORTAMIENTO DE UN PID ANTE UNA RED NEURAL

PRUEBA

Una vez terminado el aprendizaje, se presentó ante la red tanto el mismo proceso como otros procesos diferentes al utilizado en el entrenamiento, obteniéndose así puntos de comparación entre la actuación de la red y la del controlador.

En las siguientes secciones se detalla el desarrollo experimental seguido durante la presente tesis:

Sección 5.2.

Preliminares:

- Modelos de los procesos utilizados para el entrenamiento
- Algoritmo PID empleado como controlador
- Método de sintonización
- Red neural utilizada como controlador alternativo.

Sección 5.3.

Entrenamiento.

Sección 5.4.

Pruebas y resultados.

Sección 5.5.

Análisis de resultados.

5.2. PRELIMINARES

5.2.1. PROCESOS DINÁMICOS

Los procesos utilizados durante el desarrollo de la presente tesis fueron los siguientes:

- Procesos de primer orden
- Procesos de segundo orden
- Procesos con tiempo muerto

5.2.1.1. Procesos de primer orden

Ante un cambio tipo escalón en la entrada, la salida de un proceso de primer orden típico sigue un crecimiento exponencial. El modelo, los parámetros y la función de transferencia de este tipo de procesos son:

Modelo:
$$\tau \frac{dy}{dt} + y = K_P x$$
 ...5.1

Función de transferencia:
$$G_p(s) = \frac{Y(s)}{X(s)} = \frac{K_P}{\tau s + 1}$$
 ...5.2

donde:

- x = entrada al proceso
 y = salida del proceso
 K_P = ganancia del proceso
 τ = constante de tiempo

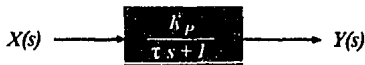


FIGURA 5.4
DIAGRAMA DE BLOQUES PARA UN PROCESO DE PRIMER ORDEN

Los parámetros dinámicos de un proceso de primer orden son la ganancia K_p (cambio en la salida/cambio en la entrada) y la constante de tiempo τ . Tradicionalmente, la constante es el tiempo en el cual se alcanza el 63.2% del valor final de la respuesta a un cambio fijo en la entrada (p.e., un cambio tipo escalón).

En la figura 5.5 se presenta la respuesta de un proceso de este tipo ante un cambio tipo escalón en la entrada.

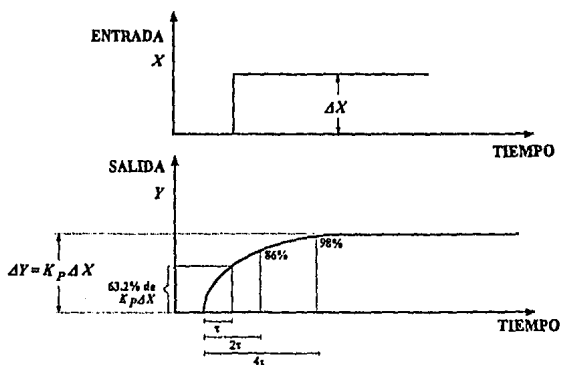


FIGURA 5.5
RESPUESTA DE UN PROCESO DE 1^{ER} ORDEN ANTE UNA ENTRADA ESCALÓN

Entre los procesos de primer orden se pueden incluir a la gran mayoría de los transmisores industriales, elementos finales de control y muchos procesos.

En la siguiente figura (5.6) se presenta un proceso típico de primer orden: un tanque agitado, en donde el flujo de entrada es igual al flujo de salida.

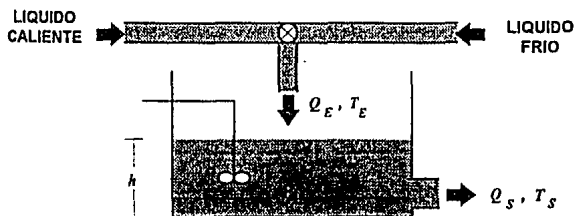


FIGURA 5.6
EJEMPLO DE UN PROCESO DE PRIMER ORDEN

En la figura anterior:

Entrada: Temperatura de entrada T_E

Salida: Temperatura de salida T_S (asumiendo que el mezclado es homogéneo)

La función de transferencia para este proceso es:

$$\frac{T_S(s)}{T_E(s)} = \frac{1}{\tau s + 1}$$

donde:

τ = tiempo de residencia en el tanque

$$\tau = \frac{\text{volumen del tanque}}{\text{flujo volumétrico}}$$

$$\tau = \frac{hA}{Q_S}$$

h y A son, respectivamente, el nivel constante del líquido en el tanque y el área de la sección transversal del tanque.

5.2.1.2. Procesos de segundo orden

Algunos procesos son inherentemente de segundo orden, tales como los sistemas masa-resorte, los sistemas capacitor-inductor, manómetros de tubos "U", etc. Tales procesos tienen las siguientes características:

Modelo:

$$\tau^2 \frac{d^2 y}{dt^2} + 2\zeta \tau \frac{dy}{dt} + y = K_P x \quad \dots 5.3$$

Función de transferencia:

$$G_P(s) = \frac{Y(s)}{X(s)} = \frac{K_P}{\tau^2 s^2 + 2\zeta \tau s + 1} \quad \dots 5.4$$

donde:

- $x =$ entrada al proceso
- $y =$ salida del proceso
- $K_P =$ ganancia del proceso
- $\tau =$ constante de tiempo
- $\omega = \tau^{-1} =$ constante de tiempo
- $\zeta =$ factor de amortiguamiento

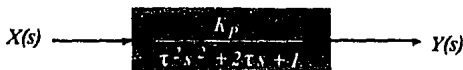


FIGURA 5.7A
DIAGRAMA DE BLOQUES PARA UN PROCESO DE SEGUNDO ORDEN

Dos procesos de primer orden en serie, en donde la salida del primero constituye la entrada del segundo, pueden ser descritos con un modelo de proceso de segundo orden, tal como se muestra en la figura 5.7B:

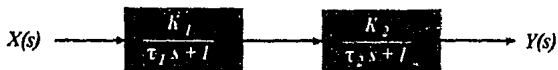


FIGURA 5.7B
DIAGRAMA DE BLOQUES PARA UN PROCESO DE SEGUNDO ORDEN (DESARROLLADO)

El proceso descrito en la figura anterior se representa por medio de la siguiente ecuación diferencial (modelo de 2° orden):

$$\tau_1\tau_2 \frac{d^2y}{dt^2} + [\tau_1 + \tau_2] \frac{dy}{dt} + y = K_1K_2x$$

La comparación de esta ecuación con el modelo normal de un proceso de segundo orden arroja los siguientes resultados:

$$\zeta = \text{factor de amortiguamiento} = \frac{1}{2} \frac{\tau_1 + \tau_2}{\sqrt{\tau_1\tau_2}} \geq 1.0 \text{ para procesos de primer orden en serie}$$

$$t = \text{constante de tiempo} = \sqrt{\tau_1\tau_2}$$

$$K_p = K_1K_2 = \text{ganancia del proceso}$$

En la figura 5.8 se presenta la respuesta de un proceso de este tipo ante un cambio tipo escalón en la entrada.

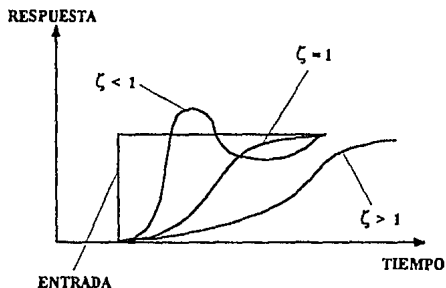


FIGURA 5.8
RESPUESTA DE UN PROCESO DE 2° ORDEN ANTE UNA ENTRADA ESCALÓN

5.2.1.3. Procesos con tiempo muerto

En este tipo de procesos, lo que sucede en la entrada se repite en la salida después de transcurridas θ_d unidades de tiempo, donde θ_d es el tiempo muerto. Por ejemplo, considérese el flujo q de líquido a través de una tubería aislada de sección transversal A y longitud L .

Durante un estado estacionario, la temperatura x del líquido a la entrada será la misma que a la salida. Si la temperatura a la entrada cambia, dicho cambio no será detectado en la salida sino hasta θ_d segundos más tarde. Este tiempo muerto es simplemente el tiempo requerido para que el líquido fluya de la entrada a la salida de la tubería, y está dado por:

$$\theta_d = \frac{\text{Volumen de la tubería} \left[\frac{\text{m}^3}{\text{s}} \right]}{\text{Flujo volumétrico} \left[\frac{\text{m}^3}{\text{s}} \right]} = \frac{LA}{q} \quad \dots 5.5$$

El modelo del proceso está dado por:

$$y(t) = x(t - \theta_d) \quad \dots 5.6$$

donde y es la temperatura de salida. Para un proceso estacionario:

$$y_S = x_S \quad \dots 5.7$$

Las últimas dos ecuaciones permiten obtener un modelo en términos de las variables de desviación:

$$Y(t) = y(t) - y_S = x(t - \theta_d) - x_S = X(t - \theta_d) \quad \dots 5.8$$

Por medio de la transformada de Laplace se obtiene la función de transferencia del elemento de tiempo muerto:

$$G_P(s) = \frac{Y(s)}{X(s)} = EXP[-\theta_d s]$$

...5.9

La respuesta Y para una entrada arbitraria X se muestra en la figura 5.9.

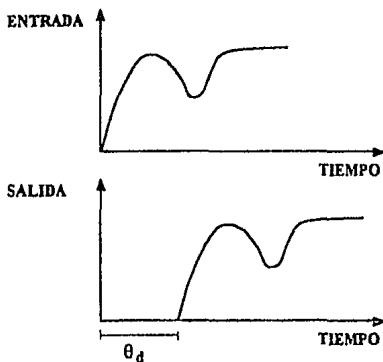


FIGURA 5.9
RESPUESTA DE UN ELEMENTO DE TIEMPO MUERTO

5.2.2. CONTROLADOR PID

La forma clásica de la ecuación del controlador PID se presentó en la ecuación 2.9:

$$C(t) = K \left[E(t) + \frac{1}{T_i} \int_0^t E(t) dt + T_d \frac{dE(t)}{dt} \right] \quad \dots 2.9$$

donde:

$C(t)$ = variable de control

$E(t)$ = error

T_i = tiempo integral

T_d = tiempo derivativo

Esta misma expresión, pero en el dominio de la frecuencia, se presentó en la ecuación 2.8:

$$G_C(s) = K \left[1 + \frac{1}{T_i s} + T_d s \right] \quad \dots 2.8$$

Durante el desarrollo de la presente tesis, se utilizó una forma alternativa de esta última ecuación. Concretamente, la forma utilizada es la que tradicionalmente utiliza el sistema de control TDC-3000 de la compañía HONEYWELL™; la razón para ello fue que es una ecuación diseñada para aplicaciones reales, en las que su eficiencia ha sido comprobada. Esta versión del controlador PID es la siguiente:

$$G_C = K_C \left[1 + \frac{1}{T_i s} \right] \left[\frac{T_d s + 1}{\alpha T_d s + 1} \right] \quad \dots 5.10$$

donde α es un parámetro empírico.

5.2.3. CONVERSIÓN DE FUNCIONES CONTINUAS EN EL DOMINIO DE LA FRECUENCIA A FUNCIONES DISCRETAS EN EL DOMINIO DEL TIEMPO: TRANSFORMADA z

Los sistemas de tiempo discreto son sistemas dinámicos en los cuales una o más variables pueden variar solamente en ciertos instantes, y difieren de los sistemas de tiempo continuo en que las señales aparecen en forma de datos muestrales. Muchos sistemas de control industrial son sistemas discretos porque invariablemente influyen en su comportamiento algunos elementos cuyas entradas y/o salidas son discretas en el tiempo. En estos sistemas, en general, no es práctico funcionar a régimen permanente, debido a que pueden ocurrir cambios en las necesidades de producción, materias primas, factores económicos, etc. Por ello, puede considerarse un comportamiento transitorio de los procesos industriales.

La transformada z permite convertir una función continua en una función discreta en el tiempo. Se han desarrollado varios métodos para obtener funciones de transferencia en el dominio de z a partir de funciones de transferencia en el dominio de s . Estas transformaciones son realizadas mediante la sustitución de funciones de z por los términos s^q en la función de transferencia en el dominio de s . Uno de los más populares métodos de aproximación es el algoritmo de Tustin ⁵. La función de z que es substituida para s^q al implementar la transformación de Tustin es:

$$s^q \cong \left[\frac{2}{\tau} \times \frac{1-z^{-1}}{1+z^{-1}} \right]^q$$

...5.11

La transformada de Tustin fue utilizada en el desarrollo de esta tesis para transformar las ecuaciones de los procesos de primer y segundo orden, con tiempo muerto y del controlador PID -todas ellas en función de la variable s - a ecuaciones en función de z , con el fin de permitir el trabajar con muestreos discretos.

(5) Houptis, C.H.; Lamont, G. DIGITAL CONTROL SYSTEMS
McGraw-Hill Book Company, E.U.A., 1987

Aplicando la transformación de Tustin a cada una de las ecuaciones mencionadas, y después de transformar las funciones de z a funciones en el tiempo k (considerando que: $z \rightarrow k$, $z^{-1} \rightarrow k-1$, $z^{-2} \rightarrow k-2, \dots, z^{-n} \rightarrow k-n$) se obtuvieron las siguientes expresiones, en las cuales:

- Y = valor de la salida
- X = valor de la entrada
- K_P = ganancia del proceso
- K_C = ganancia del controlador
- k = instante de muestreo
- τ = constante de tiempo
- θ = tiempo muerto
- I = tiempo integral
- D = tiempo derivativo

5.2.3.1. Proceso de primer orden

$$Y(k) = \frac{I}{2I_1 + \tau} [K_P \tau X(k) + K_P \tau X(k-1) + (2I_1 + \tau)Y(k-1)] \quad \dots 5.12$$

5.2.3.2. Proceso de segundo orden

$$Y(k) = \frac{I}{C} [FX(k-2) + GX(k-1) + HX(k) - AY(k-2) - BY(k-2)] \quad \dots 5.13$$

donde:

$$A = 4I_1I_2 - 2I_2\tau - 2I_1\tau + \tau^2$$

$$B = -8I_1I_2 + 2\tau^2$$

$$C = 4I_1I_2 + 2I_2I_1 + 2I_1\tau + \tau^2$$

$$F = K_P \tau^2$$

$$G = 2K_P \tau^2$$

$$H = K_P \tau^2$$

5.2.3.3. Proceso con tiempo muerto

$$Y(k) = \frac{I}{C} [FX(k-\theta-2) + GX(k-\theta-1) + HX(k) - AY(k-2) - BY(k-2)]$$

...5.14

donde:

$$A = 4I_1I_2 - 2I_2\tau - 2I_1\tau + \tau^2$$

$$B = -8I_1I_2 + 2\tau^2$$

$$C = 4I_1I_2 + 2I_1\tau + 2I_2\tau + \tau^2$$

$$F = K_p\tau^2$$

$$G = 2K_p\tau^2$$

$$H = K_p\tau^2$$

5.2.3.4. Controlador PID

$$Y(k) = \frac{I}{H} [AX(k-2) + BX(k-1) + CX(k) - FY(k-2) - GY(k-2)]$$

...5.15

donde:

$$A = K_C (4ID - 2I\tau - 2D\tau + \tau^2)$$

$$B = K_C (-4ID + 2\tau^2)$$

$$C = K_C (4ID + 2I\tau + 2D\tau + \tau^2)$$

$$F = 4\alpha ID - 2I\tau$$

$$G = -4\alpha ID$$

$$H = 4\alpha ID + 2I\tau$$

5.2.4. SINTONIZACIÓN DEL CONTROLADOR Y VARIACIONES EFECTUADAS EN LAS ENTRADAS

5.2.4.1. Sintonización

Cada una de los procesos (ecuaciones 5.12, 5.13 y 5.14) fueron controlados con el PID (ecuación 5.15). Los anterior se realizó mediante el empleo de una hoja de cálculo del programa EXCEL 4.0 de MICROSOFT™. El método de sintonización fue el siguiente⁶:

1. Colocar la acción integral del controlador en un valor máximo y la acción derivativa en un valor mínimo.
2. Se define un término alternativo para la ganancia del controlador: la banda proporcional PB , tal que $PB = 100/KC$; KC se fija entonces en un valor tal que permita que PB sea grande. Para ello el valor de KC debe de ser pequeño. Observar el comportamiento del proceso.
3. Incrementar el valor de KC hasta que el proceso muestra una gran oscilación.
4. Reducir este último valor en un 50%, y empezar a disminuir el valor del tiempo integral hasta que el comportamiento del lazo de control se amortigüe considerablemente.
5. Fijar el tiempo integral en el doble de este último valor, e incrementar la acción derivativa hasta que el error entre el punto de ajuste y la variable de proceso alcanza un valor igual o menor del 10%.

5.2.4.2. Cambios efectuados en la entrada

Para cada proceso con el controlador se utilizaron diez tipos diferentes de cambios en la entrada. Dichos cambios fueron los siguientes:

Tipos de cambio en la entrada

- Escalón unitario
- Entrada aleatoria (*random*)
- Rampa
- Pulso regular
- Pulso regular al 50%
- Pulso "pico"
- Pulso piramidal
- Parabólico
- Sinusoidal
- Sinusoidal rectificado

Estos cambios fueron escogidos debido a que constituyen los tipos de entradas más comunes en los procesos industriales.

La siguiente tabla (tabla 5.1) presenta un fragmento de la hoja de cálculo en donde se realizó la sintonización del controlador para el proceso de primer orden, mostrándose la parte superior de la respuesta del lazo de control ante una variación del punto de ajuste tipo escalón unitario. En la misma hoja se desarrollaron los demás tipos de entradas mencionadas arriba. Hojas similares se desarrollaron para cada uno de los procesos.

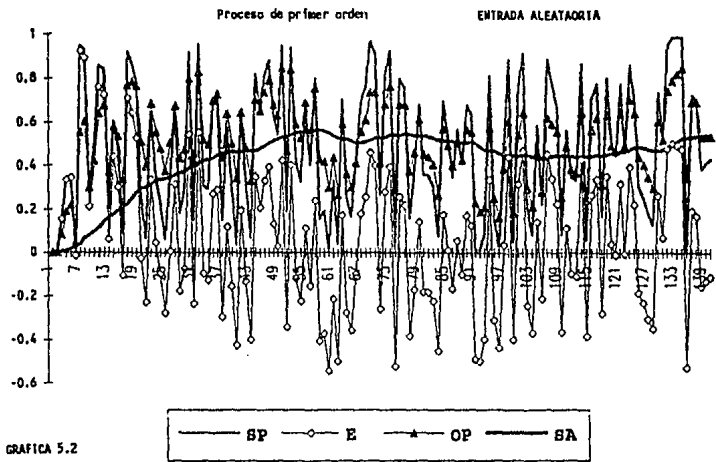
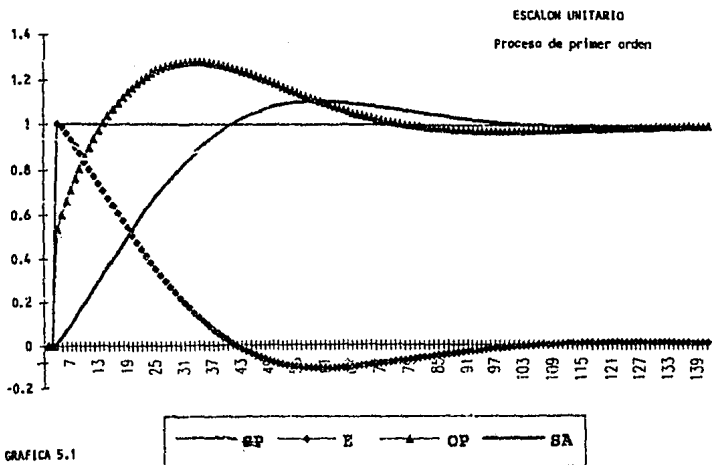
En cada caso, la nomenclatura es la siguiente:

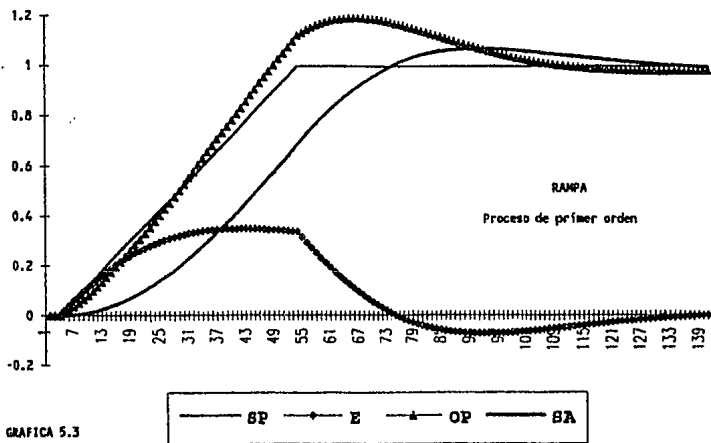
- SP:** Punto de ajuste (*set point*)
- E:** Error
- OP:** Salida del controlador
- SA:** Salida del proceso (variable de proceso)
- MP:** Valor máximo del error
- ICE:** Integral del error cuadrático ($ICE = \int_0^{\infty} [E(t)]^2 dt$).

En las gráficas siguientes se presentan estas entradas para el proceso de primer orden.

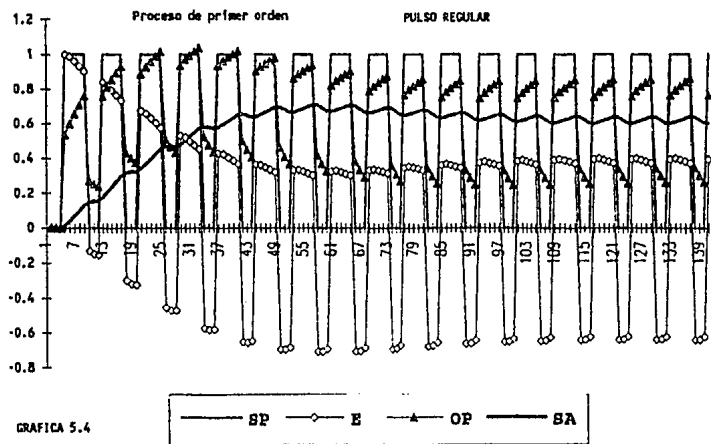
PROCESO DE PRIMER ORDEN					
PARÁMETROS DEL PROCESO					
KP			1		
T1			20		
TAU			1		
PARÁMETROS DEL CONTROLADOR					
T INTEGRAL		7	A		-6.5
T DERIVATIVO		0	B		1
ALFA		0.1	C		7.5
TAU		1	F		-14
KC		0.5	G		0
MP		10.41%	H		14
ISE		12.07			
1 ESCALON UNITARIO					
	SP	E	OP	SA	ERR CUAD
	0	0	0	0	0
	0	0	0	0	0
	1	1	0.535714286	0.013066202	1
	1	0.986933798	0.600143108	0.040132665	0.974038321
	1	0.959867335	0.656138486	0.068815989	0.9213453
	1	0.931184011	0.709334373	0.098763327	0.867103663
	1	0.901236673	0.759804299	0.129778254	0.81222754
	1	0.870221746	0.807563208	0.161676084	0.757265886
	1	0.838323918	0.852633781	0.194282055	0.702786989
	1	0.805717945	0.895046576	0.227431232	0.649181407
	1	0.772568768	0.93483937	0.26096839	0.596862602
	1	0.73903161	0.972056519	0.29474788	0.546167721
	1	0.70525212	1.006746336	0.326633468	0.497300552
	1	0.671366532	1.038970494	0.362498148	0.45073302
	1	0.637501852	1.068783453	0.396223945	0.406408611
	1	0.603776055	1.096251909	0.429701688	0.364545525
	1	0.570298312	1.121444264	0.462830781	0.325240165
	1	0.537169219	1.14443213	0.495518947	0.28855077
	1	0.504481053	1.165289842	0.527681973	0.254501133
	1	0.472318027	1.18409401	0.559243435	0.223084318
	1	0.440756565	1.200923067	0.590134416	0.18426835
	1	0.409865584	1.215856859	0.620293226	0.167988797
	1	0.379706774	1.228976566	0.649665106	0.144177234
	1	0.216748197	1.274174068	0.807102883	0.046979781
	1	0.192897117	1.276876717	0.82995281	0.037209298
	1	0.17004719	1.27841605	0.85179157	0.028916047

TABLA 5.1
PROCESO DE 1ER ORDEN Y CONTROLADOR PID ANTE UN ESCALÓN UNITARIO EN LA ENTRADA

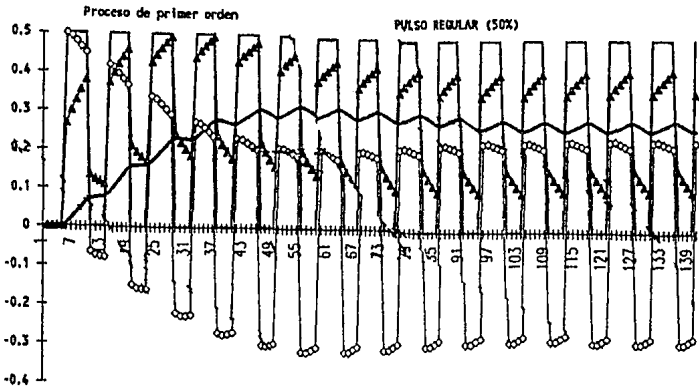




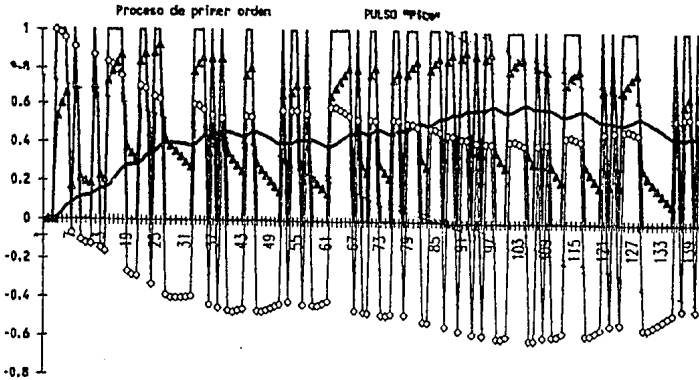
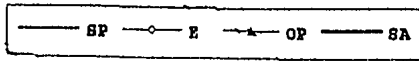
GRAFICA 5.3



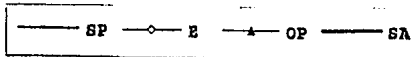
GRAFICA 5.4

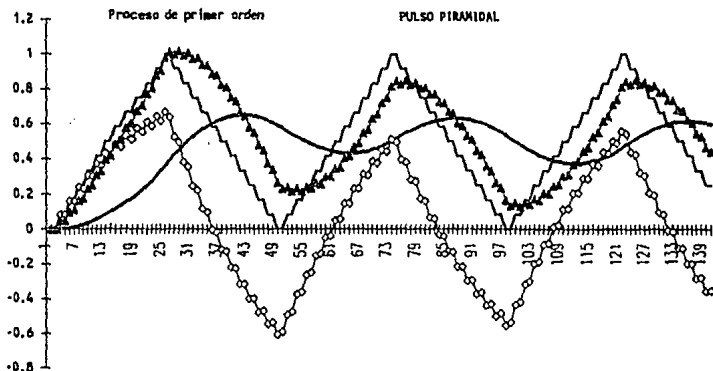


GRAFICA 5.5

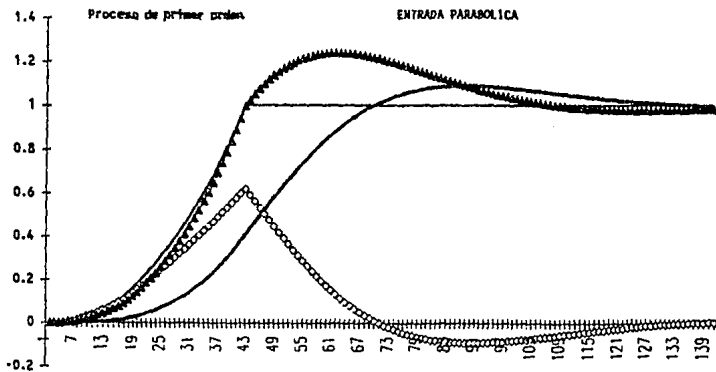
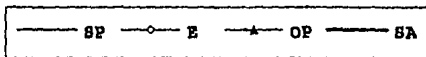


GRAFICA 5.6

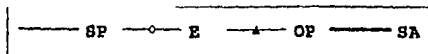


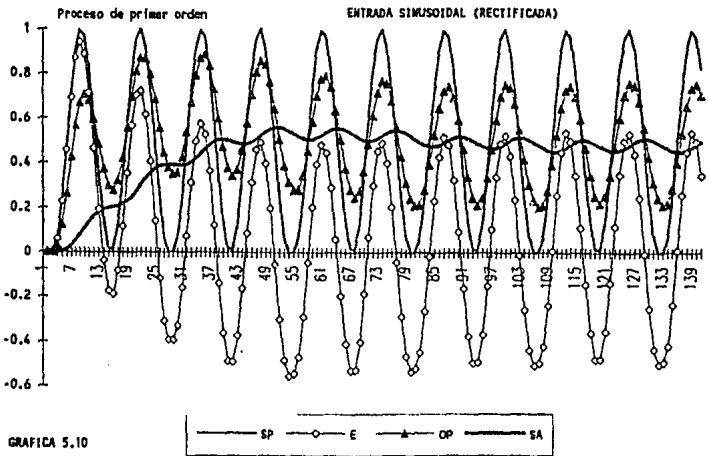
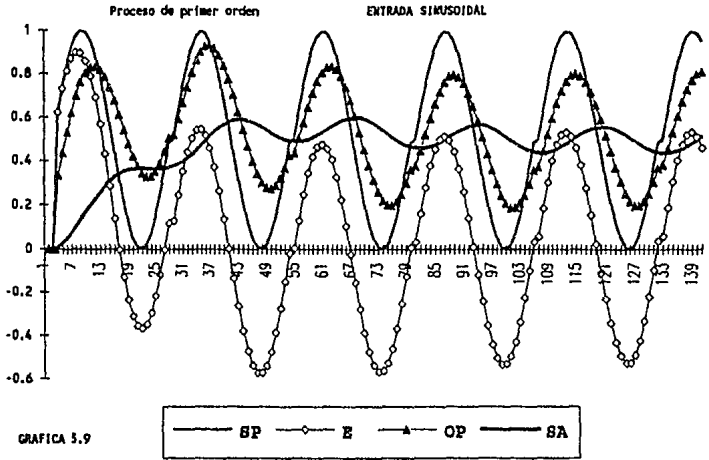


GRAFICA 5.7



GRAFICA 5.8





5.2.5. RED NEURAL

La Red Neural utilizada en la presente tesis fué generada por medio de un programa comercial de computación llamado *NEURALWORKS EXPLORER*, de la compañía *NEURALWARE INC.*™. Este programa permite crear diferentes redes neurales, y escoger diversos métodos de aprendizaje. Dispone de varias interfases (p.e., interfase para hojas de cálculo, interfase para programas en C++, etc.) que permiten que las redes "lean" y "escriban" archivos de diferentes formatos.

En particular, la Red utilizada se encuentra constituida por 18 elementos de procesamiento, acomodados como se muestra en la figura 5.10:

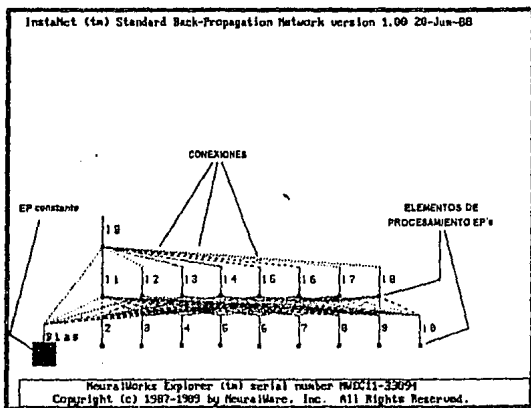


FIGURA 5.10
RED NEURAL UTILIZADA (gráfica generada por el programa NW-EXPLORER)

Como se puede apreciar en la figura 5.10, los elementos de procesamiento se encuentran agrupados en tres diferentes estratos:

ESTRATO DE ENTRADA: Elementos 1, 2, 3, 4, 5, 6, 7, 8, 9, y 10

ESTRATO INTERMEDIO: Elementos 11, 12, 13, 14, 15, 16, 17 y 18

ESTRATO DE SALIDA: Elemento 19

El elemento de procesamiento no. 1 es un EP especial en su estrato, y actúa de manera similar a una tierra en un circuito eléctrico, por lo que se utiliza como parámetro de referencia.

La arquitectura presentada fué diseñada de forma empírica. No existe aún (de acuerdo con la literatura consultada) un método que establezca el número de elementos de procesamiento necesarios para una aplicación en particular, así como tampoco existe un método que determine en el número de estratos y el número de EP's por estrato. Sin embargo, diferentes estudios^{7,8} indican que una Red que emplee como regla de aprendizaje el método de Back-Propagation puede tener como máximo cuatro estratos intermedios para una operación satisfactoria.

Debido a lo anterior, y a que el método de aprendizaje seleccionado fué precisamente el de Back-Propagation, fué necesario realizar diferentes pruebas antes de definir la arquitectura final. Estas pruebas consistieron básicamente en la modificación sucesiva del número de estratos (dentro de los límites mencionados) y el número de EP's por estrato, hasta que el error de la salida de la red respecto a la salida de un lazo de control sobre un proceso de primer orden fué de menos del 1%.

Los parámetros constantes y arbitrarios que se eligieron fueron: número de entradas (diez), número de salidas, y el modo de conexión entre los EP's de un estrato a otro (conexión total). El resultado de esta evaluación es la arquitectura presentada en la figura 5.10.

La forma de operación de los elementos de procesamiento fué descrita en el capítulo 2 de la presente tesis.

(7) Fahmenn, S.E. "AN EMPIRICAL STUDY OF LEARNING SPEED IN BACK-PROPAGATION NETWORKS"

CMU Technical Report, CMU-CS-88-162, junio de 1988

(8) Hecht-Nielsen, R. "NEAREST MATCHED FILTER CLASSIFICATION OF SPATIO-TEMPORAL PATTERNS"

Hecht-Nielsen Neuro-Computer Corporation, E.U.A., junio de 1986

5.3. ENTRENAMIENTO

Tal y como se menciona en la sección 5.1.3., la forma de entrenamiento consistió en presentarle a la Red los diferentes vectores, tanto de entrada como de salida esperada, asociados con los lazos de control PID para los tres procesos. Por medio de una utilidad del programa *NEURALWORKS EXPLORER*, a la Red se le asignó una serie de parámetros, con la información relativa al tipo de red, al método de aprendizaje, y la forma de incorporar los vectores de entrenamiento:

TIPO DE RED: Heteroasociativa
MÉTODO DE APRENDIZAJE: Back-Propagation
FUENTE: *USER-IO* (el usuario señalará los vectores de entrenamiento)
MODALIDAD: *ULOTUS* (para leer los archivos de hojas de cálculo)

Una vez designados dichos parámetros, la Red se encontraba lista para comenzar el entrenamiento. La figura 5.11 presenta el aspecto de la Red antes de comenzar el entrenamiento:

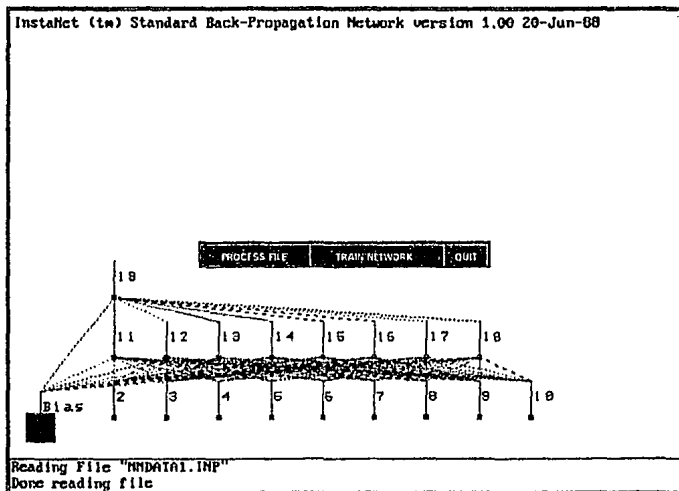
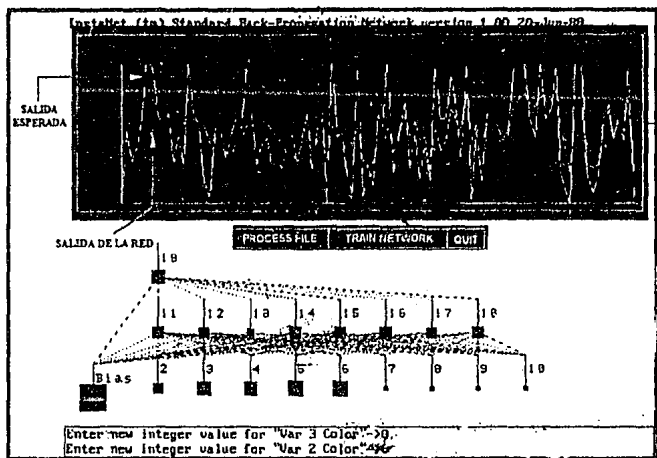


FIGURA 5.11
RED NEURAL ANTES DEL ENTRENAMIENTO (gráfica generada por el programa NW-EXPLORER)

El tamaño relativo de las EP's indican la naturaleza de los pesos. Como se presenta en la figura 5.11, los pesos son totalmente aleatorios. A medida que el aprendizaje se lleva a cabo, los pesos cambian, por lo que el tamaño relativo de los EP's también varía.

Al ejecutarse el aprendizaje, a la Red se le presentaron los diferentes archivos conteniendo los vectores de entrada y salida. La respuesta de la Red ante cada grupo de vectores se presentó a través de diferentes gráficas.

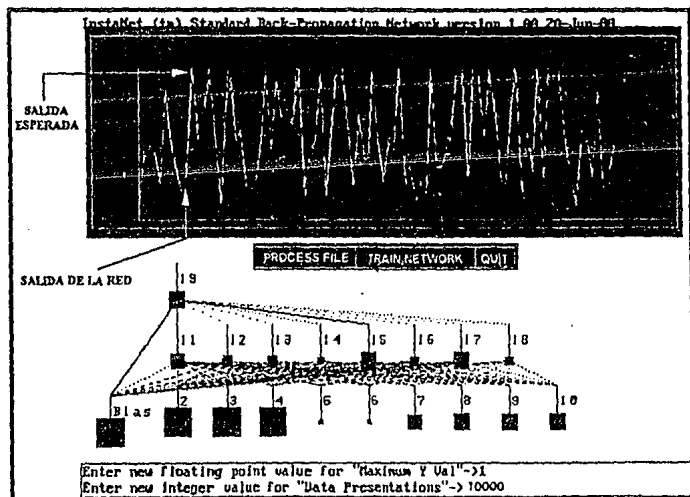
En el primer grupo de gráficas se graficaron tanto la salida esperada como la salida de la Red. Por ejemplo, la gráfica 5.11 presenta lo anterior durante los primeros instantes del entrenamiento de la red utilizando el proceso de segundo orden:



GRAFICA 5.11
RESPUESTA DE LA RED: VECTOR DE SALIDA DE LA RED Y VECTOR DE SALIDA ESPERADO, AL COMIENZO DEL ENTRENAMIENTO

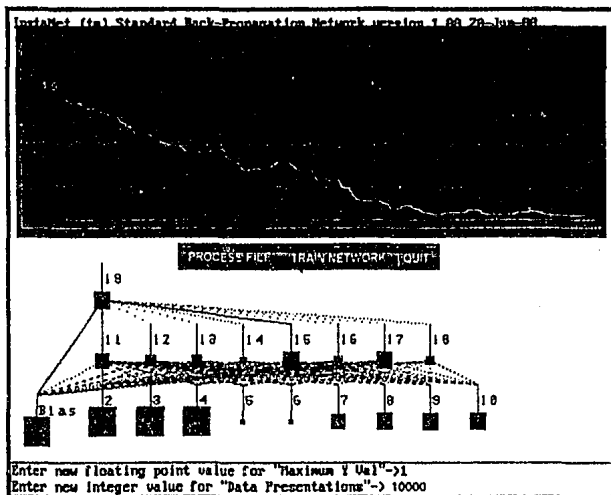
La gráfica 5.12 presenta, por otra parte, el comportamiento de ambos vectores al término del mismo entrenamiento.

Obsérvese el cambio en el tamaño de los diferentes EP's; estos cambios representan el estado de los pesos en la Red.



GRAFICA 5.12
RESPUESTA DE LA RED: VECTOR DE SALIDA DE LA RED Y VECTOR DE SALIDA ESPERADO, AL FINAL DEL ENTRENAMIENTO

La figura 5.13 presenta la gráfica del error cuadrático medio entre los vectores de salida esperada y salida real de la Red:



GRAFICA 5.13
ERROR ENTRE LA SALIDA ESPERADA Y LA SALIDA REAL

Gráficas similares se generaron durante las diferentes sesiones de entrenamiento.

Como resultado del entrenamiento se obtuvieron cuatro Redes Neurales:

- Red Neural 1, resultante del entrenamiento con el proceso de primer orden.
- Red Neural 2, resultante del entrenamiento con el proceso de segundo orden.
- Red Neural 3, resultante del entrenamiento con el proceso con tiempo muerto.
- Red Neural 4, resultante del entrenamiento efectuado con los tres procesos.

Estas cuatro Redes Neurales son similares entre sí en cuanto a arquitectura y características operacionales, pero distintas en lo que a los parámetros se refiere. En efecto, cada Red tiene un diferente grupo de pesos en sus conexiones, correspondiente cada uno al tipo de proceso utilizado. Estos grupos o vectores representan el conocimiento adquirido por la Red a lo largo de las diferentes sesiones de entrenamiento.

5.4. PRUEBAS

Las pruebas realizadas fueron de dos tipos: en el primero, se analizó el resultado del entrenamiento, por medio de la comparación entre las salidas de la Red Neural y las del controlador PID, es decir, la recapitulación. En este primer grupo de pruebas se analizó también la robustez de una Red. En el segundo tipo de pruebas, se compararon las actuaciones de la Red y del PID sobre los diferentes lazos de control utilizados.

5.4.1. RECAPITULACIÓN Y ANÁLISIS DE ROBUSTEZ

Las pruebas de recapitulación tuvieron como objetivo, como se menciona arriba, comparar las salidas de la Red con las salidas obtenidas del controlador PID actuando sobre los diferentes procesos. Esta comparación se realizó graficando ambas salidas, y calculando la integral del error cuadrático ($ICE = \int_0^{\infty} [E(t)]^2 dt$).

Para ello, se utilizaron las salidas de la Red entrenada con un proceso determinado, comparando su actuación tanto con el PID sobre el mismo proceso utilizado en el entrenamiento, como su actuación con el PID sobre los otros procesos no utilizados en el entrenamiento. De esta manera se evaluaría si una Red entrenada con un proceso determinado puede actuar sobre cualquier otro proceso diferente. Además, se realizaron pruebas adicionales de robustez, que consistieron en desactivar tres EP's de la Red y analizar que tanta desviación se presentaba; así se podría determinar si la Red actuaba satisfactoriamente con un 15% de su estructura sin funcionar, lo cual proporciona un criterio para evaluar su posible robustez. De esta manera se ejecutaron cuatro grupos de pruebas:

5.4.1.1. Red 1 (entrenada con el lazo de control para el proceso de primer orden)

1. Salida PID sobre el proceso de primer orden vs Red 1 (Gráfica 5.14).
2. Salida PID sobre el proceso de segundo orden vs Red 1 (Gráfica 5.15).
3. Salida PID sobre el proceso con tiempo muerto vs Red 1 (Gráfica 5.16).
4. Salida PID sobre el proceso con tiempo muerto vs Red 1 con tres EP's desactivados (Gráfica 5.17).

5.4.1.2 Red 2 (entrenada con el lazo de control para el proceso de segundo orden)

1. Salida PID sobre el proceso de primer orden vs Red 2 (Gráfica 5.18).
2. Salida PID sobre el proceso de segundo orden vs Red 2 (Gráfica 5.19).
3. Salida PID sobre el proceso con tiempo muerto vs Red 2 (Gráfica 5.20).
4. Salida PID sobre el proceso con tiempo muerto vs Red 2 con tres EP's desactivados (Gráfica 5.21).

5.4.1.3 Red 3 (entrenada con el lazo de control para el proceso con tiempo muerto)

1. Salida PID sobre el proceso de primer orden vs Red 3 (Gráfica 5.22).
2. Salida PID sobre el proceso de segundo orden vs Red 3 (Gráfica 5.23).
3. Salida PID sobre el proceso con tiempo muerto vs Red 3 (Gráfica 5.24).
4. Salida PID sobre el proceso de primer orden vs Red 3 con tres EP's desactivados (Gráfica 5.25).

5.4.1.4 Red 4 (entrenada con los lazo de control para los tres procesos)

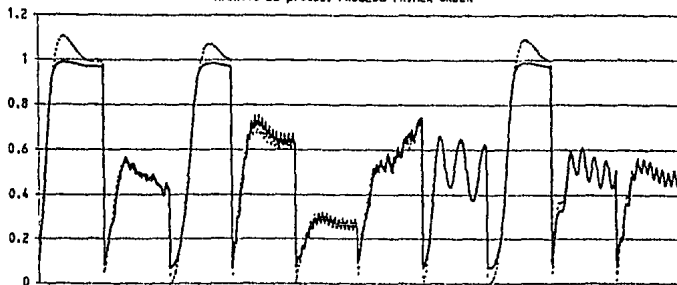
Debido a que esta Red utilizó un conjunto de vectores de entrenamiento que cubrían todos los casos, únicamente se presenta la gráfica de la salida de dicha Red vs el vector de salida esperado (Gráfica 5.28).

Las gráficas de estas pruebas se presentan en las siguientes páginas. En todas ellas la nomenclatura es como sigue:

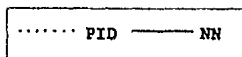
- PID:** Salida del controlador PID
NN: Salida de la Red Neural
ICE: Integral del error cuadrático

Archivo de entrenamiento: PROCESO PRIMER ORDEN

Archivo de prueba: PROCESO PRIMER ORDEN



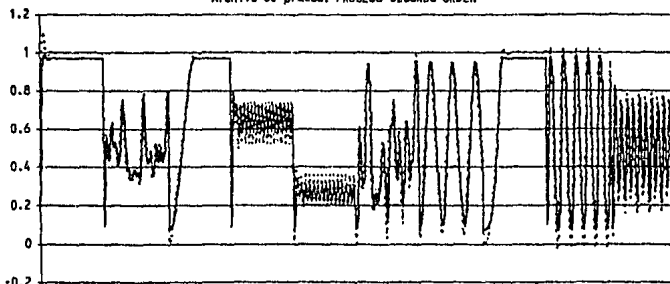
GRAFICA 5.14



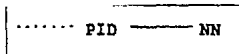
ICE = 1.74

Archivo de entrenamiento: PROCESO PRIMER ORDEN

Archivo de prueba: PROCESO SEGUNDO ORDEN



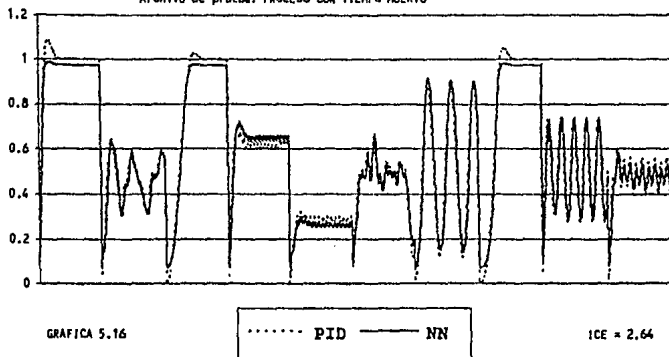
GRAFICA 5.15



ICE = 14.69

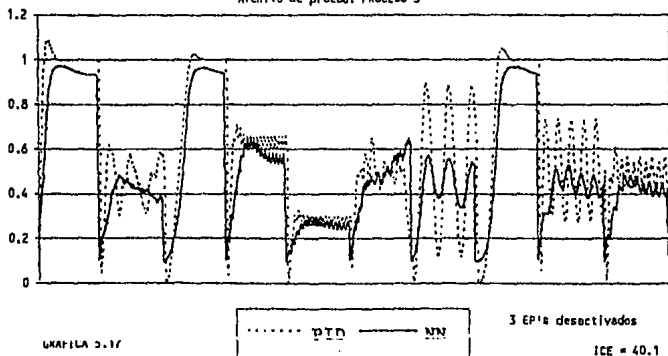
Archivo de entrenamiento: PROCESO 1

Archivo de prueba: PROCESO CON TIEMPO MUERTO



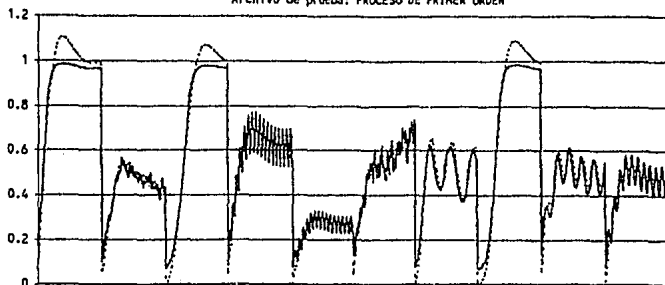
Archivo de entrenamiento: PROCESO PRIMER ORDEN

Archivo de prueba: PROCESO 3

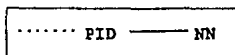


Archivo de entrenamiento: PROCESO SEGUNDO ORDEN

Archivo de prueba: PROCESO DE PRIMER ORDEN



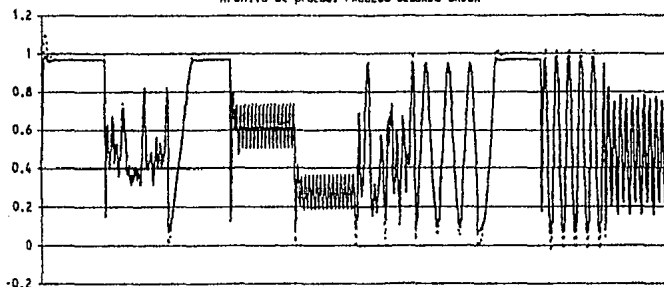
GRAFICA 5.18



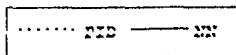
ICE = 3.08

Archivo de entrenamiento: PROCESO SEGUNDO ORDEN

Archivo de prueba: PROCESO SEGUNDO ORDEN



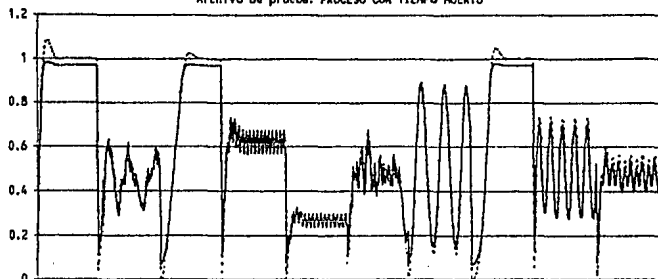
GRAFICA 5.19



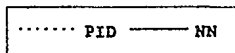
ICE = 1.34

Archivo de entrenamiento: PROCESO SEGUNDO ORDEN

Archivo de prueba: PROCESO CON TIEMPO MUERTO



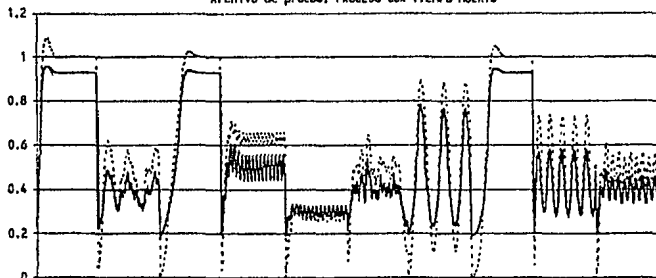
GRAFICA 5.20



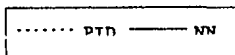
ICE = 1.73

Archivo de entrenamiento: PROCESO SEGUNDO ORDEN

Archivo de prueba: PROCESO CON TIEMPO MUERTO



GRAFICA 5.21

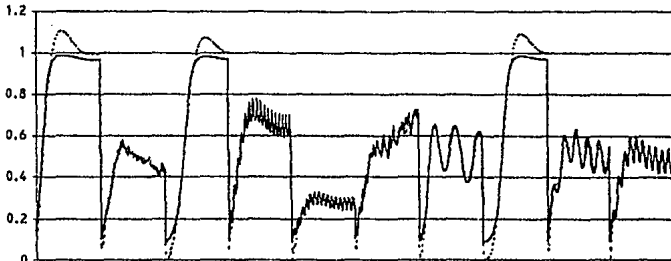


3 EP's Desactivados

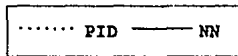
ICE = 12.50

Archivo de entrenamiento: PROCESO CON TIEMPO MUERTO

Archivo de prueba: PROCESO PRIMER ORDEN



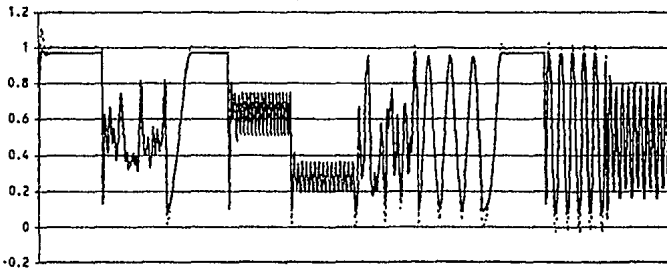
GRAFICA 5.22



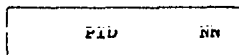
ICE = 2.25

Archivo de entrenamiento: PROCESO CON TIEMPO MUERTO

Archivo de prueba: PROCESO SEGUNDO ORDEN



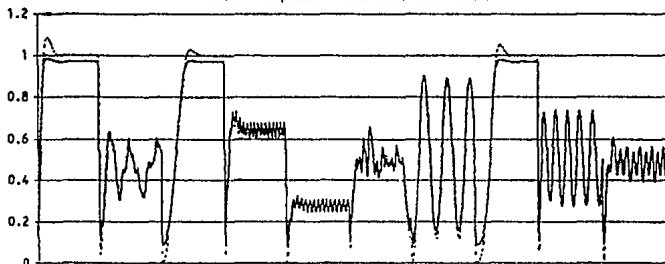
GRAFICA 5.23



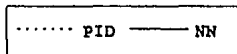
ICE = 4.17

Archivo de entrenamiento: PROCESO CON TIEMPO MUERTO

Archivo de prueba: PROCESO CON TIEMPO MUERTO



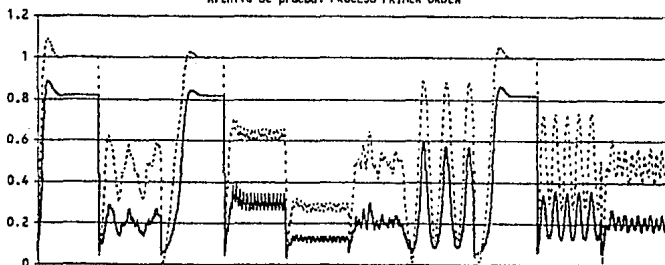
GRAFICA 5.24



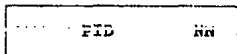
ICE = 1.09

Archivo de entrenamiento: PROCESO CON TIEMPO MUERTO

Archivo de prueba: PROCESO PRIMER ORDEN



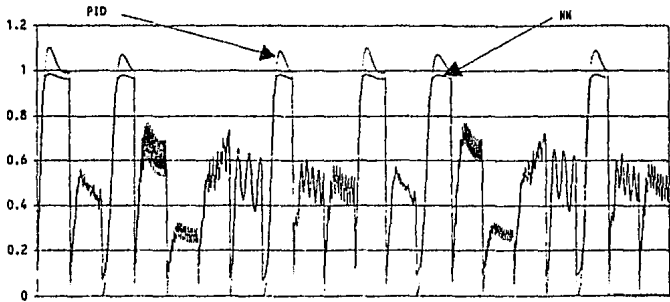
GRAFICA 5.25



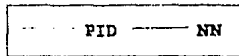
3 PE's DESACTIVADOS

ICE = 81.64

Archivo de entrenamiento: LOS TRES PROCESOS



GRAFICA 5.26



ICE = 5.33

5.4.2. ACTUACION DE LAS REDES NEURALES COMO CONTROLADORES

El segundo conjunto de pruebas tuvo como objetivo el análisis de la actuación de las Redes generadas durante el entrenamiento como controladores de los procesos considerados.

El procedimiento general utilizado para estas pruebas fué el siguiente:

PASO 1.

Utilizar cada una de las Redes como controladores sobre cada uno de los procesos.

PASO 2.

En cada pareja Red i - Proceso j ($i = 1, 2, 3, 4$; $j = 1$ [1ER orden], 2 [2º orden], 3 [tiempo muerto]) se consideraron las variaciones en la entrada ya presentadas:

Tipos de cambio en la entrada

- Escalón unitario
- Entrada aleatoria (random)
- Rampa
- Pulso regular
- Pulso regular el 50%
- Pulso "pico"
- Pulso piramidal
- Parabólico
- Sinusoidal
- Sinusoidal rectificado

PASO 3.

Para cada una de estas variaciones se calculó un valor de ICE.

PASO 4.

De esta manera, se generaron 10 valores de ICE correspondientes a cada una de las variaciones consideradas. A partir de estos valores, se calculó un ICE promedio.

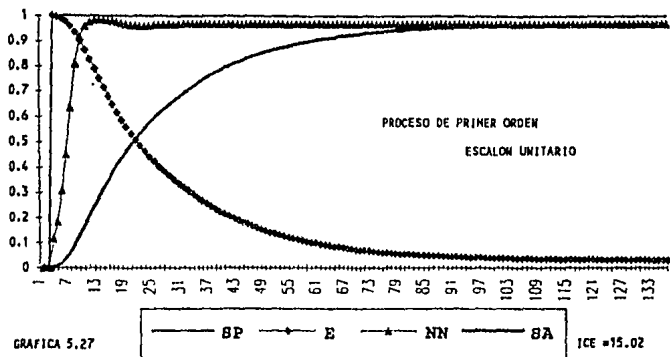
PASO 5.

Así, para cada pareja Red i - Proceso j se obtuvo un valor de ICE promedio.

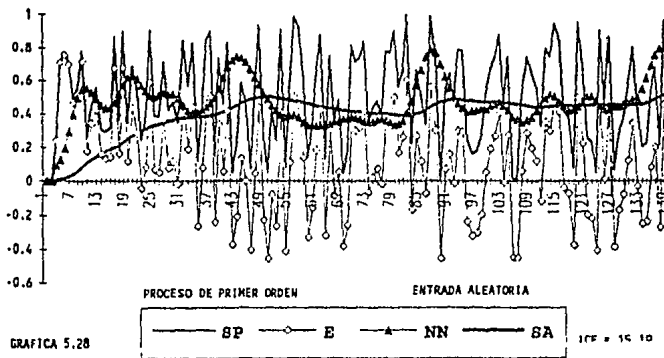
Los resultados de este análisis se compararon posteriormente con los valores de ICE originados por la actuación de los controladores PID.

Las siguientes gráficas presentan la actuación de la Red Neural 2 sobre el proceso de primer orden, con las diez variaciones en la entrada, y el valor del ICE. Gráficas similares fueron desarrolladas para cada pareja Red i - Proceso j .

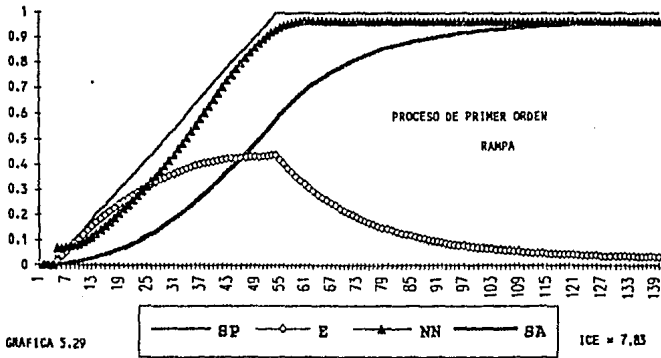
CONTROLADOR: Red Neural 2



CONTROLADOR: Red Neural 2

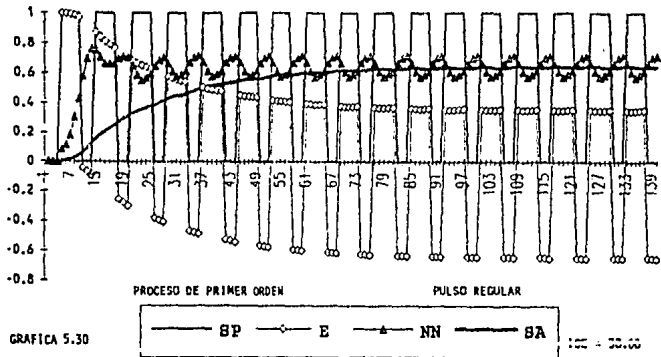


CONTROLADOR: Red Neural 2



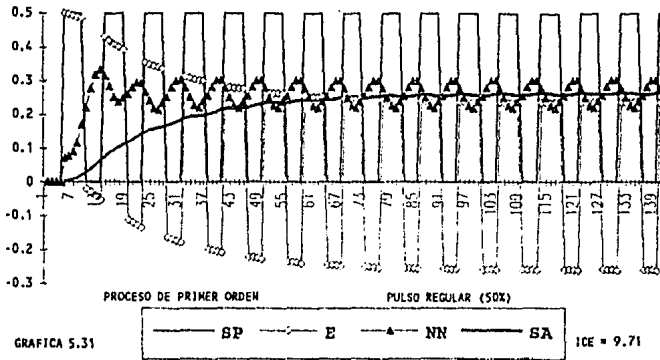
GRAFICA 5.29

CONTROLADOR: Red Neural 2

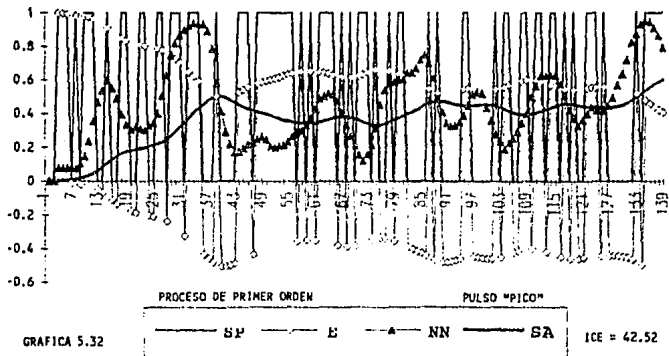


GRAFICA 5.30

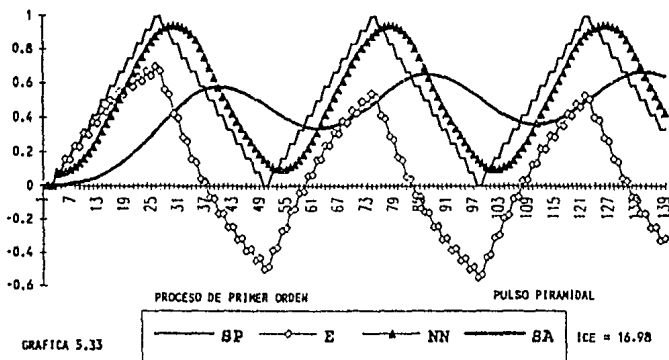
CONTROLADOR: Red Neural 2



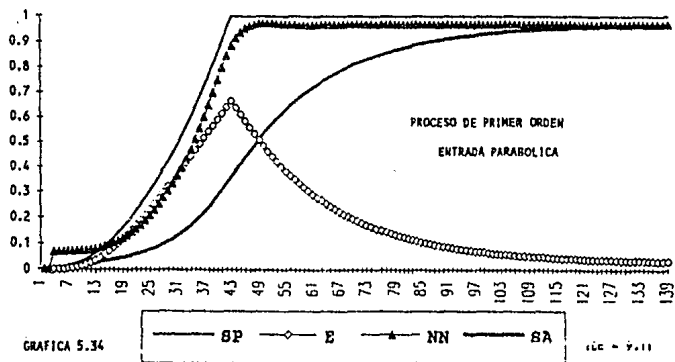
CONTROLADOR: Red Neural 2



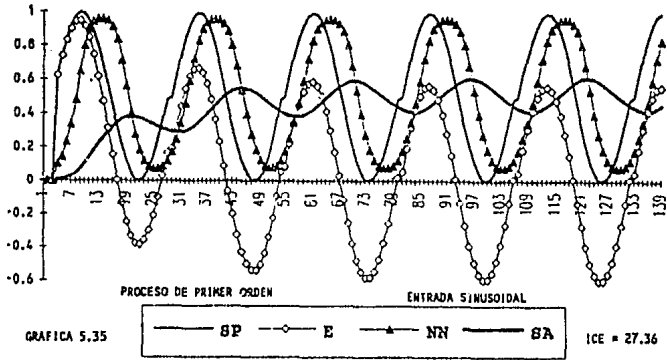
CONTROLADOR: Red Neural 2



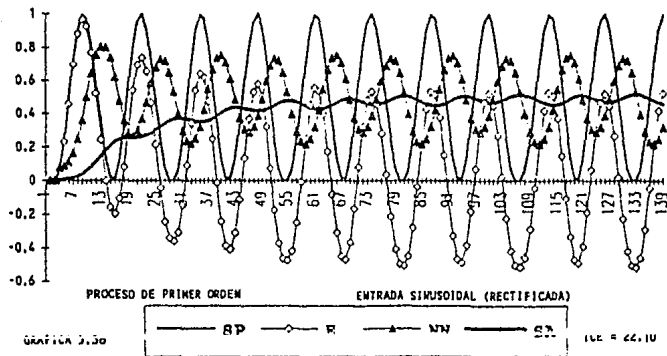
CONTROLADOR: Red Neural 2



CONTROLADOR: Red Neural 2



CONTROLADOR: Red Neural 2



5.5. ANÁLISIS DE RESULTADOS

5.5.1. PRUEBAS DE RECAPITULACIÓN

Los resultados de las pruebas de recapitulación se presentan a continuación:

RED 1 vs PID

PROCESO	ICE
Primer orden	1.74
Segundo orden	14.69
Tiempo muerto	2.64
Tiempo muerto (red con 3 PE's desactivados)	40.10

TABLA 5.2

RED 2 vs PID

PROCESO	ICE
Primer orden	3.08
Segundo orden	1.34
Tiempo muerto	1.73
Tiempo muerto (red con 3 PE's desactivados)	12.50

TABLA 5.4

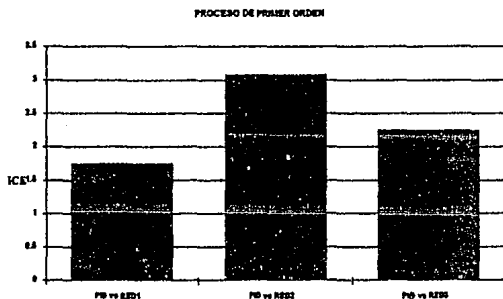
RED 3 vs PID

PROCESO	ICE
Primer orden	2.25
Segundo orden	4.17
Tiempo muerto	1.09
Tiempo muerto (red con 3 PE's desactivados)	81.64

TABLA 5.5

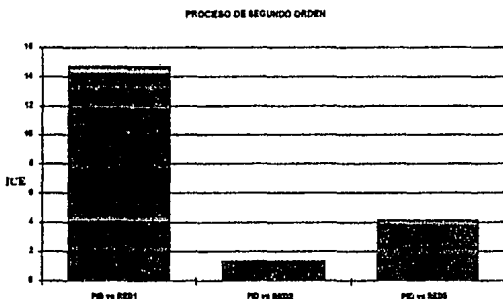
A partir de la información presentada en las tablas anteriores se analizaron los valores de los ICE's correspondientes a las comparaciones entre las actuaciones del controlador PID vs las diferentes redes neurales.

Así, como se muestra a continuación, se pudo observar de manera cuantitativa la diferencia entre los errores de actuación PID-Red. Por ejemplo, para el proceso de primer orden, cada par PID-Red actuó de manera diferente, generando un diferente valor de ICE:

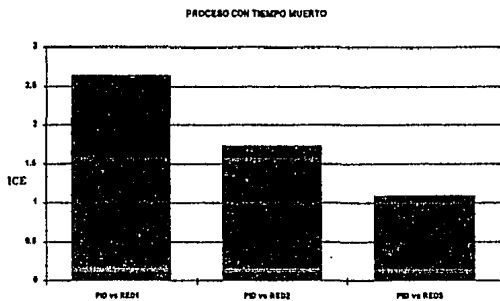


GRÁFICA 5.37
COMPARACIÓN DE LAS ACTUACIONES ENTRE PID Y LAS REDES SOBRE EL PROCESO DE PRIMER ORDEN.

Puede apreciarse fácilmente que el menor valor de ICE se obtuvo al comparar la actuación sobre el proceso de primer orden del PID con la de la Red 1, *la cuál fue entrenada con dicho proceso*. Resultados similares se obtuvieron para los otros dos casos:



GRÁFICA 5.38
COMPARACIÓN DE LAS ACTUACIONES ENTRE PID Y LAS REDES SOBRE EL PROCESO DE SEGUNDO ORDEN.



GRÁFICA 5.39
COMPARACIÓN DE LAS ACTUACIONES ENTRE PID Y LAS REDES SOBRE EL PROCESO CON TIEMPO MUERTO.

Como puede observarse, los valores de los ICE's son bajos. De ésto, y de una apreciación cualitativa de las gráficas generadas, puede concluirse que la recapitulación de las Redes Neuronales, en relación con los archivos de entrenamiento, es satisfactoria. De hecho, en muchas ocasiones los puntos generados por la red coincidían de manera casi total con los puntos generados por el PID. Cuando la coincidencia de puntos resultó baja, lo que se mantuvo fue la tendencia general: el comportamiento de los puntos resultó similar.

Por otro lado, puede apreciarse que la diferencia de actuación entre los PID's y las Redes alcanzó sus valores mas bajos cuando actuaban sobre el proceso con el cuál fue entrenada la red. Sin embargo, cuando la pareja PID-Red actuaban sobre un proceso no utilizado en el entrenamiento de ésta Red, el comportamiento resultaba bastante parecido al esperado, si bien el valor de ICE resultaba mayor. Esto permite establecer que una Red Neuronal puede actuar sobre procesos no utilizados durante el entrenamiento; es decir, puede generalizar y actuar sobre eventos para ella desconocidos.

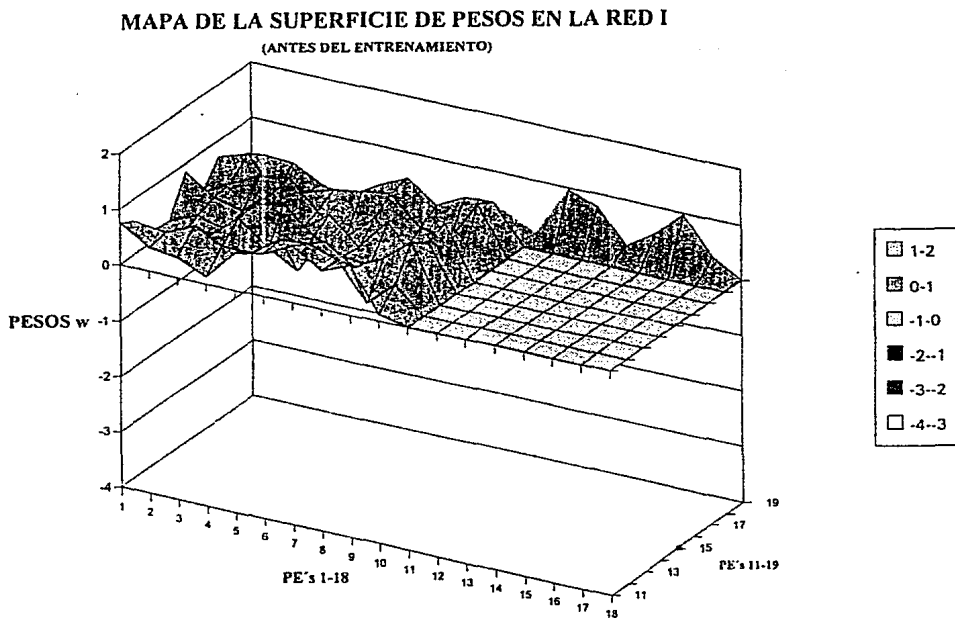
El hecho de que las Redes Neuronales permitan este tipo de generalización constituye, sin duda, una de las principales ventajas que poseen sobre los algoritmos matemáticos convencionales: si una Red va a utilizarse en alguna aplicación determinada, para la cual es rigurosamente entrenada, y si en algún momento de su actuación la aplicación cambia de alguna manera, en ese momento el comportamiento de la Red se degradará sensiblemente, pero permitirá mantener a la aplicación bajo control. De alguna manera, la información que posee la Red le permitirá continuar con su operación.

Desde luego que, al mencionarse que la aplicación cambie de alguna manera, se está hablando de cambios en las condiciones del evento, no un cambio completo. Por ejemplo, una de las técnicas de control avanzado, llamada Control Adaptivo, permite la modificación de los parámetros de un controlador cuando el proceso que sobre el que éste actúa cambia de forma tal que el se encuentre en un punto para el cual dicho controlador no fue diseñado. Las Redes Neuronales, en este contexto, no necesitan ser ajustadas a las nuevas condiciones; la información en ellas almacenada les permite continuar con su desempeño.

Esta información o conocimiento se almacena en la Red en forma de los valores de los pesos de las interconexiones entre los diferentes Elementos de Procesamiento. En las siguientes gráficas, se puede apreciar el cambio de los valores de estos pesos para la Red 1. La gráfica 5.40 muestra la superficie de pesos de la Red 1 antes del entrenamiento; en ese momento, cada interconexión tiene un peso aleatorio. Estos pesos son los que justamente son modificados durante el desarrollo del entrenamiento a través de la técnica de Back-Propagation (ver capítulo 3).

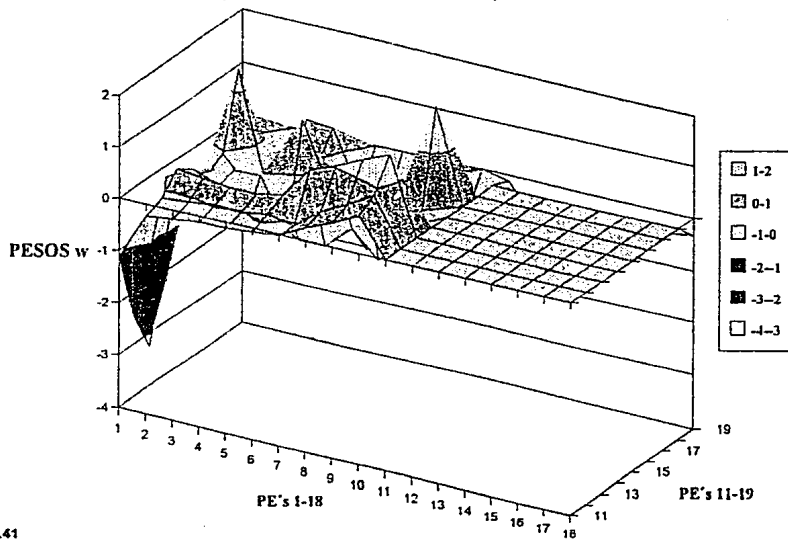
Al término del entrenamiento, este conjunto de pesos se dice que constituye el *conocimiento* de la Red. Si ésta fué entrenada para actuar como controlador PID sobre un proceso de primer orden determinado, el conjunto de pesos *w* le permitirá actuar sobre cualquier otro proceso de primer orden, tal vez no con la misma eficiencia, pero sí con las mismas ventajas generales que su actuación sobre el primer proceso reportó. Lo anterior agiliza de manera notable el desarrollo de los controladores: si se cuenta con una Red que actúa sobre un proceso determinado de orden *n*, la generación de otra Red como controlador de otro proceso del mismo orden únicamente implicará modificaciones en la primera Red, disminuyendo considerablemente el tiempo de entrenamiento.

La gráfica 5.41 presenta la superficie de pesos de la Red 1 al término del entrenamiento. Estas superficies son llamadas también superficies de energía de la Red.



GRÁFICA 5.40

MAPA DE SUPERFICIE DE PESOS EN RED I
(AL TERMINO DEL ENTRENAMIENTO)



GRÁFICA 5.41

Las pruebas de "robustez" se realizaron, como ya se ha mencionado, mediante la desactivación de tres elementos de procesamiento de las redes. Esta desactivación, de acuerdo con la sección 5.4.1. representa el 15% de la estructura de la Red. Las gráficas generadas con las redes degradadas de esta manera, presentaron valores mayores de ICE, aunque el comportamiento neto resultó similar al esperado; es decir, las gráficas no coincidían, pero el centro de oscilación de ambas se conservaba (ver, por ejemplo, la gráfica 5.17).

El hecho de que un controlador actúe de esta manera con un 15% de su estructura puede considerarse como un factor de robustez. Debido a que las Redes Neuronales se forman de varios microprocesadores (uno por elemento de procesamiento), su estructura se puede considerar robusta: aunque uno de los microprocesadores falle, el resto de la red seguirá funcionando. Ésta es una de las principales ventajas de las Redes como controladores sobre otros algoritmos y dispositivos: al tener una memoria distribuida, la pérdida de una unidad no implica la pérdida total de la memoria.

5.5.2. REDES NEURALES COMO CONTROLADORES

Los resultados de las pruebas de actuación de las Redes Neuronales actuando como controladores se presenta a continuación. Como se menciona en la sección 5.4.2., cada una de las cuatro Redes generadas actuaron como controladores de los tres procesos contemplados, utilizándose las diez variaciones a la entrada ya descritas. En cada una de estas variaciones se calculó un valor de ICE, calculándose al final un ICE promedio correspondiente a la actuación de esa Red en particular sobre el proceso.

De esta manera, cada proceso fue controlado tanto por el PID, como por las cuatro Redes Neuronales; cada una de estas actuaciones fué evaluada por el ICE, por lo que se pudo realizar una comparación entre el desempeño de cada uno de los controladores sobre un cada uno de los procesos.

En la siguiente tabla (tabla 5.6) se presenta un resumen de estos resultados.

VALORES DE ICE

PROCESO DE PRIMER ORDEN

ENTRADAS	PID	RED 1	RED 2	RED 3	RED 4
ESCALÓN UNITARIO	12.07	15.32	15.02	16.67	11.85
ALEATORIA	16.48	18.23	15.19	12.88	13.25
RAMPA	18.02	17.30	7.83	25.69	17.30
PULSO REGULAR	14.01	13.50	38.08	28.68	13.50
PULSO 50%	8.25	12.56	9.71	17.22	11.56
PULSO "PICO"	18.25	15.71	42.52	29.70	15.71
PULSO PIRAMIDAL	10.25	10.58	16.98	25.55	18.58
PARABÓLICO	13.80	10.25	9.11	17.40	10.25
SINUSOIDAL	11.26	13.84	27.36	31.05	9.48
SINUSOIDAL RECTIFICADO	14.58	10.84	22.10	25.13	11.69
SUMA	135.03	138.14	203.90	227.75	133.13
ICE PROMEDIO	13.50	13.81	20.39	22.78	13.31

PROCESO DE SEGUNDO ORDEN

ENTRADAS	PID	RED 1	RED 2	RED 3	RED 4
ESCALÓN UNITARIO	12.40	16.34	12.36	18.76	13.37
ALEATORIA	17.16	19.67	15.31	21.32	17.41
RAMPA	24.81	27.21	21.32	16.84	21.26
PULSO REGULAR	19.64	17.87	18.97	19.64	17.76
PULSO 50%	22.78	25.28	20.05	20.25	23.02
PULSO "PICO"	13.54	15.96	21.76	17.31	13.75
PULSO PIRAMIDAL	21.64	24.40	16.64	18.67	22.02
PARABÓLICO	11.33	15.32	16.84	22.26	12.33
SINUSOIDAL	10.35	12.85	20.67	25.46	10.60
SINUSOIDAL RECTIFICADO	16.17	22.56	22.18	23.13	20.45
SUMA	169.82	197.44	188.10	203.54	171.98
ICE PROMEDIO	16.98	19.74	18.81	20.35	17.20

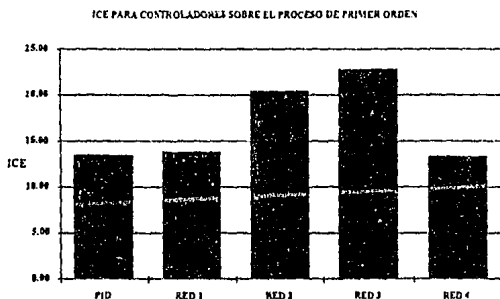
PROCESO CON TIEMPO MUERTO

ENTRADAS	PID	RED 1	RED 2	RED 3	RED 4
ESCALÓN UNITARIO	12.81	20.45	19.78	21.28	18.24
ALEATORIA	16.25	22.27	17.86	20.92	14.56
RAMPA	25.21	24.02	23.27	18.16	21.06
PULSO REGULAR	18.72	21.55	22.22	21.14	17.68
PULSO 50%	15.84	21.14	17.45	17.45	20.31
PULSO "PICO"	16.71	23.56	20.49	18.51	18.51
PULSO PIRAMIDAL	21.88	22.22	25.14	18.04	18.04
PARABÓLICO	14.36	19.93	20.71	20.19	20.19
SINUSOIDAL	13.87	22.59	22.47	18.43	17.92
SINUSOIDAL RECTIFICADO	20.43	24.90	19.87	17.64	20.53
SUMA	176.08	222.62	209.27	191.78	187.04
ICE PROMEDIO	17.61	22.26	20.93	19.18	18.70

TABLA 5.6
VALORES DE ICE PARA LOS DIFERENTES CONTROLADORES

A partir de la información presentada en la tabla 5.6 puede destacarse varios puntos:

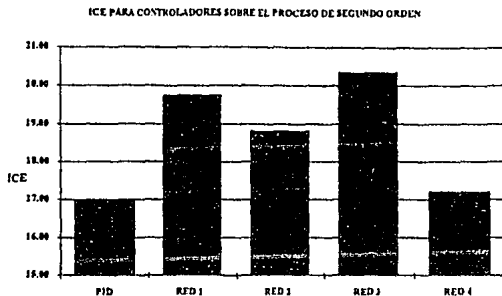
El parámetro central de comparación es justamente el valor del ICE correspondiente al PID; recuérdese que el objetivo es demostrar que una Red Neural puede emular el comportamiento del PID como controlador. Por ejemplo, en el caso del proceso de primer orden, el valor de ICE del PID fue de 13.50. Para las Redes 1, 2, 3 y 4 fueron, respectivamente, 13.81, 20.39, 22.78 y 13.31. Una comparación de estos valores se puede apreciar en la gráfica siguiente:



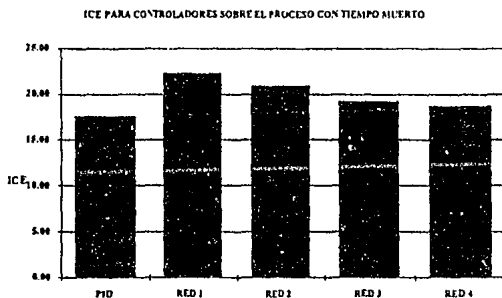
GRÁFICA 5.42

Así, con relación al ICE del PID, el valor del ICE más bajo corresponde a la actuación de la Red 4, la cual es la Red entrenada con los tres procesos. Este valor de ICE es muy parecido al correspondiente a la actuación de la Red 1, la cual fue entrenada con el proceso de primer orden. Esto último se explica fácilmente: es hasta cierto punto obvio que la Red controlará bien el proceso para el cual fue entrenada. Puede observarse que las Redes 2 y 3 se desvían bastante del comportamiento del PID: para el entrenamiento de estas dos Redes no se consideró el proceso de primer orden. En este mismo esquema, puede explicarse que la Red 4 fue la que mejor controló al proceso de primer orden debido a que en su entrenamiento se incluyó a los tres procesos.

Para los otros dos procesos se presentan gráficas similares de comparación de ICE's:



GRÁFICA 5.43



GRÁFICA 5.44

En general, el comportamiento fue el mismo: las Redes que fueron entrenadas con un proceso particular, actuaron como buenos controladores (siempre tomando al PID como criterio de comparación) sobre dichos procesos, presentando desviaciones pronunciadas al actuar sobre procesos no contemplados en su entrenamiento. Por otro lado, el desempeño de la Red 4 -la cual fue entrenada con un mayor conjunto de casos- fue el que en los tres casos se acercó más al comportamiento del controlador PID.

CAPÍTULO 6

CONCLUSIONES

6. CONCLUSIONES

Los niveles de calidad que el entorno económico mundial exige a las Industrias son cada vez mayores. Para alcanzarlos, es necesario utilizar las mas modernas herramientas tecnológicas disponibles. Una parte vital del proceso de producción la constituyen los sistemas de control: en la medida en que éstos se desarrollen y se conviertan en soluciones al *Problema Fundamental del Control*, los niveles de calidad de la producción irán en continuo crecimiento.

Una de las tecnologías que mayor auge ha adquirido en los últimos años es la Inteligencia Artificial; una de sus ramas, la referente al estudio del desarrollo y aplicación de las Redes Neuronales, se ha desarrollado vertiginosamente en diferentes áreas del conocimiento. Las Redes Neuronales, como tecnología de punta, pueden constituir una solución al *Problema Fundamental del Control* si su desempeño como controladores es similar o superior al de otras soluciones para dicho problema.

En este contexto, y a partir del análisis de los diferentes resultados obtenidos durante el desarrollo de la presente tesis, se concluyó lo siguiente:

Las Redes Neuronales pueden emular de manera satisfactoria el comportamiento del controlador de acción proporcional, integral y derivativa PID sobre procesos típicos, como son los procesos de primer y segundo orden, y procesos con tiempo muerto.

El desempeño de las Redes Neuronales tiene las siguientes ventajas sobre el PID:

a. Generalización

Una Red Neural entrenada con un conjunto de vectores de E/S correspondiente a un proceso determinado, puede actuar sobre otro proceso del mismo orden, o bien sobre otro de diferente orden. En el primer caso su actuación es muy parecida, siendo la desviación mínima. En el segundo caso, la desviación es mayor, pero la tendencia se mantiene de forma notable. Los controladores PID únicamente funcionan de manera satisfactoria cuando el proceso sobre el cual actúan se encuentra en las condiciones para las cuales son sintonizados; si el proceso se encuentra en condiciones diferentes, el controlador pierde confiabilidad y precisión, y simplemente no actúan sobre procesos no contemplados en la sintonización.

b. Memoria distribuida y robustez

La información contenida en una Red se encuentra distribuida en forma de los pesos w de las interconexiones entre los elementos de procesamiento. Debido a ello, el hecho de que una de estas unidades sufra una degradación en su desempeño, se dañe o simplemente deje de funcionar no repercute de manera total en la actuación de la Red. En un controlador convencional, como es el PID, en el que el procesamiento de información se encuentra centralizado en una sola unidad, no se cuenta con esta característica: cualquier tipo de falla o error afecta de manera total al comportamiento del controlador. Por ello, puede decirse que las Redes Neuronales permiten efectuar un control más robusto que los controladores convencionales.

c. Mínima oscilación de respuesta

La actuación de las Redes adecuadamente entrenadas permite alcanzar los valores deseados de puntos de ajuste con un mínimo de oscilaciones en la señal de respuesta. Esto representa una gran ventaja sobre los controladores PID, ya que en estos cualquier desviación con respecto a los parámetros de sintonización puede provocar importantes oscilaciones, que pueden no converger en el punto de ajuste.

d. Reconocimiento de patrones

La naturaleza del modelo matemático de las Redes Neuronales les permite encontrar diferentes relaciones entre los elementos de los vectores que participan en el entrenamiento. Es justamente este reconocimiento el que queda plasmado en el conjunto de pesos w de la Red. Los parámetros de los algoritmos de control convencionales deben de crearse a partir de la información del modelo matemático del proceso; durante el entrenamiento de las Redes, a través de mecanismos similares a la técnica de Back-Propagation, éstas encuentran los valores de sus propios parámetros.

Por otra parte, la desventaja principal de las Redes Neuronales con respecto al PID es, básicamente, el diseño y ejecución de las sesiones de entrenamiento. Para que el entrenamiento de una Red resulte eficiente, debe cumplir con dos puntos: contar con un grupo adecuado de modelos para el adiestramiento, y un tiempo adecuado para ello.

En algunas ocasiones⁹, este tiempo puede ser muy extenso. En el desarrollo experimental realizado en la presente tesis, este problema disminuyó considerablemente debido al empleo de un sistema cómputo 486 a 33 MHz, con coprocesador matemático; este sistema es lo suficientemente veloz como para que el tiempo de entrenamiento afectase en alguna forma el trabajo. En este punto es necesario mencionar que lo que se utilizó fue un software para simulación de Redes Neuronales. El aprendizaje de las Redes físicas generalmente presenta este problema.

Sin embargo, no se considera que el punto anterior le resta méritos a la utilidad de las Redes Neuronales como solución al *Problema Fundamental del Control*. En la actualidad, y como consecuencia del problema de los tiempos de aprendizaje, se han desarrollado diferentes técnicas encaminadas a disminuir la duración de las sesiones de entrenamiento, tales como los llamados *algoritmos genéticos*¹⁰. El diseño funcional de estos algoritmos permite reducir e incluso optimizar el tiempo de entrenamiento, por lo que se considera que este problema puede resolverse con relativa facilidad.

De esta manera, la conclusión principal de la presente tesis es que, ya que la actuación de las Redes Neuronales como controladores se equipara y, en algunos aspectos, es superior a la del controlador PID, los sistemas de control basados en Redes Neuronales constituyen una solución al *Problema Fundamental de Control*.

Puede verse entonces que la incorporación de tecnologías de punta dentro del esquema de los procesos de producción no solo es conveniente sino necesaria para elevar los índices de calidad, lo cual resulta imprescindible en el acontecer económico que hoy por hoy se da en los mercados internacionales. Por ello, la integración de Redes Neuronales, y otras técnicas de Inteligencia Artificial, como son los Sistemas Expertos, a los sistemas de control, conforman una buena alternativa para el desarrollo de sistemas tecnológicos que permita a la Industria reaccionar eficientemente ante la diversidad de cambios en los mercados.

(9) Lawton, G. "GENETIC ALGORITHMS FOR SCHEDULE OPTIMIZATION"
ALEXPERT, E.U.A., marzo 1990.

(10) Austin, S. "AN INTRODUCTION TO GENETIC ALGORITHMS"
ALEXPERT, E.U.A., marzo 1990

Las posibilidades de aplicación de este tipo de tecnologías en los procesos industriales son enormes. En México, los grupos académicos dedicados a las Matemáticas y a la Cibernética, poco a poco han empezado a ver con interés el creciente auge de las técnicas de Inteligencia Artificial en otros países. Sin embargo, muchos de los posibles desarrollos pueden darse en los procesos de producción, por lo que la IA debe impulsarse tanto en medios industriales como académicos.

De manera particular, la Facultad de Química de la UNAM, debido a la diversidad de las áreas de estudio que maneja, y por su misma estrecha relación con la Industria y los procesos productivos, resulta uno de los lugares ideales para el estudio y desarrollo de la Inteligencia Artificial en nuestro país, por lo que debe analizarse la posible inclusión de este tipo de técnicas en los planes de estudio, ya sea de licenciatura o posgrado. Así, los egresados de la Facultad contarán con una novedosa perspectiva que les permitirá contribuir de mejor manera al desarrollo de la Industria en México.

BIBLIOGRAFÍA

BIBLIOGRAFÍA

- Anderson, J.A.; Rosenfeld, E.
"NEUROCOMPUTING: Foundations of Research"
The MIT Press, Massachusetts Institute of Technology
E.U.A., 1988
- Austin, S.
"AN INTRODUCTION TO GENETIC ALGORITHMS"
AI-EXPERT
E.U.A., mar. 1990
- Fahlmann, S.E.
"AN EMPIRICAL STUDY OF LEARNING SPEED IN BACK-PROPAGATION NETWORKS"
CMU Technical report, CMU-CS-88-162
E.U.A. jun. 1988
- Guha, A.; Haggerty A.; Jelinek, J.
"NEURAL NETWORKS AS A CONTROL TECHNOLOGY"
Scientific Honeyweller
E.U.A., 1989
- Hecht-Nielsen, R.
"NEAREST MATCHED FILTER CLASSIFICATION OF SPACIO-TEMPORAL PATTERNS"
Hecht-Nielsen Neuro-Computer Corporation
E.U.A. 1986

-
- Houpls, C.H.; Lamont, G.
"DIGITAL CONTROL SYSTEMS"
McGraw-Hill Book Company
E.U.A. 1987
 - Jones, W.P.; Hoskins, J.
"BACK-PROPAGATION: A GENERALIZED DELTA LEARNING RULE"
BYTE-Magazine
E.U.A. oct. 1987
 - Klimasauskas, C.; Giver, J.; Pelton, G.
"NEURALWORKS EXPLORER MANUAL"
NeuralWare, Inc.
E.U.A., 1989
 - Lauton, G.
"GENETIC ALGORITHMS FOR SCHEDULE OPTIMIZATION"
AI-EXPERT
E.U.A., may. 1992
 - Luyben, W.L.
"PROCESS MODELING, SIMULATION AND CONTROL FOR CHEMICAL ENGINEERS"
McGraw-Hill Publishing Company
E.U.A. 1990
 - Miller, W.T.; Sutton, R.S.; Werbos, P.J.
"NEURAL NETWORKS FOR CONTROL"
The MIT Press, Massachusetts Institute of Technology
E.U.A., 1990

-
- Narendra, K.S.; Parthasarty, K.
"BACK-PROPAGATION IN DYNAMICAL SYSTEMS CONTAINING NEURAL NETWORKS"
Technical report 8905. Center for Systems Science. Department of Electric Engineering, Yale University
New Haven, CT. E.U.A., 1987

 - Ogata, K.
"INGENIERÍA DE CONTROL MODERNA"
Prentice-Hall Internacional
España, 1970

 - Pradeep, B.D.; Raymond, H.A.
"COMPUTER PROCESS CONTROL"
Instrument Society of America
E.U.A., 1988

 - Pretz, D.M.; García, C.
"FUNDAMENTAL PROCESS CONTROL"
Butterworths Series in Chemical Engineering
E.U.A., 1988

 - Przybylski, F.J.
"INDUSTRIAL CONTROL: A TUTORIAL"
Scientific Honeyweller
E.U.A., 1989

ANEXO I

FRAGMENTO DE HOJA DE "MACRO"

Hoja de macro utilizada en la elaboración de los archivos de entrenamiento para las redes neurales. El archivo consta de 1242 líneas, por lo que únicamente se presenta un fragmento.

NNDATA (a)

=ACTIVAR("NN-DATA.XLS")

1 =SELECCIONAR("R5C2:R9C2")
=COPIAR()
=SELECCIONAR("R5C5")
=PEGADO.ESPECIAL(3,1,FALSO,VERDADERO)
=SELECCIONAR("R5C3:R9C3")
=COPIAR()
=SELECCIONAR("R5C10")
=PEGADO.ESPECIAL(3,1,FALSO,VERDADERO)
2 =SELECCIONAR("R6C2:R10C2")
=COPIAR()
=SELECCIONAR("R6C5")
=PEGADO.ESPECIAL(3,1,FALSO,VERDADERO)
=SELECCIONAR("R6C3:R10C3")
=COPIAR()
=SELECCIONAR("R6C10")
=PEGADO.ESPECIAL(3,1,FALSO,VERDADERO)
3 =SELECCIONAR("R7C2:R11C2")
=COPIAR()
=SELECCIONAR("R7C5")
=PEGADO.ESPECIAL(3,1,FALSO,VERDADERO)
=SELECCIONAR("R7C3:R11C3")
=COPIAR()
=SELECCIONAR("R7C10")
=PEGADO.ESPECIAL(3,1,FALSO,VERDADERO)
4 =SELECCIONAR("R8C2:R12C2")
=COPIAR()
=SELECCIONAR("R8C5")
=PEGADO.ESPECIAL(3,1,FALSO,VERDADERO)
=SELECCIONAR("R8C3:R12C3")
=COPIAR()
=SELECCIONAR("R8C10")
=PEGADO.ESPECIAL(3,1,FALSO,VERDADERO)
5 =SELECCIONAR("R9C2:R13C2")
=COPIAR()
=SELECCIONAR("R9C5")
=PEGADO.ESPECIAL(3,1,FALSO,VERDADERO)
=SELECCIONAR("R9C3:R13C3")
=COPIAR()
=SELECCIONAR("R9C10")
=PEGADO.ESPECIAL(3,1,FALSO,VERDADERO)
6 =SELECCIONAR("R10C2:R14C2")
=COPIAR()
=SELECCIONAR("R10C5")
=PEGADO.ESPECIAL(3,1,FALSO,VERDADERO)
=SELECCIONAR("R10C3:R14C3")
=COPIAR()
=SELECCIONAR("R10C10")
=PEGADO.ESPECIAL(3,1,FALSO,VERDADERO)

ANEXO II

**ESTRUCTURA DE LA RED NEURAL 1
(AL TÉRMINO DEL ENTRENAMIENTO)**

(del programa NEURALWORKS EXPLORER)

Title: InstaNet (tm) Standard Back-Propagation Network version 1.00 20-Jun-88
Display Mode: Network Type: hetero-Associative

Display Style: default
Control Strategy: backprop

L/R Schedule: backprop

30002 Learn	O Recall	O Layer			
O Aux 1	O Aux 2	O Aux 3			
L/R Schedule: backprop					
Recall Step	1	0	0	0	0
Input Clamp	0.0000	0.0000	0.0000	0.0000	0.0000
Firing Density	100.0000	0.0000	0.0000	0.0000	0.0000
Temperature	0.0000	0.0000	0.0000	0.0000	0.0000
Gain	1.0000	0.0000	0.0000	0.0000	0.0000
Modifier	1.0000	0.0000	0.0000	0.0000	0.0000
Learn Step	5000	0	0	0	0
Coefficient 1	0.9000	0.0000	0.0000	0.0000	0.0000
Coefficient 2	0.6000	0.0000	0.0000	0.0000	0.0000
Coefficient 3	0.0000	0.0000	0.0000	0.0000	0.0000
Temperature	0.0000	0.0000	0.0000	0.0000	0.0000

IO Parameters

Learn Data: User
Recall Data: User
Result File: Output
UserIO Program: ulotus
I/P Ranges: 0.0000, 1.0000
O/P Ranges: 0.0000, 1.0000
I/P Start Col: 1
MinMax Table: default

O/P Start Col: 10
entries: 0

Layer: 1

PEs: 1	Sum: Sum		
Spacing: 5	F' offset: 0.00	Transfer: Linear	
Shape: Square	Output: Direct		
Scale: 1.00	Low Limit: 0.00	Error Func: standard	
Offset: 0.00	High Limit: 9999.00	Learn: --None--	
Init Low: -0.100	Init High: 0.100	L/R Schedule: (Network)	
Winner 1: None	Winner 2: None		
PE: Bias	1.000 Error Factor	1.000 Transfer	1.000 Output
0.000 Sum	0 Weights	9.149 Error	0.000 Current Error

Layer: In

PEs: 9	Sum: Sum				
Spacing: 5	F' offset: 0.00	Transfer: Linear			
Shape: Square	Output: Direct				
Scale: 1.00	Low Limit: -9999.00	Error Func: standard			
Offset: 0.00	High Limit: 9999.00	Learn: --None--			
Init Low: -0.100	Init High: 0.100	L/R Schedule: (Network)			
Winner 1: None	Winner 2: None				
PE: 2	1.000 Error Factor	0.841 Transfer	0.841 Output		
0.841 Sum	0 Weights	-0.000 Error	0.000 Current Error		
Input PE	Input Value	Weight	Type	Delta	Weight

PE: 3

1.000 Error Factor	0.921 Sum	0.921 Transfer	0.921 Output		
0.921 Sum	0 Weights	-0.000 Error	0.000 Current Error		
Input PE	Input Value	Weight	Type	Delta	Weight

PE: 4

1.000 Error Factor	0.974 Sum	0.974 Transfer	0.974 Output		
0.974 Sum	0 Weights	-0.000 Error	0.000 Current Error		
Input PE	Input Value	Weight	Type	Delta	Weight

PE: 5

1.000 Error Factor	0.999 Sum	0.999 Transfer	0.999 Output
0.999 Sum	0 Weights	-0.000 Error	0.000 Current Error

	Input PE	Input Value	Weight	Type	Delta	Weight
PE: 6						
1.000	Error Factor					
0.992	Sum	0.992	Transfer		0.992	Output
0	Weights	-0.000	Error		0.000	Current Error
	Input PE	Input Value	Weight	Type	Delta	Weight
PE: 7						
1.000	Error Factor					
0.500	Sum	0.500	Transfer		0.500	Output
0	Weights	-0.001	Error		0.000	Current Error
	Input PE	Input Value	Weight	Type	Delta	Weight
PE: 8						
1.000	Error Factor					
0.509	Sum	0.509	Transfer		0.509	Output
0	Weights	-0.001	Error		0.000	Current Error
	Input PE	Input Value	Weight	Type	Delta	Weight
PE: 9						
1.000	Error Factor					
0.521	Sum	0.521	Transfer		0.521	Output
0	Weights	-0.002	Error		0.000	Current Error
	Input PE	Input Value	Weight	Type	Delta	Weight
PE: 10						
1.000	Error Factor					
0.534	Sum	0.534	Transfer		0.534	Output
0	Weights	-0.002	Error		0.000	Current Error
	Input PE	Input Value	Weight	Type	Delta	Weight
Layer: Hidden 1						
PEs: 8						
Spacing:	5	F' offset:	0.00		Sum:	Sum
Shape:	Square				Transfer:	Sigmoid
Scale:	1.00	Low Limit:	-9999.00		Output:	Direct
Offset:	0.00	High Limit:	9999.00		Error Func:	standard
Init Low:	-0.100	Init High:	0.100		Learn:	Delta-Rule
Winner 1:	None				L/R Schedule:	{Network}
					Winner 2:	None
PE: 11						
1.000	Error Factor					
-1.069	Sum	0.256	Transfer		0.256	Output
10	Weights	-0.000	Error		-0.002	Current Error
	Input PE	Input Value	Weight	Type	Delta	Weight
	Bias	+1.0000	-1.1242	V-a	-0.0005	
	2	+0.8408	-0.2341	V-a	-0.0004	
	3	+0.9207	-0.1801	V-a	-0.0003	
	4	+0.9745	-0.1140	V-a	-0.0004	
	5	+0.9987	-0.1289	V-a	-0.0004	
	6	+0.9920	-0.0769	V-a	-0.0004	
	7	+0.5005	-0.0227	V-a	-0.0003	
	8	+0.5093	+0.2459	V-a	-0.0003	
	9	+0.5206	+0.4539	V-a	-0.0003	
	10	+0.5336	+0.7148	V-a	-0.0003	
PE: 12						
1.000	Error Factor					
-1.593	Sum	0.169	Transfer		0.169	Output
10	Weights	-0.000	Error		-0.003	Current Error
	Input PE	Input Value	Weight	Type	Delta	Weight
	Bias	+1.0000	-2.5316	V-a	-0.0005	
	2	+0.8408	-0.0833	V-a	-0.0004	
	3	+0.9207	-0.0378	V-a	-0.0003	
	4	+0.9745	+0.0166	V-a	-0.0004	
	5	+0.9987	-0.0124	V-a	-0.0004	
	6	+0.9920	+0.0678	V-a	-0.0004	
	7	+0.5005	-0.0361	V-a	-0.0002	
	8	+0.5093	+0.3619	V-a	-0.0002	
	9	+0.5206	+0.6158	V-a	-0.0002	
	10	+0.5336	+0.9045	V-a	-0.0002	
PE: 13						
1.000	Error Factor					

-2.306	Sum	0.091	Transfer	0.091	Output
10	Weights	-0.000	Error	-0.005	Current Error
	Input PE	Input Value	Weight Type	Delta Weight	
	Bias	+1.0000	-5.6647 V-a	-0.0004	
	2	+0.8408	+0.2609 V-a	-0.0002	
	3	+0.9207	+0.2023 V-a	-0.0002	
	4	+0.9745	+0.1424 V-a	-0.0003	
	5	+0.9987	+0.1545 V-a	-0.0003	
	6	+0.9920	+0.2569 V-a	-0.0003	
	7	+0.5005	+0.8312 V-a	-0.0001	
	8	+0.5093	+1.0339 V-a	-0.0001	
	9	+0.5206	+1.2787 V-a	-0.0001	
	10	+0.5336	+1.4893 V-a	-0.0001	

PE: 14

1.000	Error Factor	0.072	Transfer	0.072	Output
-2.563	Sum	0.000	Error	0.001	Current Error
10	Weights	0.072	Transfer	0.001	Current Error
	Input PE	Input Value	Weight Type	Delta Weight	
	Bias	+1.0000	-0.2143 V-a	+0.0001	
	2	+0.8408	+0.0550 V-a	+0.0001	
	3	+0.9207	-0.1393 V-a	+0.0000	
	4	+0.9745	-0.3277 V-a	+0.0001	
	5	+0.9987	-0.4437 V-a	+0.0001	
	6	+0.9920	-0.0685 V-a	+0.0001	
	7	+0.5005	-0.4887 V-a	+0.0001	
	8	+0.5093	-0.6554 V-a	+0.0001	
	9	+0.5206	-0.7848 V-a	+0.0001	
	10	+0.5336	-0.8416 V-a	+0.0001	

PE: 15

1.000	Error Factor	0.102	Transfer	0.102	Output
-2.177	Sum	0.000	Error	0.001	Current Error
10	Weights	0.102	Transfer	0.001	Current Error
	Input PE	Input Value	Weight Type	Delta Weight	
	Bias	+1.0000	-0.3570 V-a	+0.0001	
	2	+0.8408	-0.1822 V-a	+0.0001	
	3	+0.9207	-0.0584 V-a	+0.0001	
	4	+0.9745	-0.2445 V-a	+0.0001	
	5	+0.9987	-0.2446 V-a	+0.0001	
	6	+0.9920	-0.0094 V-a	+0.0001	
	7	+0.5005	-0.3803 V-a	+0.0001	
	8	+0.5093	-0.4781 V-a	+0.0001	
	9	+0.5206	-0.5611 V-a	+0.0001	
	10	+0.5336	-0.7395 V-a	+0.0001	

PE: 16

1.000	Error Factor	0.105	Transfer	0.105	Output
-2.146	Sum	0.000	Error	0.002	Current Error
10	Weights	0.105	Transfer	0.002	Current Error
	Input PE	Input Value	Weight Type	Delta Weight	
	Bias	+1.0000	-0.1423 V-a	+0.0001	
	2	+0.8408	-0.3244 V-a	+0.0002	
	3	+0.9207	-0.2295 V-a	+0.0001	
	4	+0.9745	+0.0984 V-a	+0.0001	
	5	+0.9987	+0.1046 V-a	+0.0001	
	6	+0.9920	-0.1431 V-a	+0.0001	
	7	+0.5005	-0.3737 V-a	+0.0001	
	8	+0.5093	-0.7207 V-a	+0.0001	
	9	+0.5206	-0.8364 V-a	+0.0001	
	10	+0.5336	-1.1023 V-a	+0.0001	

PE: 17

1.000	Error Factor	0.031	Transfer	0.031	Output
3.452	Sum	0.000	Error	0.002	Current Error
10	Weights	0.031	Transfer	0.002	Current Error
	Input PE	Input Value	Weight Type	Delta Weight	
	Bias	+1.0000	-0.1892 V-a	-0.0001	
	2	+0.8408	-0.5857 V-a	+0.0001	
	3	+0.9207	-0.1968 V-a	-0.0000	

4	+0.9745	-0.1317	V-a	+0.0000
5	+0.9987	-0.0226	V-a	+0.0000
6	+0.9920	-0.0935	V-a	+0.0000
7	+0.5005	-0.8438	V-a	+0.0000
8	+0.5093	-1.0983	V-a	+0.0000
9	+0.5206	-1.1261	V-a	+0.0000
10	+0.5336	-1.4699	V-a	+0.0000

PE: 18

1.000 Error Factor

2.065 Sum 0.887 Transfer

0.887 Output
0.003 Current Error

10 Weights

Input PE	Input Value	Weight	Type	Delta Weight
Bias	+1.0000	+3.7255	V-a	+0.0003
2	+0.8408	+0.0856	V-a	+0.0002
3	+0.9207	+0.0347	V-a	+0.0002
4	+0.9745	-0.0043	V-a	+0.0003
5	+0.9987	-0.0369	V-a	+0.0003
6	+0.9920	+0.0446	V-a	+0.0003
7	+0.5005	-0.5344	V-a	+0.0001
8	+0.5093	-0.7665	V-a	+0.0001
9	+0.5206	-0.9679	V-a	+0.0001
10	+0.5336	-1.1324	V-a	+0.0001

Layer: Out

PEs: 1
Spacing: 5 F' offset: 0.00
Shape: Square
Scale: 1.00 Low Limit: -9999.00
Offset: 0.00 High Limit: 9999.00
Init Low: -0.100 Init High: 0.100
Winner 1: None

Sum: Sum
Transfer: Sigmoid
Output: Direct
Error Func: standard
Learn: Delta-Rule
L/R Schedule: (Network)
Winner 2: None

PE: 19

1.000 Error Factor

0.211 Sum 0.552 Transfer

0.552 Output
-0.005 Current Error

9 Weights

Input PE	Input Value	Weight	Type	Delta Weight
Bias	+1.0000	+1.5298	V-a	-0.0013
11	+0.2557	+1.7757	V-a	-0.0004
12	+0.1690	+2.4281	V-a	-0.0002
13	+0.0906	+4.2359	V-a	-0.0000
14	+0.0715	-1.1794	V-a	-0.0000
15	+0.1018	-0.8883	V-a	-0.0001
16	+0.1047	-1.3341	V-a	-0.0000
17	+0.0305	-1.7900	V-a	+0.0001
18	+0.8874	-2.4797	V-a	-0.0011