

**DISEÑO E IMPLEMENTACIÓN DE UN CONTROLADOR BASADO EN REDES  
NEURONALES CON ENTRENAMIENTO RÁPIDO PARA SISTEMAS DE  
CONTROL 2 x 2**

**ERICK FRANK SARMIENTO JURADO**

**UNIVERSIDAD DEL NORTE  
DIVISIÓN DE INGENIERÍAS  
MAESTRÍA EN INGENIERÍA MECÁNICA  
BARRANQUILLA**

**2006**

**DISEÑO E IMPLEMENTACIÓN DE UN CONTROLADOR BASADO EN REDES  
NEURONALES CON ENTRENAMIENTO RÁPIDO PARA SISTEMAS  
DE CONTROL 2 x 2**

**ERICK FRANK SARMIENTO JURADO**

**Director:**

**Ing. MARCO E. SANJUAN, Ph.D.**

**Correctores:**

**Ing. CARLOS PATERNITA, Ph.D.**

**Ing. ARIF ESLAIT, Ms. Sc.**

**Trabajo presentado para optar por el título de Magíster en Ingeniería Mecánica**

**UNIVERSIDAD DEL NORTE  
DIVISIÓN DE INGENIERÍAS  
MAESTRÍA EN INGENIERÍA MECÁNICA  
BARRANQUILLA**

**2006**

Nota de aceptación:

---

Marco Sanjuán  
Director

---

Carlos Paternina  
Miembro del Comité Evaluador

---

Arif Eslait  
Miembro del Comité Evaluador

Barranquilla 15-07-2006

## **DEDICATORIA**

A mi familia, profesores y compañeros de maestría, por su invaluable apoyo en la consecución del presente trabajo.

## TABLA DE CONTENIDO

1. INTRODUCCIÓN	1
1.1 GENERALIDADES	1
1.2 OBJETIVOS	2
1.3 JUSTIFICACIÓN	3
1.4 DESCRIPCIÓN DE LA INVESTIGACIÓN	4
1.4.1 Antecedentes	4
1.4.2 Planteamiento del Problema	5
1.4.3 Solución Propuesta	5
2. MARCO REFERENCIAL Y REVISIÓN DEL ESTADO DEL ARTE	7
2.1 CONTROL DE PROCESOS MULTIVARIABLES BASADO EN PID	7
2.1.1 Interacción en Sistemas Multivariantes	9
2.1.2 Interacción Positiva y Negativa	12
2.1.3 Apareamiento de Variables Controladas y Manipuladas	13
2.1.4 Sintonización de Controladores PID para Sistemas 2 x 2	13
2.1.5 Ecuaciones de Sintonización	14
2.2 DISEÑO DE DESACOPLADORES PARA SISTEMAS 2 x 2	16
2.3 REDES NEURONALES ARTIFICIALES	19
2.3.1 Introducción y Generalidades	19
2.3.2 Inspiración en la Neurociencia	20
2.3.3 Conceptos Básicos	21

2.3.3.1 Arquitectura de la Red	22
2.3.4 Métodos de Aprendizaje	27
2.3.4.1 Aprendizaje Supervisado	28
2.3.4.2 Aprendizaje no Supervisado	30
2.3.4.3 Aprendizaje por Reforzamiento	31
2.3.5 Reglas de Aprendizaje	33
2.3.5.1 Regla de Hebb's	34
2.3.5.2 La Regla Delta	34
2.3.5.3 Backpropagation	35
2.3.5.4 Redes de Hopfield	36
2.3.5.5 Activación Aleatoria de los Pesos de la Red (Rawn)	37
2.3.6 Validación	38
3 DISEÑO DEL CONTROLADOR NEURONAL	39
3.1 CONTROL PREDICTIVO	40
3.1.1 Control Predictivo Basado en Modelos	40
3.1.2 Estrategia del Controlador	42
3.1.3 Función Objetivo	44
3.1.4 Obtención de la Ley de Control	47
3.2 FASES DEL DISEÑO DEL CONTROLADOR NEURONAL	47
3.2.1 Fase de Recolección de Datos	48
3.2.2 Fase de Entrenamiento	49
3.2.3 Fase para Evaluar la Calidad de la Predicción	50
3.2.4 Fase de Control	51
3.3 ALGORITMO DEL CONTROLADOR NEURONAL	52

3.3.1 Simulación del Proceso en Lazo Abierto	52
3.3.2 Entrenamiento de la Red Neuronal	54
3.3.3 Algoritmo de Predicción	57
3.3.4 Función Objetivo	57
3.3.5 Algoritmo de Control	58
4 IMPLEMENTACIÓN DEL CONTROLADOR NEURONAL Y EVALUACIÓN DEL RENDIMIENTO EN SIMULACIÓN	61
4.1 TANQUE DE MEZCLADO	62
4.1.1 Descripción del Proceso	62
4.1.2 Entradas y Salidas en Lazo Abierto	63
4.1.3 Definición de Variables Controladas y Manipuladas	63
4.1.4 Determinación de las Funciones de Transferencia	64
4.1.5 Apareamiento de Variables	66
4.1.6 Sintonización de los Controladores PID	67
4.1.7 Implementación de los Desacopladores	68
4.1.8 Entrenamiento de la Red y Ajuste de los Parámetros del Controlador Neuronal	68
4.1.8.1 Determinación del Factor de Agresividad	70
4.1.8.2 Determinación del Tiempo de Muestreo	71
4.1.8.3 Determinación del Número de Predicciones	73
4.2 PRUEBAS Y COMPARACIÓN DEL DESEMPEÑO SOBRE EL TANQUE DE MEZCLADO	74
4.2.1 Prueba 1. Cambio en el Set-point de Composición	76
4.2.1.1 Comparaciones de la Salida del Proceso	76

4.2.1.2 Comparaciones de la Salida de los Controladores	79
4.2.2 Prueba 2. Cambio en el Set-point de Flujo	82
4.2.2.1 Comparaciones de la Salida del Proceso	82
4.2.2.2 Comparaciones de la Salida de los Controladores	84
4.3 TANQUE DE CALENTAMIENTO	87
4.3.1 Descripción Del Proceso	87
4.3.2 Entradas y Salidas en Lazo Abierto	88
4.3.3 Definición de Variables Controladas y Manipuladas	88
4.3.4 Funciones de Transferencia	88
4.3.5 Apareamiento de Variables	90
4.3.6 Sintonización de los Controladores PID	91
4.3.7 Implementación de los Desacopladores	91
4.3.8 Entrenamiento de la Red Implementación y Ajuste de los Parámetros del Controlador Neuronal.	92
4.3.8.1 Determinación del Factor de Agresividad	93
4.3.8.2 Determinación del Tiempo de Muestreo	94
4.3.8.3 Determinación del Número de Predicciones	95
4.4 PRUEBAS Y COMPARACIÓN DEL DESEMPEÑO SOBRE EL TANQUE DE CALENTAMIENTO	97
4.4.1 Prueba 1. Cambio en el Set-point de Temperatura	98
4.4.1.1 Comparación de la Salida del Proceso	98
4.4.1.2 Comparación de la Salida de los Controladores	100
4.4.2 Prueba 2. Cambio en el Set-point de Flujo	102
4.4.2.1 Comparación de la Salida del Proceso	102



4.4.2.2 Comparación de la Salida de los Controladores	104
4.4.3 Prueba 3. Cambios Sucesivos de Set-points más Perturbación	107
4.4.4 Prueba 4. Cambios Simultáneos de los Puntos de Control, más Perturbación y Presencia de ruido	108
5 CONCLUSIONES	112
6 APORTES Y RECOMENDACIONES	115
6.1 APORTES	115
6.2 RECOMENDACIONES	116
REFERENCIAS	118
ANEXO A. Códigos de Fuente del Algoritmo de Control	122
ANEXO B. Modelamiento del Proceso del Tanque de Mezclado	129
ANEXO C. Modelamiento del Proceso del Tanque de Calentamiento	132

# 1. INTRODUCCIÓN

## 1.1 GENERALIDADES

El presente trabajo de tesis tiene como marco conceptual e investigativo dos de los avances científicos recientes más importantes del conocimiento:

Los Sistemas de Control Automáticos

Las Redes Neuronales Artificiales

En esta investigación se modelan dos procesos, en cada uno de los cuales se pretenden controlar automáticamente más de una variable de salida. Para lo anterior se aplica una estrategia de control predictivo usando redes neuronales artificiales.

También se comparan los resultados obtenidos por el controlador neuronal con el referente más común en el área de control industrial, el controlador PID, consiguiendo de esta manera poder comprobar su rendimiento sobre procesos no lineales.

## **1.2 OBJETIVOS**

### **OBJETIVO GENERAL**

Diseñar e implementar un controlador basado en redes neuronales con entrenamiento rápido para sistemas de control 2 x 2.

### **OBJETIVOS ESPECÍFICOS**

- Diseñar un algoritmo capaz de realizar la labor de control, a través de la implementación de una estrategia predictiva que usa un modelo neuronal del proceso, para obtener resultados superiores a los tradicionales controladores PID.
- Integrar de manera exitosa el controlador a dos procesos multivariantes no lineales, en simulación, para obtener las respuestas de control deseadas.
- Implementar un sistema de control para procesos multivariantes 2 x 2, usando la técnica del PID desacoplado, para obtener una referencia comparativa respecto al controlador neuronal.
- Comparar las acciones de control obtenidas usando controladores PID desacoplados, no desacoplados y el controlador neuronal, por medio de

parámetros cuantitativos y cualitativos, que permitan evaluar el desempeño de los tres tipos de control utilizados.

### **1.3 JUSTIFICACIÓN**

En recientes años, ha habido un creciente incremento en la aplicación de una herramienta novedosa denominada Red Neuronal Artificial (RNA) en identificación de la dinámica (modelado), predicción y control de sistemas complejos. Las redes neuronales se caracterizan por su habilidad de aprender de los ejemplos en lugar de tener que programarse en un sentido convencional. Su uso posibilita que la dinámica de sistemas complejos sea modelada y un control exacto sea logrado a través del entrenamiento, sin tener información a priori sobre los parámetros del sistema

Por otro lado es común encontrar en la industria dificultades asociadas al control de sistemas dinámicos complejos, que están usualmente relacionadas a la existencia de fenómenos no lineales y de perturbaciones impredecibles o parcialmente conocidas, las cuales afectan el desempeño de los controladores básicos. Para dar solución a la problemática antes expuesta, es importante incorporar estrategias de control que permitan manejar situaciones inciertas para, así, lograr satisfactoriamente las especificaciones requeridas.

Con el objetivo de diseñar un modelo general y satisfactorio de Control Inteligente, además de cumplir con parámetros de modernidad y flexibilidad exigidos en la actualidad, se implementará un controlador basado en redes neuronales.

## **1.4 DESCRIPCIÓN DE LA INVESTIGACIÓN**

En esta sección se hace la presentación del problema que motivó la realización de esta investigación, se comentan algunos antecedentes y se termina con la solución propuesta.

### **1.4.1 Antecedentes**

Algunas aplicaciones industriales han sido automatizadas usando controladores con redes neuronales, usando el método de entrenamiento Backpropagation. Sin embargo, un reciente método parece ser la nueva alternativa a los posibles problemas de Backpropagation, (lentitud en el proceso de aprendizaje y demasiados cálculos), ya que solo necesita una iteración para llevar a cabo el entrenamiento, este método es llamado RAWN (Random Activation Weight Neural Network).

También se han desarrollado y presentado sistemas de control neuronal que han sido aplicados con eficacia sobre procesos en los cuales se pretende mantener a una sola variable en un punto o valor deseado.

Iniciamos, entonces, nuestra investigación tomando como punto de partida un estudio previo desarrollado por los Ingenieros Juan Saucedo Mercado y Efrén Ospino Martínez sobre el diseño y la implementación de un controlador con redes neuronales de entrenamiento rápido (RAWN) basado en identificación del proceso como un sistema de primer orden más tiempo muerto aplicado en sistemas univariados.

#### **1.4.2 Planteamiento Del Problema**

La relevancia del uso de las redes neuronales en control se debe principalmente a que es frecuente toparse con la necesidad de diseñar estrategias de control automático para procesos no lineales y multivariados, pero son pocos los diseños de control que para este tipo de sistemas se han desarrollado. A pesar de los muy buenos resultados que se han reportado en las investigaciones que se han efectuado y la gran cantidad de esquemas de control que se han probado, las aplicaciones del neurocontrol multivariable no son aún una rutina y la mayoría de los esquemas diseñados se aplican sólo para procesos SISO (Simple Input Simple Output).

Ante la necesidad de tener controladores basados en redes neuronales artificiales capaces de poder manejar satisfactoriamente procesos del tipo MIMO (Multiple Inputs Multiple Outputs), con un desempeño igual o mejor que los controladores PID, se hace necesaria la búsqueda de una solución óptima al problema antes mencionado.

### **1.4.3 Solución Propuesta**

La solución propuesta para este problema, es diseñar un controlador neuronal, entrenado con el método RAWN, que tenga múltiples salidas y que sea capaz de mantener a más de una sola variable del proceso en su valor deseado y que además genere resultados superiores a los obtenidos si se usaran los tradicionales controladores del tipo PID.

## 2. MARCO REFERENCIAL Y REVISIÓN DEL ESTADO DEL ARTE

### 2.1 CONTROL DE PROCESOS MULTIVARIABLES BASADO EN PID

En muchos procesos industriales el objetivo consiste en mantener más de una variable en su punto de control (set-point) o valor deseado. En este tipo de casos se dice que el problema de control es multivariable (lo cual implica que el número de variables a controlar puede ser 2 o mayor de 2). Un ejemplo de control de sistemas multivariables se presenta en el reactor tanque agitado mostrado en la figura 2.1.

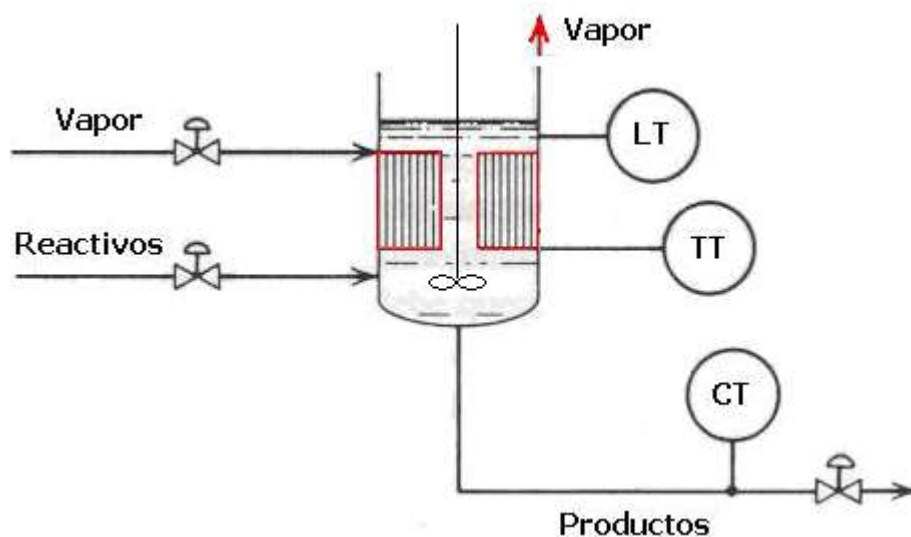


Figura 2.1 Reactor Tanque Agitado.



En este proceso se desea controlar el nivel, la temperatura y la composición del flujo de salida del reactor. Como potenciales variables manipuladas podemos emplear el flujo de la corriente de producto, la carga térmica del medio de calentamiento del reactor y el flujo de los reactivos:

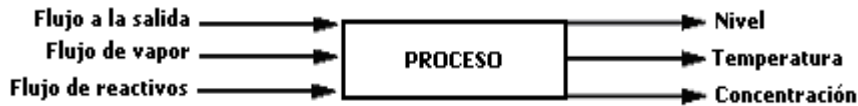


Figura 2.2 Proceso con múltiples variables manipuladas y controladas

En este caso, es más o menos simple tomar la decisión de qué variable manipulada se utiliza para compensar a qué variable controlada y se proponga el siguiente apareamiento de lazos de control:

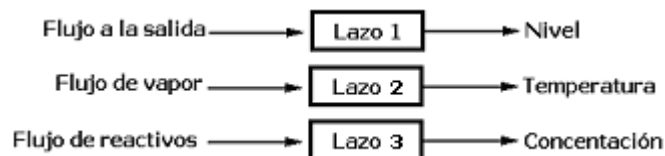


Figura 2.3 Apareamiento de lazos de control

Esta forma de resolver el problema de control de sistemas multivariables se reduce a diseñar cada lazo de control de manera independiente del resto de los lazos (es decir, sin tomar en cuenta la presencia del resto de los lazos de control). Cada lazo se diseña usando los métodos para el control de sistemas univariados. Sin embargo, la presencia del efecto de interacción hará que el desempeño de un lazo afecte al de los otros y viceversa. En otras palabras, la interacción complica la sintonización de los lazos de manera independiente.

Existe otro esquema para resolver el problema del control de sistemas multivariables y se basa en el uso de un solo controlador que manipula simultáneamente todas las variables manipuladas cuando algún o algunas variables controladas se alejan de su set-point. De esta forma se consigue compensar por el efecto de interacción. El grado de compensación depende del diseño del controlador.

En sistemas multivariables la interacción entre variables es el efecto más importante en el diseño y sintonización de lazos de control. Un proceso presenta interacción cuando una variable de entrada afecta a más de una variable de salida; afecta quiere decir que la variable controlada se aleja de su valor de estado estacionario en virtud de cambios en la variable de entrada.

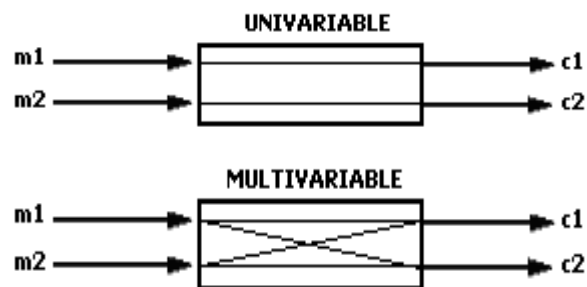


Figura 2.4 Interacción en sistemas multivariables. (Tomado de [1])

### 2.1.1 Interacción en Sistemas Multivariables

Tiene sentido que cada variable controlada se controle mediante la variable manipulada que ejerce mayor influencia sobre aquella. En este contexto, la influencia y la ganancia tienen el mismo significado y, en consecuencia, para

tomar una decisión se deben encontrar todas las ganancias del proceso. Específicamente, las ganancias de estado estacionario de lazo abierto de interés para sistemas 2 x 2 son las siguientes:

$$K_{11} = \left. \frac{\Delta c_1}{\Delta m_1} \right|_{m_2} \quad K_{12} = \left. \frac{\Delta c_1}{\Delta m_2} \right|_{m_1}$$

(Ecuaciones 2.1) [2]

$$K_{21} = \left. \frac{\Delta c_2}{\Delta m_1} \right|_{m_2} \quad K_{22} = \left. \frac{\Delta c_2}{\Delta m_2} \right|_{m_1}$$

donde  $K_{ij}$  es la ganancia que relaciona la variable controlada  $c_i$  con la variable manipulada  $m_j$ .

Las cuatro ganancias se pueden ordenar en forma de matriz, a fin de tener una descripción gráfica de su relación con las variables controlada y manipulada. La matriz se denomina matriz de ganancia en estado estacionario. Debido a que las ganancias  $K_{ij}$  pueden tener diferentes unidades se debe normalizar esta matriz a fin de poder realizar la elección. Con este último propósito la ganancia relativa  $\mu_{ij}$  entre la salida  $c_i$  y la entrada  $m_j$  (o sea, el cociente entre dos ganancias en estado estacionario) se puede escribir

$$\mu_{ij} = \frac{\left. \frac{\partial c_i}{\partial m_j} \right|_m}{\left. \frac{\partial c_i}{\partial m_j} \right|_c} = \text{todos los lazos abiertos} / \text{todos los lazos abiertos, excepto } i \neq j$$

(Ecuación 2.2) [2]

Como se aprecia en esta definición, el término de ganancia relativa es adimensional, la cual se puede utilizar para decidir la agrupación por pares de las variables controlada y manipulada.

Una vez se evalúan los términos de ganancia relativa, es posible formar la matriz de ganancia relativa que se muestra en la figura 2.5.

	<b>m<sub>1</sub></b>	<b>m<sub>2</sub></b>
<b>c<sub>1</sub></b>	u <sub>11</sub>	u <sub>12</sub>
<b>c<sub>2</sub></b>	u <sub>21</sub>	u <sub>22</sub>

Figura 2.5 Matriz de Ganancia Relativa

Una propiedad de esta matriz es que la suma de todos los términos en cada columna debe ser igual a uno, lo cual quiere decir que para un sistema 2 x 2 únicamente se requiere evaluar un término y los otros pueden obtenerse mediante esta propiedad. Además el valor numérico de  $\mu_{ij}$  es una medida de interacción entre los circuitos.

Si  $\mu_{ij} = 1$ , esto significa que cuando se abren los demás lazos, la ganancia del proceso es la misma que cuando se cierran. Esto significa que entre el lazo de interés y los demás lazos no hay interacciones.

Si  $\mu_{ij} = 0$ , esto puede deberse a dos posibilidades: primera, la ganancia de lazo abierto puede ser cero, en este caso  $m_j$  no afecta a  $c_i$ . Segunda, la ganancia de lazo cerrado es tan grande que  $\mu_{ij} \approx 0$ , lo que representa que los demás interactúan en gran medida con los circuitos de interés para mantener constantes las otras variables. Cualquiera de las dos posibilidades indica que  $c_i$  no debe ser controlada con  $m_j$ .

Si  $\mu_{ij} = \infty$ , la ganancia de lazo cerrado es cero o muy pequeña, lo cual significa que cuando los demás lazos se sitúan en automático, el lazo de interés no se puede controlar, porque  $m_j$  afecta a  $c_i$  muy poco o nada en absoluto.

### 2.1.2 Interacción Positiva y Negativa

La interacción positiva es la interacción que existe cuando todos los términos de ganancia relativa son positivos. Para observar bajo qué condiciones se da este tipo de interacción analicemos la expresión  $\mu_{11}$  para el sistema 2 x 2:

$$\mu_{11} = \frac{K_{11}K_{22}}{K_{11}K_{22} - K_{12}K_{21}} = \frac{1}{1 - \frac{K_{12}K_{21}}{K_{11}K_{22}}} \quad (\text{Ecuación 2.3}) [2]$$

Si la cantidad de  $K$  positivas es impar, entonces el valor de  $\mu_{11}$  será positivo y, además, su valor numérico estará entre 0 y 1. En sistemas con interacción positiva existe ayuda de los lazos de control entre sí.

Cuando existe una cantidad par de valores positivos de  $K$  o una cantidad igual de valores positivos y negativos de  $K$ , el valor de  $\mu_{11}$  es mayor de 1 o menor que 0; en cualquier caso habrá algunas  $\mu_{ij}$  con valores negativos en la misma fila y columna. En este caso la interacción es negativa y los lazos de control se combaten entre sí.

### **2.1.3 Apareamiento de Variables Controladas y Manipuladas.**

En general los valores de ganancias relativas cercanos a la unidad representan combinaciones de variables controladas y manipuladas que se pueden controlar; los valores de ganancias relativas que tienden a cero representan combinaciones que no se pueden controlar. Con base en esta premisa se puede presentar la siguiente regla: aparear variables de entrada/salida que tengan elementos positivos de la Matriz de Ganancia Relativa tan cercanos a 1 como sea posible.

### **2.1.4 Sintonización de Controladores PID para sistemas 2 x 2**

El procedimiento para diseñar sistemas de control de lazos múltiples involucra los siguientes pasos:

- a) Selección del esquema de apareamiento.
- b) Sintonización de los controladores de los lazos individuales usando las ecuaciones adaptadas.

En sistemas que presentan poca interacción (es decir, sistemas para los cuales los términos de la Matriz de Ganancia Relativa están muy cercanos a 1) los métodos de sintonización de controladores tradicionales (por ejemplo Ziegler-Nichols), bastaran para tener un buen desempeño del esquema de control de lazos múltiples. Sin embargo, dichas reglas de sintonización podrían resultar inapropiadas para sistemas que presenten valores grandes o negativos de la Matriz de Ganancia Relativa. El control podría ser prácticamente imposible de obtener en presencia de interacciones fuertes.

Bajo condiciones de fuerte interacción se debe sintonizar cada lazo de control de manera independiente y luego reajustar los parámetros de los controladores hasta obtener un buen desempeño de los lazos de control.

### 2.1.5 Ecuaciones de Sintonización

En [3] se define que sintonizar un controlador PID significa conseguir los valores apropiados de los parámetros de Ganancia ( $K_C$ ), Tiempo Integral ( $\tau_I$ ) y Tiempo Derivativo ( $\tau_D$ ) que permitan obtener una respuesta específica de circuito cerrado.

Las ecuaciones de sintonización adaptadas para sistemas multivariables 2 x 2 son las siguientes:

Lazo 1	Lazo 2
$K_{C1} = \frac{1}{1.2K_{P11}} \left( \frac{\tau_{11}}{t_{011}} \right)$	$K_{C2} = \frac{\tau_{22}}{K_{P22}} \frac{1}{(1-\alpha)} \frac{1}{(\Gamma + t_{022})}$

$$\tau_{I1} = \tau_{11}$$

$$\tau_{I2} = \tau_{22}$$

$$\tau_{D1} = \frac{t_{011}}{2}$$

$$\tau_{D2} = \frac{T_d}{2}$$

(Ecuaciones 2.5) Tomadas de [4]

Donde el lazo 1 es aquel con la relación  $\frac{t_0}{\tau}$  más pequeña.

Los parámetros adicionales se definen de la siguiente manera:

$$\alpha = \frac{K_{P12} K_{P21}}{K_{P11} K_{P22}} \quad \text{(Ecuación 2.6) [4]}$$

$$T_d = t_{o12} + t_{o21} - t_{o22} \quad \text{(Ecuación 2.7) [4]}$$

La constante de tiempo de lazo cerrado  $\Gamma$  se obtiene a partir de las siguientes expresiones:

$$\ln \frac{\Gamma}{t_{o22}} = 1.389\alpha + 1.487\alpha^2 + 0.170\gamma - 0.111\theta - 0.110\delta \quad \text{(Ecuación 2.8) [4]}$$

Donde:

$$\gamma = \frac{t_{o11}}{t_{o22}} \quad \theta = \frac{t_{o12}}{t_{o21}} \quad \delta = \frac{\tau_{12}\tau_{11}}{\tau_{11}\tau_{22}} \quad \text{(Ecuación 2.9) [4]}$$



## 2.2 DISEÑO DE DESACOPLADORES PARA SISTEMAS 2 x 2

Con el objeto de reducir o eliminar el efecto de interacción en un sistema multivariable se puede implementar un sistema de control en el que se eliminen los lazos que interactúan. A tal proceso se denomina desacoplamiento. En la figura 2.5 se muestra el sistema general de control 2 x 2 con interacción, o sistema TITO (Twice Input Twice Output [5] y [6]).

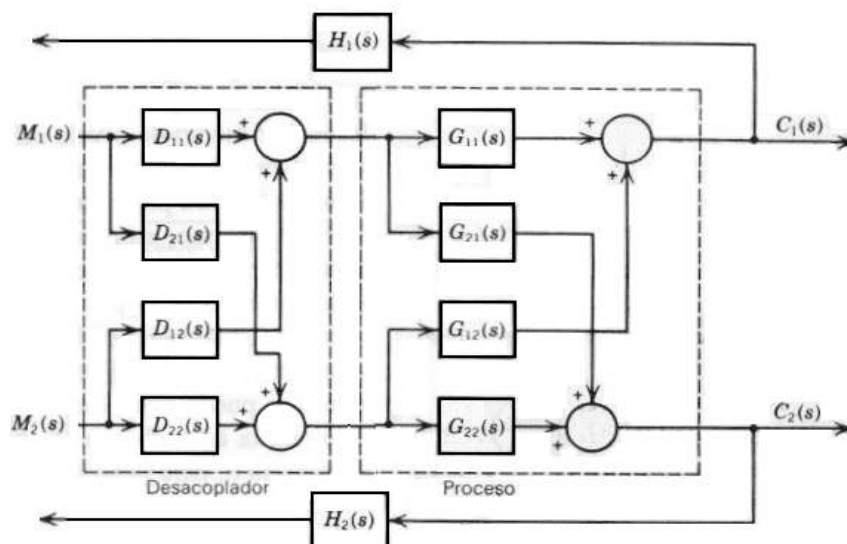


Figura 2.5 Sistema general de control 2 x 2 con interacción. (Tomado de [2])

En la figura 2.5 se usa la siguiente notación:

$C_i(s)$  = Transformada de Laplace de la variable controlada  $i$

$M_i(s)$  = Transformada de Laplace de la variable manipulada  $i$

$H_i(s)$  = Función de transferencia del sensor-transmisor  $i$

$G_{ij}(s)$  = Función de transferencia que describe la interacción de la variable manipulada  $j$  sobre la variable controlada  $i$ , integrando válvula y proceso

$D_{ij}(s)$  = Función de transferencia del desacoplador que describe la interacción de la variable manipulada  $j$  sobre la variable controlada  $i$

El desacoplador se debe diseñar de tal manera que se obtengan dos lazos de control que parezcan independientes, es decir, de tal forma que con un cambio en la salida del controlador 1,  $M_1(s)$ , se produzca un cambio en  $C_1(s)$ , pero no en  $C_2(s)$ ; de igual forma, con un cambio en la salida del controlador 2,  $M_2(s)$ , se debe producir un cambio en  $C_2(s)$ , pero no en  $C_1(s)$ .

Usando la figura 2.6 y por medio de álgebra de bloques, para un sistema 2 x 2 la salida del controlador 2,  $M_2(s)$ , afecta a la variable controlada 1,  $C_1(s)$ , de la forma en que se expresa la siguiente ecuación:

$$C_1(s) = \Phi_{12}(s)G_{11}(s) + D_{22}(s)G_{12}(s)\bar{M}_2(s) \quad \text{(Ecuación 2.10)}$$

Igualmente, la forma en que  $M_1(s)$  influye sobre  $C_2(s)$  se puede escribir:

$$C_2(s) = \Phi_{21}(s)G_{22}(s) + D_{11}(s)G_{21}(s)\bar{M}_1(s) \quad \text{(Ecuación 2.11)}$$

Las ecuaciones 2.11 y 2.12 forman un sistema de dos ecuaciones con cuatro incógnitas,  $D_{11}(s), D_{12}(s), D_{21}(s)$  y  $D_{22}(s)$ . Para poder resolver este sistema se elige comúnmente fijar a la unidad las expresiones para  $D_{11}(s)$  y  $D_{22}(s)$ . Además, se quiere diseñar  $D_{12}(s)$  de tal manera que cuando la salida del controlador 2 cambie, la variable controlada 1 permanezca constante, entonces

esto último hace que  $C_1(s) = 0$  (si una variable tiene que permanecer constante, su variable de desviación es cero). Como buscamos lo mismo para  $D_{21}(s)$ , entonces también  $C_2(s) = 0$ . Lo que resulta luego de aplicar las dos premisas anteriores a las ecuaciones 2.11 y 2.12 es un sistema de dos ecuaciones con dos incógnitas, y a partir de ellas se obtienen, entonces, las dos ecuaciones de diseño del desacoplador para un sistema de control multivariable de tamaño 2 x 2:

$$D_{12}(s) = -\frac{G_{12}(s)}{G_{11}(s)} \quad (\text{Ecuación 2.12})$$

$$D_{21}(s) = -\frac{G_{21}(s)}{G_{22}(s)} \quad (\text{Ecuación 2.13})$$

Definido de esta manera, un bloque desacoplador para un sistema 2 x 2 se construye a partir de las cuatro funciones de transferencias de los procesos que se generan para este tipo de sistemas. Para sistemas de primer orden más tiempo muerto un desacoplador consta generalmente de un parámetro de ganancia, uno de adelanto/retraso y un término de compensación de tiempo muerto. En este punto es importante destacar que el diseño de desacopladores para sistemas de mayor orden que los de 2 x 2 no es sencillo debido al gran número de éstos que se requieren implementar, por ejemplo, para un sistema de control 4 x 4 se necesita diseñar 12 desacopladores y para un sistema de control 5 x 5 se necesitan diseñar 20 desacopladores. Además se necesita recurrir a potentes herramientas computacionales para diseñar tales dispositivos de control.

## **2.3 REDES NEURONALES ARTIFICIALES**

### **2.3.1 Introducción y generalidades**

El término inteligencia artificial (IA) fue acuñado en 1956 por John McCarthy, del Instituto de Tecnología de Massachussets. En su sentido más amplio, este término indica la capacidad de un dispositivo o sistema de realizar los mismos tipos de funciones que caracterizan al pensamiento humano. La posibilidad de desarrollar un artefacto así ha despertado la curiosidad del ser humano desde la antigüedad; sin embargo, no fue hasta la segunda mitad del Siglo XX, cuando esa posibilidad se materializó en herramientas tangibles, apoyado esto más que todo, en los avances de la computación.

Con el desarrollo de algunos campos de la IA apareció un gran número de problemas complejos en los cuales no era conveniente una representación explícita del conocimiento y no se contaba con un procedimiento de razonamiento lógico para resolverlos. Por este motivo los métodos estándar no eran apropiados para resolver tales problemas. Fue así como surgieron las redes neuronales artificiales como estructuras de computación alternativas, creadas con el fin de imitar las funciones del cerebro humano. A partir de lo anterior, se puso de manifiesto la viabilidad de abordar ciertos problemas mediante sistemas que procesen información y aprendan mediante principios tomados de los sistemas cerebrales de los seres vivos. A estos sistemas se los

denomina Redes Neuronales Artificiales, las cuales nacieron de la intención del hombre de simular de manera artificial los sistemas cerebrales biológicos.

### 2.3.2 Inspiración en la Neurociencia

Se ha observado que, mediante reglas aprendidas de la experiencia, el ser humano se manifiesta mucho más eficaz que las computadoras en la resolución de problemas definidos de manera imprecisa, ambiguos, o que requieren procesar gran cantidad de información. Nuestro cerebro alcanza estos objetivos, mediante miles de millones de células simples e interconectadas entre sí, llamadas neuronas, localizadas en el cerebro. A continuación, y con el objeto de introducir los conceptos básicos de las redes neuronales artificiales, de [7] se toman y describen algunos aspectos esenciales relacionados con la neurología (ciencia que se ocupa del sistema nervioso o de cada uno de sus diversos aspectos y funciones especializadas).

El cerebro está compuesto por un elevado número de neuronas que reciben señales electroquímicas de otras neuronas a través de las conexiones sinápticas que unen el axón de las neuronas emisoras y las dendritas de las receptoras (Figura 2.6).

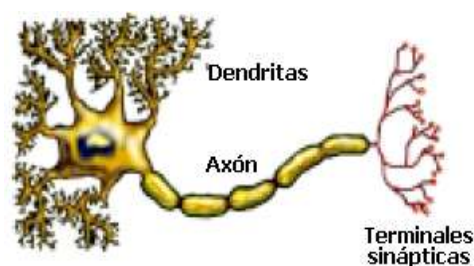


Figura 2.6 Neurona y sus partes. (Tomado de [7])

De acuerdo a las informaciones o impulsos recibidos, la neurona computa y envía su propia señal. Este proceso es controlado por el potencial interno asociado a cada neurona. Si este potencial supera un cierto umbral, se envía un impulso eléctrico al axón; en caso contrario no se envía la señal.

Inspiradas en las características neurofisiológicas descritas anteriormente, las redes neuronales artificiales se pueden enmarcar como modelos computacionales, que están formadas por un gran número de componentes procesadores, o neuronas, dispuestos en varias capas e interconectados entre sí mediante pesos. Los procesadores realizan cálculos simples basados en la información que reciben de los procesadores vecinos. Cabe aclarar que las redes neuronales artificiales usan un proceso de aprendizaje en donde los pesos de las conexiones son ajustados automáticamente para reproducir un conjunto de patrones representativo del problema a aprender. Este aprendizaje también está inspirado en la forma de aprender que tiene lugar en las neuronas humanas, cambiando la efectividad de las sinapsis, de tal manera que la influencia de una neurona a otra varía.

### **2.3.3 Conceptos Básicos.**

Como se mencionó anteriormente, las Redes Neuronales Artificiales son sistemas paralelos para el procesamiento de la información, inspirados en el modo en el que las redes de neuronas biológicas del cerebro procesan ésta. Esto indica que se han intentado plasmar los aspectos esenciales de una

neurona real a la hora de diseñar una neurona artificial. El trabajo de una neurona es simple y único, y consiste en recibir las entradas de las células vecinas y calcular un valor de salida, el cual es enviado a todas las células restantes.

En general, se puede encontrar que una red neuronal (NN) se caracteriza por dos partes fundamentales: La arquitectura de la red y la regla de aprendizaje.

### **2.3.3.1 Arquitectura de la Red.**

La arquitectura de una red neuronal consiste en el tipo, organización y disposición de las neuronas en la red, formando capas o agrupaciones de neuronas. Una capa o nivel, es un conjunto de neuronas cuyas entradas provienen de la misma fuente (que puede ser otra capa de neuronas) y cuyas salidas se dirigen al mismo destino (que puede ser otra capa de neuronas). De esta manera podemos destacar tres tipos de unidades:

1. Las **unidades de entrada** reciben señales desde el entorno (son señales que proceden de sensores u otros componentes del sistema).
2. Las **unidades de salida** envían la señal fuera del sistema (son señales que pueden ser usadas para controlar directamente sistemas).

3. Las **unidades ocultas** son aquellas cuyas entradas y salidas se encuentran dentro del sistema (son señales que no tienen contacto con el exterior).

Usando la figura 2.7 se puede describir con más detalle los componentes de una red neuronal: los círculos representan las neuronas o elementos computacionales de cálculo. Las líneas representan la información ponderada que pasa de un nivel inferior (entrada) a un nivel superior (salida) en la red. Los valores de  $u_i$  con  $i=1:n$  son las entradas a la red, mientras que los valores de  $Y_j^{nn}$  con  $j=1:R$  representan la salida de la red. Este tipo de red es prealimentada, es decir, no hay información de una capa superior que vuelva a una capa inferior. También se encuentra totalmente conectada, es decir, la información de una neurona de la capa inferior se distribuye a través de todas las neuronas de la capa inmediatamente superior. Es esta gran interconexión la que otorga la característica de robustez a la red [Rumehart y McClelland; 1987] [8].

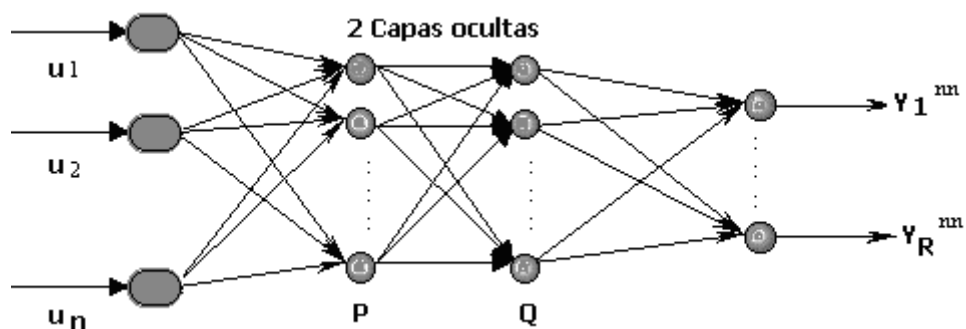


Figura 2.7 Ejemplo de una red con dos capas ocultas. (Tomado de [9])



De esta manera, la arquitectura de una red depende del número de variables en la capa de entrada, del número de capas ocultas de neuronas, del número de neuronas por cada capa oculta y del número de variables de salida en la última capa. Siendo todos factores importantes a la hora de determinar la configuración de una red.

Cada una de las neuronas (círculos) mostradas en la figura 2.7 posee los siguientes elementos [Aguiar y Maciel; 1998] [8], los cuales se pueden observar con más detalle en la figura 2.8. En esta última se muestran los elementos que conforman cada una de las neuronas de las capas ocultas y de salida. En una red prealimentada, la neurona  $j$  de la capa  $K + 1$  tiene como valores de entrada,  $a_i^k$  con  $i = 1:M$ , las salidas de todas las neuronas (red totalmente conectada) de la capa (anterior)  $k$  ponderadas en el término  $W_{ij}^k$  con  $i = 1:M$ .

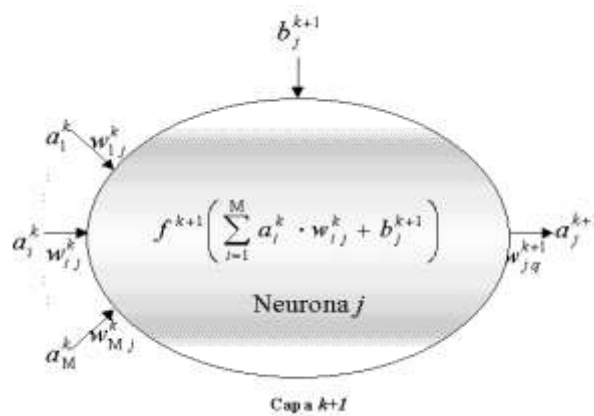


Figura 2.8 Elementos de la neurona  $j$  de la capa  $k+1$ . (Tomado de [9])

Además, tiene un término que representa la tendencia o estado actual de activación de la neurona,  $b_j^{k+1}$ , con una ponderación unitaria. De esta manera, la entrada neta a la neurona  $j$  de la capa  $k+1$  estará dada por:

$$\psi_j^{k+1} = \sum_i^M a_i^k \cdot w_{ij}^k + b_j^{k+1} \quad (\text{Ecuación 2.14}) [9]$$

con  $j=1$  hasta el número total de neuronas de la capa  $k+1$  hasta la capa de salida. La entrada neta a la neurona es evaluada en la función de activación o función de transferencia de la neurona, la cual entrega el estado de activación actual de la neurona. Las funciones de activación pueden ser de diversos tipos, siendo las más utilizadas, para las redes con al menos una capa oculta [10]: la función tangente hiperbólica y la función sigmoideal. Dichas funciones tienen como imagen un rango continuo de valores dentro de los intervalos  $[0,1]$  y  $[-1,1]$ , respectivamente, y vienen dadas por las siguientes expresiones:

$$\text{Función tangente hiperbólica: } f^{k+1}(\psi_j^{k+1}) = \frac{2}{1 + e^{-2 \cdot \psi_j^{k+1}}} - 1 \quad (\text{Ecuación 2.15}) [9]$$

$$\text{Función sigmoideal: } f^{k+1}(\psi_j^{k+1}) = \frac{1}{1 + e^{-\psi_j^{k+1}}} \quad (\text{Ecuación 2.16}) [9]$$

Al evaluar la entrada neta a la neurona  $j$  de la capa  $k+1$  (Ecuación 2.14) en la función de activación (Ecuación 2.15 ó 2.16), se obtiene el estado actual de activación de la neurona  $j$ , que es distribuido a todas las neuronas de la capa siguiente,  $k+2$  ya que representa la salida efectiva de la neurona  $j$  en la capa  $k+1$  (ver figura 2.7):

$$a_j^{k+1} = f^{k+1}(\psi_j^{k+1}) \quad (\text{Ecuación 2.18}) [9]$$

La capa de entrada presentada en la figura 2.7 tiene por único objetivo distribuir la información a las neuronas de la primera capa oculta; es por eso que esta capa no tiene valores ponderadores de la información de entrada, ni función de transferencia.

En cuanto a las capas ocultas, se tiende a pensar que un gran número de éstas garantiza mejores resultados. Sin embargo, se ha mostrado en [11] que con al menos una capa oculta de neuronas, con un número suficientes de neuronas en dicha capa, es suficiente para aproximar cualquier función continua en  $\mathcal{R}^n$ .

El último aspecto a definir en la arquitectura de una red neuronal prealimentada de múltiples capas tiene relación con el número de neuronas de cada una de las capas involucradas. La capa de entrada tiene un número de neuronas determinado por el número de variables de entrada a la red, es decir, el número de neuronas en la capa de entrada es igual al número de variables de entrada. De la misma manera, el número de neuronas en la capa de salida es igual al número de variables de salida.

En general, determinar el número óptimo de neuronas en la(s) capa(s) oculta(s) es un procedimiento que depende de la aplicación que se esté llevando a cabo. Por ejemplo, el principal objetivo de utilizar redes de gran complejidad (muchas capas y/o muchas neuronas) es maximizar la precisión del mapa de entradas/salidas (IO's). Sin embargo, si se procede en tal dirección hay un alto precio que pagar. En primer lugar, con un gran número de neuronas hay una

fuerte tendencia al sobre ajuste de datos y un pobre desempeño con datos desconocidos. En segundo lugar, un gran número de parámetros implica una gran posibilidad de error en la estimación de estos parámetros. Es por ello que al momento de decidir cuántas neuronas habrá en la(s) capa(s) oculta(s), se debe realizar un esfuerzo experimental con el fin de reducir el tamaño y la complejidad de la red. En [10] se sugiere probar con un número grande de neuronas y luego disminuirlo hasta encontrar la configuración con menos neuronas (y capas ocultas) que permite realizar una representación del mapa IO's, con un error mínimo aceptado.

#### **2.3.4 Métodos de Aprendizaje.**

La característica más interesante de las redes neuronales artificiales es su capacidad de aprendizaje. El aprendizaje en las redes neuronales artificiales consiste en determinar los valores precisos de los pesos para todas sus conexiones, que la capacite para la solución eficiente de un problema.

Las redes neuronales artificiales usan métodos de aprendizaje basados en ejemplos. Desde este punto de vista el conjunto de datos usados para el entrenamiento debe tener dos importantes características: ser significativo, de modo que haya un número suficiente de ejemplos que le permitan a la red adaptar sus pesos de forma eficaz, y ser representativos, es decir, que todo el intervalo de posibles salidas esté suficientemente representado.

Dependiendo del esquema de aprendizaje y del problema a resolver se pueden distinguir tres tipos: aprendizaje supervisado, aprendizaje auto-organizativo o no supervisado y aprendizaje por reforzamiento.

### 2.3.4.1 Aprendizaje Supervisado

Para este tipo de aprendizaje, se dice que hay un profesor externo encargado de determinar si la red se está comportando de forma adecuada, mediante la comparación entre la salida producida y la esperada, y de actuar en consecuencia modificando apropiadamente los valores de los pesos. Esto implica que el conjunto de datos usados para el entrenamiento deben tener dos tipos de atributos: los datos propiamente dichos,  $u$ , y cierta información relativa a la solución deseada del problema,  $y^d$ . En este tipo de sistemas, como se muestra en la figura 2.9, se tiene disponible un conjunto de entrenamiento  $u(k), y^d(k) \stackrel{i_j}{k=1}$  que permite al supervisor evaluar el desempeño del aprendizaje mediante algún criterio de optimización.

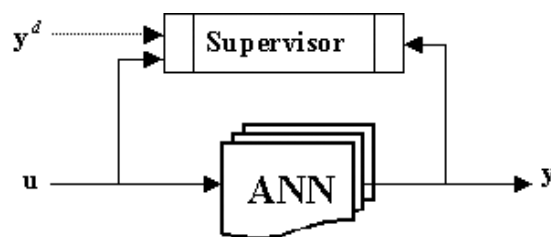


Figura 2.9 Aprendizaje Supervisado. (Tomado de [9])

Cada vez que un ejemplo es introducido y se procesa para obtener una salida, dicha salida se compara con la salida que debería haber producido, y de la que

se dispone al estar incluida esta información en el conjunto de aprendizaje. La diferencia entre ambas influirá en cómo se modificaran los pesos. En la mayoría de los casos lo que se hace es minimizar la sumatoria del error cuadrático de la salida mediante la adaptación de los parámetros de la red (Ec. 2.43), donde  $y_j$  es la j-ésima salida para todo instante discreto  $k$ .

$$J_{\theta}(k) = \sum_{j=1}^p (y_j^d(k) - y_j(k))^2 = \sum_{j=1}^p e_j^2(k) \quad (\text{Ecuación 2.43}) [9]$$

Las redes multicapa usan este tipo de aprendizaje. Existen otros criterios de optimización que incluyen entre otros aspectos, la energía de excitación o el tiempo de entrenamiento. Sin embargo,  $J_{\theta}$  establece eficiencia computacional y buenos resultados, debido a los efectos netos sobre la salida y su naturaleza cuadrática.

Dentro del sistema de aprendizaje supervisado se pueden diferenciar dos enfoques, de acuerdo a la forma de implementar los algoritmos de entrenamiento [12]:

*Aprendizaje batch* o aprendizaje por lotes, provee al algoritmo de entrenamiento de todos los datos  $u(k), y^d(k)_{k=1}^T$  disponibles, para adaptar los parámetros de la red en un único instante. De esta forma se considera un índice a optimizar de la forma:

$$J_{\theta} = \sum_{k=1}^{t_j} e_y^T(k) \cdot Q \cdot e_y(k) \quad \text{(Ecuación 2.20) [9]}$$

donde  $e_y = [e_1, \dots, e_p]^T$  es el error de la salida vectorial y  $Q$  es matriz definida positiva que pondera la importancia de alguna salida en particular.

*Aprendizaje instantáneo* o en línea, provee a los algoritmos de entrenamiento del dato  $u(k), y^d(k)$  presente en el instante  $k$  para adaptar los parámetros de la red. Es así como este tipo de aprendizaje optimiza el índice:

$$J_{\theta} = e_y^T(k) \cdot Q \cdot e_y(k) \quad \text{(Ecuación 2.21) [9]}$$

Estableciendo la ventaja que la estructura actualiza sólo los parámetros que contribuyen a la salida, es decir, genera un aprendizaje de naturaleza local, debido a que diferentes entradas son transformadas por diferentes grupos de parámetros.

#### **2.3.4.2 Aprendizaje no Supervisado**

En este tipo de aprendizaje los datos del conjunto de aprendizaje sólo tienen información de los ejemplos, y no está presente un supervisor externo, como se muestra en la figura 2.10, que permita guiar en el proceso de aprendizaje. La red modificará los valores de los pesos a partir de información interna. Debido a esto, en general dichos sistemas, son capaces de descubrir por sí

solos relaciones de interés tales como: patrones, regularidades, correlaciones, o categorías en los datos de entrada, trasladando tales relaciones descubiertas, a la salida.

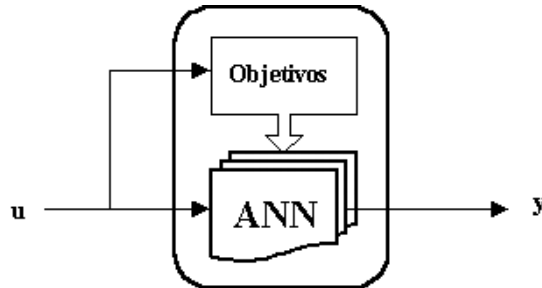


Figura 2.10 Aprendizaje no supervisado. (Tomado de [9])

La denominación de sistemas de aprendizaje no supervisado se debe a que mientras el sistema descubre características del espacio de entrada, adapta sus parámetros y estructura mediante la optimización de alguna función objetivo predeterminada, sin realimentación instructiva de un agente externo, es decir, se establece un aprendizaje en lazo abierto en donde el ajuste depende únicamente de los datos recibidos como entrada.

### 2.3.4.3 Aprendizaje por Reforzamiento

Es una variante del aprendizaje supervisado en el que no se dispone de información concreta del error cometido por la red para cada dato o ejemplo de aprendizaje, sino que simplemente se determina si la salida producida para dicho ejemplo es o no adecuada. En el caso general la señal de reforzamiento  $r$ , (ver figura 2.11), toma estados correspondientes a fallas o sucesos, descritos



por valores binarios o normalizados, la señal de reforzamiento interno  $\rho$  representa la evaluación del comportamiento de la red. Luego, el objetivo de este sistema de aprendizaje, contenido en el bloque denominado predictor adaptativo, es optimizar un índice, función de  $r$  sobre  $N$  pasos futuros,

$$J(u_k, y_k, K) = \sum_{j=k}^N p_k - \gamma^{j-k} \cdot r(u_j, y_j, j) \quad (\text{Ecuación 2.23}) [9]$$

donde  $p_k$  es la secuencia de predicción. Dentro de los problemas abordados por este sistema de aprendizaje, el más general ataca a sistemas de dinámica compleja, donde las señales dependen de las salidas pasadas [10].

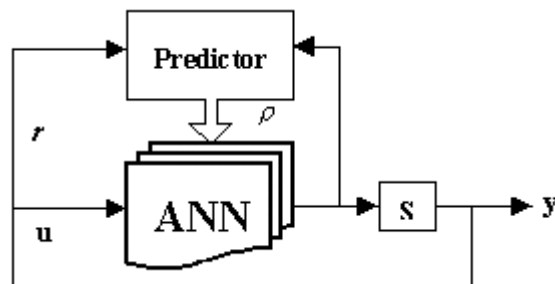


Figura 2.11 Aprendizaje por reforzamiento. (Tomado de [9])

Luego de haber descrito los tres métodos de aprendizaje más usados, se debe mencionar que éstos, de manera general, tendrán que cumplir con un cierto criterio de convergencia; de no ser así se repite el proceso y todos los datos o ejemplos del conjunto de entrenamiento vuelven a ser introducidos. El criterio de convergencia depende del tipo de red utilizado o del tipo de problema a resolver. Para determinar cuándo finaliza un proceso de aprendizaje se usan los siguientes criterios [10]:

*Realización de un número fijo de ciclos:* se decide con anterioridad las veces que será introducido el conjunto de entrenamiento y una vez superado dicho valor se detiene el proceso y se da por aceptada la red resultante.

*Alcanzar un valor óptimo del error:* se decide con anterioridad un valor aceptable para el error cometido por la red y se para el proceso de aprendizaje cuando la red produzca un valor de error por debajo del prefijado.

*Cuando la modificación de los pesos sea irrelevante:* cuando se usan modelos que hacen que las conexiones vayan modificándose cada vez con menos intensidad. Si el proceso de aprendizaje continúa, llegará un momento en que ya no se producirán variaciones de los pesos de ninguna conexión; en ese momento se establece que la red ha convergido y se detiene el proceso de aprendizaje.

### **2.3.5 Reglas de Aprendizaje.**

Las reglas de aprendizaje son los mecanismos mediante los cuales se van adaptando y modificando todos los parámetros de la red. Existen varias reglas de aprendizaje y la mayoría de los modelos están orientados en reproducir el esquema de aprendizaje biológico. Sin embargo, el proceso de aprendizaje biológico sigue siendo mucho más complejo que cualquier regla de aprendizaje desarrollada hasta este momento.

A continuación presentaremos algunas de las reglas de aprendizaje mas conocidos:

### **2.3.5.1 Regla de Hebb's**

Este es el modelo de aprendizaje más antiguo, fue introducido por Donald Hebb; está basado en observaciones fisiológicas. Su regla básica es la siguiente [9]: “si una neurona recibe una señal de entrada de otra neurona y ambas son altamente activas (matemáticamente tienen el mismo signo), los pesos entre neuronas se reforzarán”. Esta regla de aprendizaje trabaja bien solamente si las entradas son ortonormales. Solamente vectores ortonormales pueden ser perfectamente clasificados con  $n$  igual a la dimensión del vector de entrada.

### **2.3.5.2 La Regla Delta**

Se trata de una variación de la regla de Hebb's en la cual se usa el método de los mínimos cuadrados. La regla está basada en la simple idea de modificar la fuerza de las conexiones de entrada para reducir la diferencia (delta) entre el valor deseado de la salida y el valor de la salida actual de un proceso. Esta regla trabaja bien si los vectores de entrada son linealmente independientes; sino, por lo menos encontrará una matriz de pesos que minimice el cuadrado del error entre la salida deseada y la salida actual. Este método de encontrar los pesos garantiza la convergencia en un número finito de pasos. Usando la

regla delta solo  $n$  vectores linealmente independientes pueden ser perfectamente clasificados, siendo  $n$  la dimensión de los vectores de entrada. Un requerimiento más severo para los vectores de entrada es que deben ser linealmente separables, de otra manera el proceso oscilará continuamente. El set completo de entrenamiento puede ser usado repetidamente hasta que el error es lo suficientemente pequeño para satisfacer al usuario. Estas reglas aplican a las redes de propagación hacia delante con una capa simple. [9]

### **2.3.5.3 Backpropagation**

Uno de los métodos más usados para el aprendizaje con redes neuronales es el que se realiza siguiendo el enfoque de la Retropropagación, Backpropagation [Rumelhart, 1986], el cual necesita introducir a priori el número y conexiones de las unidades ocultas, y determinar los pesos por minimización de un error.

Este tipo de redes aprende un conjunto predefinido de pares de entrada y salidas dadas como ejemplo (aprendizaje supervisado), empleando un ciclo propagación-adaptación de dos fases. Una vez que se ha aplicado una trama de entrada como estímulo para la primera capa de unidades de la red, ésta se va propagando a través de todas las capas superiores hasta generar una salida. La señal de salida, se compara entonces con la salida deseada, y se calcula una señal de error para cada unidad de salida. Las señales de error se transmiten entonces hacia atrás, partiendo de la capa de salida, hacia todos los nodos de la capa intermedia que contribuyan directamente a la salida. Sin

embargo, las unidades de la capa intermedia sólo reciben una fracción de la señal total de error, basándose aproximadamente en la contribución relativa que haya aportado la unidad a la salida original. De ahí viene el nombre de algoritmo de retropropagación, pues los errores para las neuronas de la red se retropropagan hacia todas las neuronas de la capa anterior. Este proceso se repite capa por capa, hasta que todos los nodos de la red, hayan recibido una señal de error que describa su contribución relativa al error total. Basándose en la señal de error percibida, se actualizan los pesos de conexión de cada unidad, para hacer que la red converja hacia un estado que permita codificar todas las tramas de entrenamiento.

El principal defecto de este enfoque consiste en la búsqueda de la mejor arquitectura a través de prueba y error.

#### **2.3.5.4 Redes de Hopfield**

El algoritmo de aprendizaje de este tipo de red es muy diferente a los otros presentados anteriormente, ya que las estructuras de estas redes tienen retroalimentación de otras capas.

En las redes de Hopfield, los pesos no son cambiados, son preseleccionados y se mantienen fijos. Cuando una entrada es aplicada, la red se ajusta a un estado estable de actividad lo cual representa la solución.

Las oscilaciones son menores que en una red sin retroalimentación. Se usa normalmente con entradas binarias. Si se usan muchos patrones, la red puede converger a un patrón desconocido, pero Hopfield demostró en [13] que esto no ocurre frecuentemente si el número de patrones almacenados es menor que el 15% del número de nodos en la red y esta a su vez es la limitante de este tipo de redes.

#### **2.3.5.5 Random Activation Weight Neural Net (RAWN).**

Este algoritmo de entrenamiento fue propuesto por Te Braake y Van Straten en 1994 [14]. El procedimiento usual para entrenar redes neuronales hacia delante, es backpropagation, que a pesar de las mejoras como aceleración, velocidad de aprendizaje adaptativo, inicializaciones especiales y redes auxiliares, backpropagation es lento y tiene pocas propiedades de convergencia. Otros métodos no proveen un mejor desempeño, sin embargo, el entrenamiento de redes neuronales, es aún, un cuello de botella en aplicaciones de tiempo real. La novedad del método RAWN, es que se pueden obtener muy buenos mapeados, solamente tomando aleatoriamente los pesos en la capa oculta. En aquel caso, la estimación de los pesos en la capa de salida llega a estar en parámetros lineales que se pueden resolver fácilmente por los métodos ordinarios estándares de mínimos cuadrados. Esto significa que entrenar la red tomará sólo una iteración, la cual, comparada con backpropagation que puede tomar miles de iteraciones, es un gran avance, y también un factor conveniente en operaciones de tiempo real.

### 2.3.6 Validación

Una vez que ha terminado el proceso de aprendizaje y los pesos de la red neuronal han sido calculados, es importante comprobar la calidad del modelo resultante. Por ejemplo, en el caso del aprendizaje supervisado, una medida de la calidad puede darse en términos de los errores entre los valores de salida deseados y los obtenidos por la red neuronal. Algunas medidas estándar del error son la suma de los cuadrados de los errores, la raíz cuadrada del error cuadrático medio y el error máximo. Generalmente para comprobar que el entrenamiento fue exitoso bastará con realizar una comparación gráfica entre las curvas de los datos reales de las salidas del proceso con las salidas predichas por la red [15].

En ocasiones también es necesario realizar una validación cruzada para obtener una medida de la calidad de predicción del modelo. Por este propósito, los datos obtenidos se pueden dividir en dos partes, una parte destinada al entrenamiento de la red y otra parte a la comprobación (validación). Cuando el error de comprobación es mucho mayor que el error de entrenamiento, se produce un problema de sobreajuste durante el proceso de entrenamiento. Al fenómeno anterior se le denomina *sobreaprendizaje* [10] y puede ser producido por un entrenamiento muy riguroso, el cual anula la capacidad de generalización de la red, o por la utilización de demasiadas neuronas ocultas, las cuales ajustan con mucha exactitud los patrones de entrenamiento.

### **3. DISEÑO DEL CONTROLADOR NEURONAL**

Este capítulo registra la manera en la cual, basándose en una estrategia de control predictivo, el controlador neuronal es diseñado. Además se detalla la formulación y creación del algoritmo de control en Matlab® v.6.5 y Simulink® v.5.0

Se puede considerar que en la actualidad el objetivo de todo sistema de control consiste en actuar sobre las variables manipuladas, de forma que puedan satisfacerse múltiples y cambiantes criterios de funcionamiento (económicos, medioambientales o de calidad) en presencia de cambios en las características del proceso. Las técnicas de control predictivo parecen ser eficientes herramientas para afrontar estos retos y en su forma más general, aceptan variedad de modelos y funciones objetivo o restricciones, siendo la metodología que actualmente puede reflejar más directamente los múltiples criterios de funcionamiento en las industrias de procesos.

Por las razones antes expuestas, la estrategia de control predictivo es la estrategia elegida para el diseño y desarrollo del controlador neuronal para sistemas multivariables  $2 \times 2$ .



### **3.1 CONTROL PREDICTIVO**

El control predictivo es una metodología de control avanzado que ha alcanzado en la actualidad una gran aceptación industrial, esencialmente debido a que permite enfrentar problemas de control de sistemas que presentan una dinámica compleja tales como los sistemas multivariables y no lineales. El control predictivo fue desarrollado fundamentalmente con base en diversos algoritmos que surgieron a finales de los años 70, los cuales usaban explícitamente un modelo dinámico del proceso para predecir el efecto de futuras acciones de control en la salida.

Existen diversas estrategias de control predictivo, tales como el Control Predictivo Basado en Modelos (MPC) que aparece a finales de los 70 , el Control Predictivo Generalizado (GPC) desarrollado a mediados de los 80 por el grupo del profesor D. W. Clarke en la Universidad de Oxford o el Dynamic Matrix Control (DMC) desarrollado a fines de los años 70 por ingenieros de Shell Oil Co. [16] Todas estas estrategias son muy similares y se basan en los mismos principios diferenciándose esencialmente en los objetivos de control, el tipo de procesos a controlar y las funciones objetivo. En este punto se estudiará básicamente el control predictivo basado en modelos.

#### **3.1.1 Control Predictivo Basado en Modelos**

Esta estrategia hace uso explícito de un modelo del proceso para obtener la señal de control, mediante la minimización de una cierta función objetivo. Este

tipo de mecanismo de control, además de garantizar la operación estable del proceso, satisface una serie de criterios secundarios de control tales como económicos, de seguridad o de calidad del producto final. Muchos de estos criterios admiten una representación matemática muy natural bajo la forma de funciones objetivo y el control predictivo basado en modelos constituye una metodología que es capaz de manejar esta representación de forma sistemática.

Las ideas básicas que aparecen en mayor o menor medida en toda familia de controladores predictivos son:

1. Uso explícito de un modelo para predecir la salida del proceso en futuros instantes de tiempo.
2. Cálculo de las señales de control minimizando cierta función objetivo
3. Estrategia deslizante, de manera tal, que en cada instante de tiempo el horizonte se va desplazando hacia el futuro.
4. Aplicación de la primera señal de control desechando el resto y repitiendo el cálculo para cada instante de muestreo.

Los distintos algoritmos de control predictivo basados en modelos (MPC) difieren entre sí, casi exclusivamente en el modelo usado para representar el proceso y las perturbaciones, y en la función a minimizar. El MPC es un tipo de control de naturaleza abierta que ha encontrado gran aceptación tanto en aplicaciones industriales como en el mundo académico. En la actualidad

existen numerosas aplicaciones de controladores predictivos funcionando con éxito, especialmente en la industria de procesos [17].

### 3.1.2 Estrategia del Controlador

La estrategia utilizada por los MPC se basa en el siguiente algoritmo, (Tomado de [18]):

1. Haciendo uso de un modelo del proceso, en cada instante de tiempo se predicen las futuras salidas en un determinado horizonte de tiempo llamado horizonte de predicción. Estas salidas predichas  $y_p(k+t)$  para  $k=1\dots N$ , dependen de los valores conocidos hasta el instante  $t$  (entradas y salidas pasadas) y de las señales de control futuras  $u(k+t)$  que se pretenden mandar al sistema y que son las que se requieren calcular.
2. El conjunto de señales de control futuras, se calcula optimizando un determinado criterio, en el que se pretende mantener el proceso lo más próximo posible a la trayectoria de referencia  $w(k)$  (que puede ser directamente el set-point o una suave aproximación a éste). Este criterio suele tomar la forma de una función cuadrática de los errores entre la salida predicha y la trayectoria de referencia también predicha, incluyendo en muchos casos el esfuerzo de control.

- La señal de control  $u^*/t$  es enviada al proceso mientras que las siguientes señales de control calculadas son desechadas, puesto que en el siguiente instante de muestreo ya se conoce  $y^{t+1}$  y se repite el paso 1 con este nuevo valor y todas las secuencias son actualizadas. Se calcula por tanto  $u^{t+1/t+1}$  (que en principio será diferente a  $u^{t+1/t}$  al disponer de nueva información), haciendo uso del concepto de horizonte deslizante.

La figura 3.1 muestra un diagrama en bloques que representa la estructura básica del MPC.

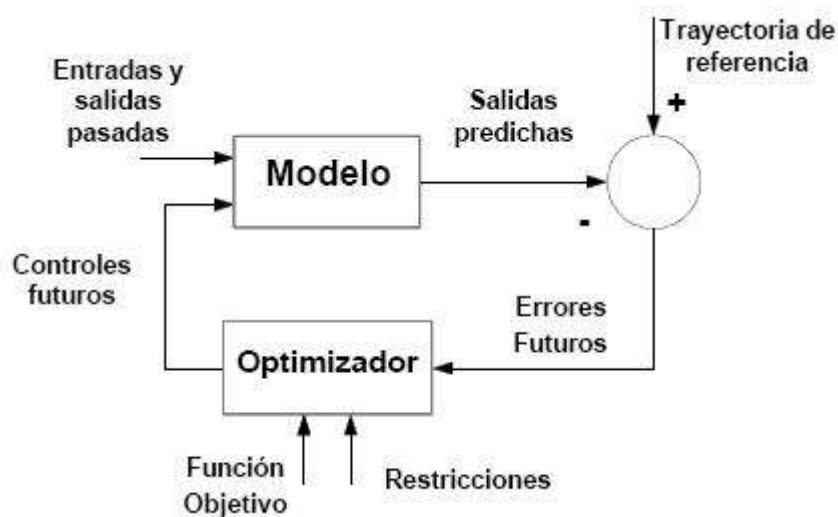


Figura 3.1 Estructura básica de un MPC. (Tomado de [18])

Para predecir las salidas futuras se usa un modelo del proceso a controlar. Las salidas futuras se basan a su vez en las futuras señales de control propuestas por el optimizador. Como es de suponer, el modelo del proceso juega un papel importante y el desempeño del controlador depende en gran medida de la exactitud de éste. Se puede observar en la figura 3.1 que, quien proporciona

las acciones de control es el optimizador. También en esta figura se pueden identificar tres elementos principales: el ***predictor***, el cual es un modelo del proceso que permite obtener las salidas futuras de la planta para estructurar la ley de control, el ***optimizador***, que tiene como propósito la minimización de la función objetivo y de esta forma generar las señales de control y el ***modelo*** a ser controlado.

### 3.1.3 Función Objetivo

El tipo de función objetivo varía entre los diferentes algoritmos MPC pero en general lo que se busca es minimizar el error entre la salida del sistema y la señal de referencia en un determinado horizonte de predicción.

La expresión general de la función objetivo se muestra en la siguiente ecuación

$$J_{1,2} = \sum_{i=N1}^{N2} \theta(i) \left[ y_{r,2}(i) - \hat{y}_{1,2}(i) \right]^2 + \sum_{i=1}^{Nu} \lambda(i) u_{1,2}(i) - u_{1,2}(i-1) \quad \text{(Ecuación 3.1.) [9]}$$

donde:

$u(i)$  es la salida del controlador (variable desconocida a optimizar)

$u(i-1)$  es la salida del controlador un tiempo de muestreo anterior.

$y_r$  es el set-point.

$\hat{y}(1)$  es la salida predicha del proceso.

$\theta$  y  $\lambda$  son factores para las predicciones y acciones de control respectivamente

$N1$  y  $N2$  definen el horizonte de predicción

$Nu$  define el horizonte de control

Los subíndices 1 y 2 corresponden a las dos señales de control

La función objetivo consta de dos partes. La primera toma en consideración el error entre el set-point y la salida predicha; es decir, evalúa cuanto diferirá la salida del proceso respecto al set-point. Esta magnitud se calcula en un intervalo de tiempo futuro determinado por  $N1$  y  $N2$ , llamado horizonte de predicción. El horizonte de predicción determina hasta que punto del futuro el controlador es capaz de ver.

El primer término en la ecuación 3.1 es una suma de errores al cuadrado debido a que lo que realmente importa es la magnitud de los errores. El factor  $\theta$ , se usa para darle más importancia a casos particulares en todo el grupo de predicciones.

La segunda parte considera el cambio actual de las variables manipuladas. Esta magnitud es importante dado que es una indicación de que tan agresivo es el controlador cuando aplica una acción correctiva. Si el controlador no puede *ver* muy lejos en el futuro, lo que corresponde a un horizonte de predicción pequeño, tenderá a llevar el proceso hacia el punto de referencia o set-point en forma muy agresiva. Es por eso que se hace uso del factor,  $\lambda$  o factor de agresividad, el cual permite regular la agresividad del controlador para prevenir cambios súbitos que vayan en detrimento de la válvula u otro elemento final de control. También se puede prevenir la presencia de sobre impulsos que

induzca grandes oscilaciones en la variable controlada. Igual que la primera parte, esta también es una suma de errores al cuadrado y se toman de acuerdo al horizonte de control, definido por  $N_U$ .

El horizonte de predicción ( $P$ ) puede ser expresado en unidades de tiempo real como:

$$P = step\_time * \sqrt{2 - (N_U - 1)} \quad (\text{Ecuación 3.2}) [9]$$

El horizonte de predicción debe ser lo suficientemente grande para que el controlador sea capaz de advertir la salida del proceso hasta que se obtenga la condición de estado estable. Sin embargo, mientras más grande sea el horizonte de predicción mas tiempo de cómputo es requerido.

Normalmente se le da a todas las predicciones igual *importancia* o peso de manera que todos los componentes de  $\theta$  son 1. El horizonte de control  $N_U$  se hace habitualmente 1 cuando se trabaja con modelos imperfectos del proceso ya que un valor grande de este factor tiende a generar inestabilidad. Además, la minimización de múltiples valores de  $u$  aumenta sensiblemente el tiempo de cómputo.

Para esta investigación  $\theta = 1$  y  $N_U = 1$  lo que indica que las predicciones tienen el mismo peso y el valor de  $u$  se mantiene constante a través de todo el horizonte de predicción.

El factor  $\lambda$  usado para regular la agresividad del controlador se selecciona luego de realizar unos pocos ensayos que se describirán posteriormente.

La expresión del esquema MPC para esta investigación queda de la siguiente forma:

$$J_{1,2} = \sum_1^{60} \theta(i) \left[ y_{r1,2}(i) - \hat{y}_{1,2}(i) \right]^2 + \lambda(i) u_{1,2}(i) - u_{1,2}(i-1) \quad \text{(Ecuación 3.2)}$$

### 3.1.4 Obtención de la Ley de Control

Para obtener la ley de control, es necesario calcular las salidas predichas  $y_p(k+t)$ , en función de los valores pasados de entradas y salidas y de las señales de control futuras. Para este efecto se usa el modelo que se haya elegido previamente y luego el nuevo valor de la salida predicha, se sustituye en la función objetivo representada en la ecuación 3.1 obteniéndose una expresión cuya minimización conduce a la generación de las señales de control requeridas (valores de  $u(k+t)$ ).

## 3.2 DISEÑO DEL CONTROLADOR NEURAL

En este punto se consignan las etapas de diseño y experimentación del sistema de control predictivo desarrollado en el presente trabajo. Se detallará la evolución cronológica del controlador, partiendo de la idea de dividir la implementación del controlador en fases básicas que faciliten el trabajo, como



son la de recolección de datos, el entrenamiento, predictor, optimizador no lineal, y el control.

### **3.2.1 Fase de Recolección de Datos**

El proceso de recolección de datos es una de las fases básicas en el desarrollo del controlador neuronal.

Como consecuencia de la no linealidad de los procesos a evaluar, se necesita que los datos que se recolecten, examinen correctamente el comportamiento dinámico del proceso, tarea necesaria para que el entrenamiento sea exitoso, y la red sea capaz de hacer buenas predicciones ante cualquier entrada del sistema, inclusive, aún si son desconocidas para ella.

Para ello, se crea un modelo de Simulink, que incorpora como entrada un generador de pulsos de amplitud aleatoria, en el que se puede especificar el número de pulsos que se desean generar, su duración y el rango mínimo y máximo de valores que puede tomar.

Para todos los procesos se generan pulsos, cada uno con una duración igual al tiempo de establecimiento del proceso que se analiza, obteniendo la información necesaria del comportamiento de la salida del proceso ante cada entrada hasta estabilizarse, lo que le permite a la red en el proceso posterior de entrenamiento, tener la posibilidad de aprender como será la salida del proceso ante cada entrada, lo que se traducirá en una correcta predicción.

El modelo incluye también unos recolectores de datos, que son bloques diseñados para guardar los conjuntos de valores actuales y pasados de las variables manipuladas y controladas usadas para entrenar la red.

### **3.2.2 Fase de Entrenamiento**

Una vez almacenados los conjuntos de datos de entrenamiento se está en capacidad de comenzar a entrenar la red.

La red neuronal se entrenará con los datos provenientes del modelo dinámico del proceso, tomando la implementación del controlador neuronal con estos datos.

Para diseñar y realizar el entrenamiento, se tomó como base teórica el método de entrenamiento RAWN (Random Activation Weight Neural Net) descrito en la Sección 2.3.5.6.

Como objetivo básico de este método de entrenamiento se tiene la rapidez, ya que con solo una iteración se logra hallar los pesos de la red neuronal, tomando aleatoriamente los pesos de la capa oculta y a partir del cálculo de la salida de esta capa, (que no es más que aplicarle la función de activación a la sumatoria de las entradas por los pesos de esta matriz) y de los datos de la salida del proceso, se hallan con el método de mínimos cuadrados los pesos de la capa de salida.

Como los pesos de la capa oculta se asignan aleatoriamente, es posible que hayan unos pesos que hagan que la red neuronal tenga un menor error de predicción y por ende una predicción más cercana a la salida del proceso, para esto se hacen diversos intentos con distintas matrices de pesos aleatorias, escogiendo la matriz de pesos que de un error de predicción mínimo.

Otro de los objetivos del entrenamiento es encontrar el número óptimo de neuronas que debe tener la capa oculta, para lo cual se hacen pruebas con distintos números de neuronas, empezando desde una hasta 8 en este caso, y escogiendo el que arroje mejores predicciones. En realidad, lo que se tiene es un entrenamiento que combina todos los requerimientos para tener una predicción más cercana a la variable controlada, ya que la búsqueda de estos parámetros es integrada, siendo que para cada prueba con un conjunto de neuronas distintas se hacen los 16 intentos con distintas matrices aleatorias, hasta hallar en alguno de esos intentos el número de neuronas óptimo y la matriz aleatoria que hacen la mejor predicción.

### **3.2.3 Fase para Evaluar la Calidad de la Predicción**

El diseño del predictor, se basa en el hecho que los sistemas dinámicos, no responden inmediatamente a las entradas aplicadas. Por lo tanto la respuesta del proceso  $c_{1,2}(n)$  es una función de las entradas  $m_{1,2}(n-1)$  y las salidas  $c_{1,2}(n-1)$ , e inclusive  $c_{1,2}(n-2)$ ,  $c_{1,2}(n-3)$ ..., lo que refuerza el hecho que la

red neuronal trabaje como predictor. El éxito de esta fase, es decir, lograr un buen predictor, depende totalmente de la selección de un conjunto de datos que represente bien el comportamiento de los procesos y de la realización de un buen entrenamiento, ya que depende de ellos que la predicción sea la más exacta posible a la salida del proceso.

Para cada salida del controlador, el predictor neuronal calcula la salida del proceso tantos pasos en el futuro como lo determine el horizonte de predicción, el predictor se implementa en un algoritmo que hace las predicciones de las salidas del proceso (variables controladas) basado en sus propias predicciones. Como las predicciones dadas por el predictor no son una copia fiel de las variables controladas, existe una diferencia entre estas variables a la que llamaremos error de predicción, que se mantiene fijo durante cada predicción y se le suma a las futuras predicciones, con el fin de evitar que el error llegue a ser acumulativo, controlar los efectos de las perturbaciones y para corregir las inexactitudes del modelo del proceso. Este procedimiento se constituye en la retroalimentación del sistema.

#### **3.2.4 Fase de Control**

Esta fase se desarrolló paralelamente con las descritas anteriormente; como su nombre lo indica, constituye la columna vertebral del controlador neuronal. Básicamente se compone de un algoritmo y un modelo de Simulink que integran a las fases de predicción y optimización así como algunas características inherentes al controlador como el tiempo de muestreo y las

condiciones iniciales con el objeto de llevar a cabo la acción de control. El tiempo de muestreo del controlador define cada cuanto tiempo el controlador debe calcular su salida óptima.

La rutina principal del neurocontrolador está implementada con un bloque de *S-function* para que pueda ser integrada al modelo de control en lazo cerrado.

### **3.3 ALGORITMO DEL CONTROLADOR NEURONAL**

En esta sección se describirá en forma detallada y secuencial cada uno de los algoritmos que se diseñaron para la implementación del controlador neuronal, así como los que se crearon para obtener los datos de los modelos. En el Anexo A se muestran los códigos de fuente de los algoritmos desarrollados en esta investigación.

#### **3.3.1 Simulación del Proceso en Lazo Abierto**

***Lazo\_Abierto\_2X2\_N*** es el programa en Simulink que se encarga de realizar las simulaciones del proceso en lazo abierto y recolecta los datos del proceso para conformar los conjuntos de muestras usados para el entrenamiento.

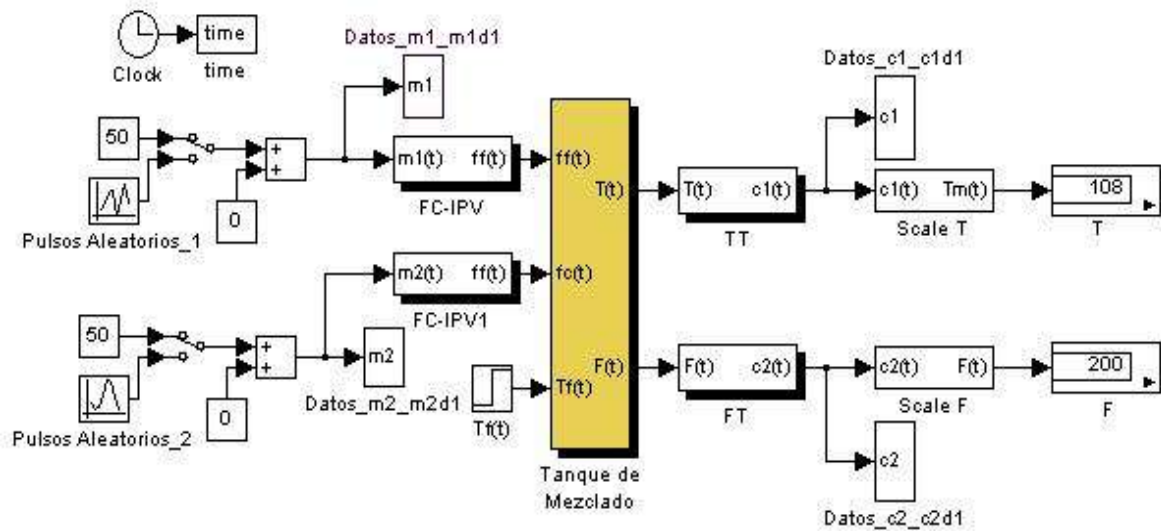


Figura 3.2 Bloques conformados en Lazo\_Abierto\_2x2\_N

En la figura 3.2 se muestra el esquema de los bloques conformado para realizar las simulaciones del proceso en lazo abierto y guardar los datos en Matlab.

El objetivo de este programa es almacenar las siguientes matrices:

$m1d1$ : Primera variable manipulada retrasada en 1 tiempo de muestreo

$m2d1$ : Segunda variable manipulada retrasada en 1 tiempo de muestreo

$c1d1$ : Primera variable controlada retrasada en 1 tiempo de muestreo

$c2d1$ : Segunda variable controlada retrasada en 1 tiempo de muestreo

Con los anteriores datos se conforman las entradas para el entrenamiento de la red, de la siguiente manera:

$$entradas = [md, cd]^T, \text{ donde: } md = [m1d1, m2d1]^T \text{ y } cd = [c1d1, c2d1]^T$$

También se guardan las salidas del proceso sin retardos:

$$salidas = [c1, c2]$$

### 3.3.2 Entrenamiento de la Red Neuronal

El algoritmo llamado **entrenamiento\_1.m** se encarga de hacer el entrenamiento de la red, siguiendo los parámetros teóricos del método de entrenamiento por generación aleatoria de los pesos de activación RAWN (Random Activation Weight Neural Net).

Este algoritmo requiere que se le ingresen los siguientes datos:

*entradas*: Constituyen los vectores de datos que ingresan a la red neuronal (matriz de entrada). El número de entradas se representa en el programa como  $N_i$ .

*salidas*: Es el vector de datos compuesto por las salidas del proceso  $c1$  y  $c2$  este vector es requerido por la red para hacer las predicciones.

Además de se han fijado los siguientes parámetros del algoritmo:

*Neuronas (h)*: Este valor define el número máximo de neuronas con la que se desea evaluar la red. Se trabaja con un número máximo de 8 neuronas o nodos.

*Intentos (j)*: Este valor define el número de pruebas con matrices aleatorias que hace la red para realizar las predicciones con el mínimo SSE posible. Se trabaja con un número máximo de 16 intentos. A mayor número de *intentos* mayor es el tiempo de simulación empleado.

Una vez ingresados estos datos, el programa adiciona el bias a la matriz de entrada (capa de entrada), este bias induce a que la respuesta de la red sea excitatoria o inhibitoria.

Se prosigue con una prueba mediante un bucle FOR, para determinar cual es el numero de neuronas (*h*) óptimo, para obtener la mejor respuesta posible, lo que se traduce como el menor error de predicción. Este proceso se lleva acabo de la siguiente manera:

Se genera una matriz ( $N_{i+1} \times h$ ) con entradas aleatorias tomadas de una distribución normal con media cero y varianza uno ( $R$ ), que incluye la selección de los pesos para el bias. Seguidamente se hace un proceso de regularización de los pesos de la matriz de la capa oculta, para evitar problemas de saturación a causa de entradas muy grandes o evitar que solo opere en el rango lineal cuando las entradas sean muy pequeñas. La matriz de pesos regularizada ( $W$ ) se implementa usando la siguiente expresión:

$$W = \sqrt{\frac{a}{\max \sum_{N_i} \text{entradas}^2}} * R \quad \text{Ecuación 3.3 [19]}$$



En donde el término  $a$  es el límite superior de la función de activación usada en la red. A continuación se calcula la subred ( $Z$ ) de cada neurona, formada por la multiplicación de la matriz regularizada ( $W$ ) por las entradas (*entradas*). Ya teniendo las subredes, se calcula la salida de la capa oculta ( $V$ ) aplicándole a la sumatoria de todas las subredes de cada neurona, la función de activación o función de transferencia de la neurona, que debido al éxito en estudios previos se selecciona la tangente sigmoideal. Lo anterior implica que en la ecuación 3.3  $a$  sea igual a 1.

El próximo paso es adicionar el bias a ( $V$ ) para después calcular los pesos de la capa de salida ( $W_{out}$ ) mediante el método de los mínimos cuadrados. La expresión que se implementa se muestra en la ecuación (3.4):

$$W_{OUT} = (V^T V)^{-1} V^T Y \quad \text{Ecuación 3.4 [19]}$$

Con la obtención de ( $W_{out}$ ), es posible conocer las predicciones de la salida del proceso ( $Y_{pred}$ ), llevando a cabo el producto ( $V$ ) x ( $W_{out}$ ). Conociendo ya las predicciones, se calcula un parámetro de calidad, que no es más que el error de predicción ( $SSE$ ) que se define como la suma del cuadrado de la diferencia entre la salida real ( $Y$ ) y la predicha ( $Y_{pred}$ ). Al obtener el error de predicción mínimo ( $SSE_{min}$ ), se guarda el número de neuronas y la matriz de pesos aleatorios que produjeron dicho error mínimo.

El programa además arroja otros datos como el error para cada predicción, los pesos de la matriz oculta y el número de neuronas óptimo.

El algoritmo guarda una matriz de datos llamada *tasas conformada* por los pesos de la red  $Wh, Bh, Wo, Bo$  y el número óptimo de neuronas (*neuronas*).

### 3.3.3 Algoritmo de Predicción

En el algoritmo ***predictor\_2X2.m*** la red neural hace la predicción de la salida del proceso, tomando como entradas a la red los datos del proceso retrasados en un tiempo de muestreo  $md$  y  $cd$ , donde  $md = [m1d1, m2d1]$  y  $cd = [c1d1, c2d1]$ .

La predicción se lleva a cabo identificando inicialmente las variables globales usadas, para luego cargar los pesos de la red obtenidos con *entrenamiento\_1.m*. A continuación se identifica el número de entradas y neuronas, para luego adicionar el bias a las entradas y a los pesos, lo que permitirá hacer el cálculo de la red para cada neurona (*net*). Una vez hallada la red de cada neurona le aplicamos la función de activación obteniendo así la salida de cada neurona (*outs*), a la cual se le adiciona el bias y se multiplica por los pesos de la capa de salida para conseguir la predicción de la salida (*c\_pred*).

### 3.3.4 Función Objetivo

***Contro\_Setup\_2X2.m*** es el algoritmo que recibe en múltiples ocasiones valores distintos de la salida del controlador, con los cuales, calcula distintos

valores de una función objetivo, con la que se definirá cual es la salida optima del controlador. La función objetivo implementada queda de la siguiente forma:

$$fobj = \lambda * \left[ \frac{1}{2} (n_1 - m_1 d)^2 + \frac{1}{2} (n_2 - m_2 d)^2 \right] + SSE_1 + g * SSE_2 \quad (\text{Ecuación 3.3})$$

Siendo  $\lambda$  el factor denominado de *agresividad*, ya que él es el responsable de cuan violenta es la salida del controlador; y  $g$  es el factor de *ponderación* entre la suma del cuadrado de los errores de las variables controladas.

Este algoritmo llama a la rutina de predicción (*predictor\_2X2.m*), para calcular  $n$  predicciones (*horizonte de predicción*) de la salida del proceso, a las cuales se les suma un factor de desviación  $d$ . Este factor de desviación es el error de predicción. Luego se calcula la suma del cuadrado de los errores de modelaje  $SSE$ . Ya teniendo el  $SSE$  y un posible valor de la salida del controlador, se prosigue a calcular la función objetivo.

El factor de agresividad, el tiempo de muestreo y el número de predicciones son los parámetros que caracterizan la acción del controlador neuronal. Posteriormente se ilustrará el procedimiento seguido para determinar dichos parámetros.

### 3.3.5 Algoritmo De Control

**Control\_Neural\_2X2.m** es el algoritmo que contiene las rutinas principales del controlador neural, por lo que requiere estar integrado al lazo cerrado de

control montado sobre simulink, *sistema\_lazo\_cerrado\_2X2\_N.mdl* (ver figura 3.3). Es por ello que se implementó el algoritmo como una estructura de tipo *S-function* [20] que permite llamar un archivo *\*.m* desde un modelo de Simulink.

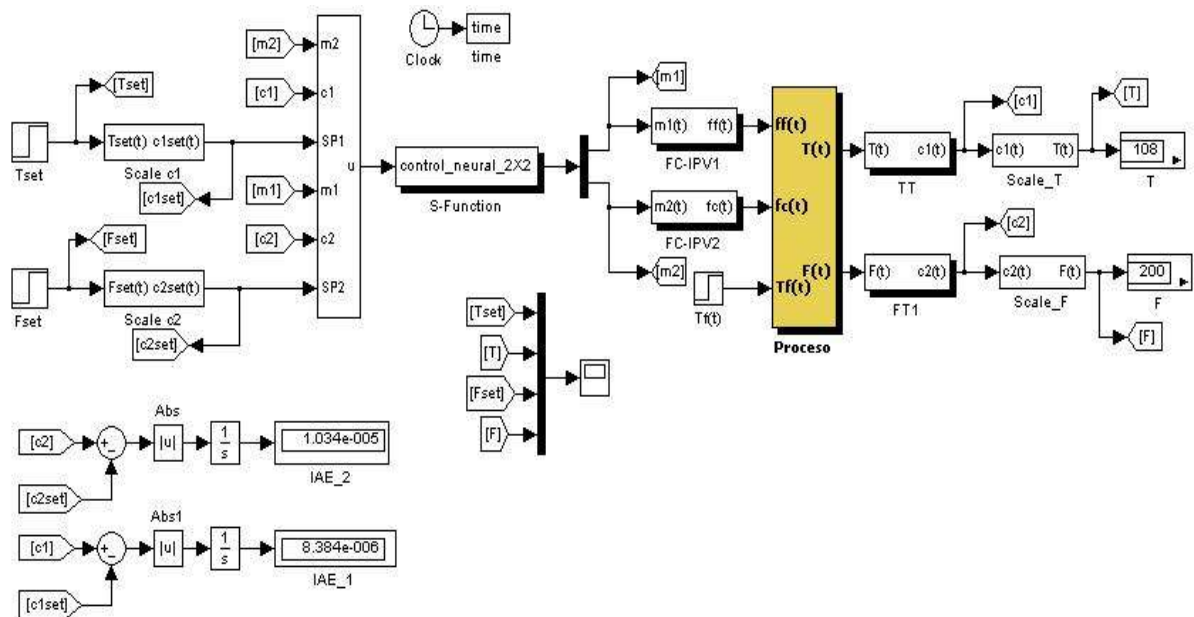


Figura 3.3 Bloques generales de un sistema de control en lazo cerrado implementado con redes neuronales.

Para ingresar los datos requeridos por el algoritmo se hizo un bloque distribuidor de datos el cual genera un vector de datos  $u$  compuesto por las salidas actuales del proceso  $c1$  y  $c2$ , los *set-points* y los valores pasados de las variables manipuladas ( $m1$  y  $m2$ ) y controladas ( $c1$  y  $c2$ ).

El objeto del algoritmo es generar una salida óptima para el controlador  $M = [m1, m2]$  a partir de los datos del vector  $u$ . En el tiempo 0 la salida del controlador es el valor inicial de estado estable de las variables manipuladas

( $m_1$  y  $m_2$ ). De ahí en adelante, el controlador calculará una acción de control cada tiempo  $t$  (tiempo de muestreo).

La rutina de control es llevada a cabo de la siguiente manera:

Con los datos consignados en el vector  $u$ , se hace la predicción de la salida actual del proceso llamando a la rutina de predicción *predictor\_2X2.m*, a continuación se calcula la desviación efectuando la diferencia entre las salidas del proceso y las predichas. El próximo paso es crear un vector de datos *fin*, el cual servirá para alimentar de datos al optimizador. Este vector está compuesto de los valores actuales y pasados de las variables controladas y manipuladas así como de la desviación. Ya teniendo definido este vector, se llama a *control\_setup\_2X2.m* y se encuentra el valor de la matriz  $M$  que minimiza una función, descrita anteriormente como función objetivo, usando el comando *fmincon* del toolbox de optimización de Matlab [21], limitando además la salida del controlador a un rango entre 0 y 100% evitando así el reset windup. El método de optimización de *fmincon* se denomina Sequential Quadratic Programming (SQP) y representa el estado del arte en lo que se refiere a métodos de programación no lineal [21]. Una vez calculadas las salidas del controlador, éstas son enviadas al proceso.

#### **4. IMPLEMENTACIÓN DEL CONTROLADOR NEURONAL Y EVALUACIÓN DEL RENDIMIENTO EN SIMULACIÓN**

En este capítulo se presenta la implementación de los controladores PID y neuronal sobre dos procesos no lineales multivariables 2 x 2 en ambiente Simulink de Matlab. En cada proceso se describe el procedimiento realizado para la selección del mejor apareamiento entre variables manipuladas y controladas y, luego se procede a implementar y comparar el desempeño de los diferentes tipos de control. El primer proceso es un tanque de mezclado en donde se debe controlar la composición y el flujo de la corriente de salida. El segundo proceso es un tanque de calentamiento en el que se requiere controlar la temperatura, con un retraso por transporte, y el flujo de salida.

Debe destacarse que para no extenderse demasiado, se muestran sólo los resultados más significativos que constituyen parte de la gran cantidad de ensayos efectuados, pues se realizaron pruebas con diferentes señales de referencia, para diferentes valores de agresividad y ponderación en el optimizador. También se realizaron pruebas para diferentes razones de aprendizaje y tiempos en la red de identificación.

## 4.1 TANQUE DE MEZCLADO

### 4.1.1 Descripción del Proceso

La figura 4.1 ilustra el esquema del primer proceso usado para realizar el análisis comparativo y la evaluación del rendimiento respecto al controlador neuronal y a los controladores PID.

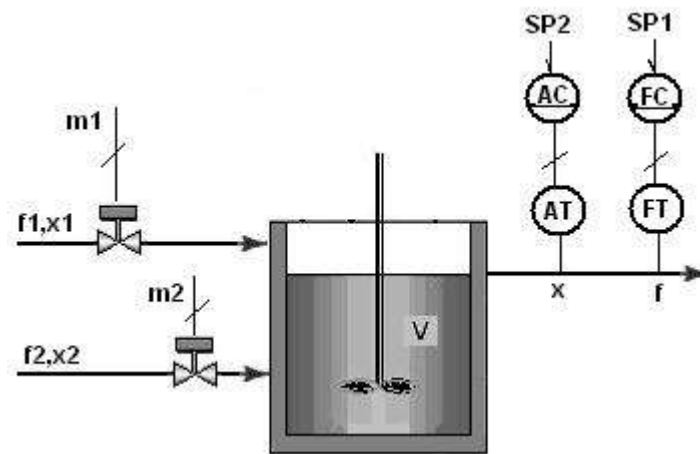


Figura 4.1 Tanque de mezclado. Control de Flujo y Composición. (Tomado de [2])

La corriente de flujo uno  $f_1$  con una composición  $x_1$  se mezcla con la corriente dos  $f_2$  con una composición  $x_2$  en el tanque por medio de agitación continua. Las corrientes de flujo son reguladas por medio de válvulas de control. La composición  $x$  y el flujo a la salida  $f$  del tanque son variables medidas. Las composiciones  $x_1$  y  $x_2$  se consideran constantes.

Las válvulas de control son lineales y diseñadas para un flujo máximo de 150  $lb/min$ . Las constantes de tiempo de las válvulas se consideran despreciables. El transmisor de flujo es lineal con un rango de 0 a 150  $lb/min$  y constante de tiempo despreciable. El analizador transmisor tiene un rango de 5 a 35% en masa y su constante de tiempo es de 5 minutos.

El modelamiento detallado de este proceso se muestra en el anexo B.

#### **4.1.2 Entradas y Salidas del Proceso en Lazo Abierto**

Se consideran como entradas del proceso los flujos que ingresan al tanque  $f_1, f_2$ .

Se consideran como salidas del proceso el flujo de salida del tanque  $f$ , y la composición del flujo de salida  $x$

#### **4.1.3 Definición de Variables Controladas y Manipuladas.**

Las variables controladas y manipuladas para el proceso del tanque de mezclado se definen de la siguiente manera:

Variable controlada uno ( $c_1$ ):  $f$

Variable controlada dos ( $c_2$ ):  $x$

Variabes manipulada uno ( $m_1$ ) :  $f_1$

Variable manipulada dos ( $m_2$ ):  $f_2$



#### 4.1.4 Determinación de las Funciones de Transferencia

Para un sistema 2 x 2 las cuatro combinaciones posibles de variables manipuladas y controladas se modelan cada una como un sistema de primer orden más tiempo muerto (FOPDT).

$$G_{C,M}(s) = \frac{K_{C,M} e^{-t_{0C,M} s}}{\tau_{C,M} s + 1} \quad (\text{Ecuación 4.4}) [2]$$

siendo  $K_{C,M}$  es la ganancia del proceso en estado estacionario,  $t_{0C,M}$  es el tiempo muerto efectivo del proceso y  $\tau_{C,M}$  es la constante de tiempo efectiva del proceso, donde los subíndices  $C$  y  $M$  indican la respectiva variable controlada y manipulada.

Utilizando una prueba escalón y el método de cálculo Fit 3 [2] se obtienen los parámetros de cada proceso. Para los cálculos se realizó un paso escalón, en lazo abierto, a la señal de las válvulas ( $m1$  y  $m2$ ).

Los datos se registraron en la tabla 4.1 y la representación gráfica de la respuesta del proceso se muestra en las figuras 4.2 y 4.3. En la tabla 4.2 se muestra también la nomenclatura usada para designar las variables.

Tabla 4.1

Designación de variables		Cond. de Diseño	Prueba m1-	Prueba m2-
Flujo 1 (f1)	m1 (%CO)	66.667	60	66.667
Flujo 2 (f2)	m2 (%CO)	66.667	66.667	55
Flujo (f)	c1 (%TO)	66.667	73.33	78.33
Concentración (x)	c2 (%TO)	50	47	54.93

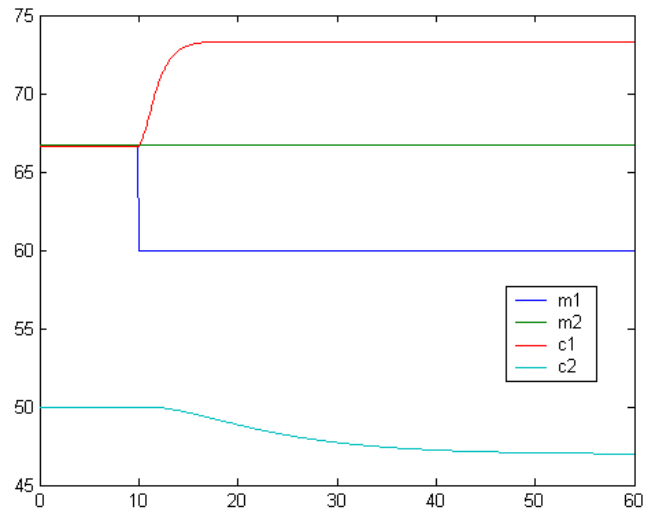


Figura 4.2 Respuesta del proceso ante un cambio escalón en m1

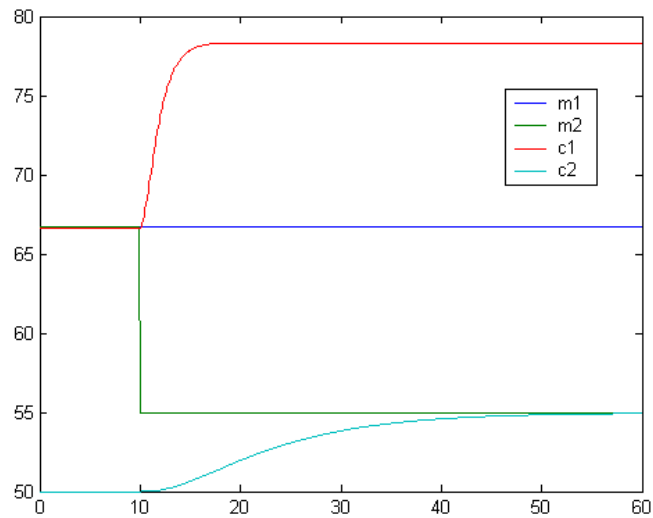


Figura 4.3 Respuesta del proceso ante un cambio escalón en m2

La figura 4.2 muestra que ante el paso escalón en  $f_1$  la variable controlada 1, el flujo de salida del tanque, se estabiliza bastante más rápido que la variable controlada 2, la concentración. Un comportamiento similar se observa en la figura 4.3 donde se representa el efecto sobre las variables controladas ocasionado por el cambio escalón de  $f_2$ .

Luego se determinan las cuatro funciones de transferencia cuyas ganancias de lazo abierto, de acuerdo con las ecuaciones 2.1, y ganancias relativas, ecuaciones 2.2, son:

Tabla 4.2

Ganancias de Lazo abierto		Ganancias Relativas	
K11	-1.00	U11	0.48
K12	-1.00	U12	0.52
K21	0.45	U21	0.52
K22	-0.42	U22	0.48

Las funciones de transferencia son:

$$G_{(1,1)f-f1} = \frac{-1e^{-0.4s}}{1.8s+1} \qquad G_{(1,2)f-f2} = \frac{-1e^{-0.4s}}{1.8s+1}$$

$$G_{(2,1)x-f1} = \frac{0.45e^{-4.1s}}{11.7s+1} \qquad G_{(2,2)x-f2} = \frac{0.7e^{-4.35s}}{10.95s+1}$$

#### 4.1.5 Apareamiento de Variables

Los elementos de la matriz de ganancia relativa, calculados de acuerdo a la ecuación 2.2, se muestran en la tabla 4.3.

Tabla 4.3

	m1	m2
c1	0.48	0.52
c2	0.52	0.48
El apareamiento es x-f1 y f-f2		

De la matriz de ganancias relativas y considerando las reglas de apareamiento de la sección 2.1.3, se establece que la concentración  $x$  se debe controlar manipulando  $f1$  y que el flujo de salida  $f$  se debe controlar manipulando  $f2$ .

#### 4.1.6 Sintonización de los Controladores PID

Utilizando las ecuaciones 2.5 a 2.9 de la sección 2.1.5 calculamos los valores de los parámetros de sintonía de los controladores PID:

Lazo 1:  $f - f2$  Controlando el flujo de salida por manipulación del flujo 2.

$$K_{c1} = -3.8$$

$$T_{i1} = 2$$

$$T_{d1} = 0.2$$

Lazo 2:  $x - f1$  Controlando la composición manipulando el flujo 2.

$$K_{c2} = 3.05$$

$$T_{i2} = 11.85$$

$$T_{d2} = 0.35$$

#### 4.1.7 Implementación de los Desacopladores

Las funciones de transferencia de los dos desacopladores, de acuerdo con las ecuaciones 2.12 y 2.13 son:

$$D_{x-f2} = 0.93 \left( \frac{10.95s + 1}{11.85s + 1} \right) e^{-0.3} \quad \text{y} \quad D_{f-f1} = -1 \left( \frac{1.8s + 1}{1.8s + 1} \right) e^{-0.05}$$

Luego de hallar los parámetros necesarios para implementar el control PID sobre el proceso del tanque de mezclado, en las siguientes secciones se ilustran los pasos requeridos para determinar los parámetros del controlador neuronal.

#### 4.1.8 Entrenamiento de la Red y Ajuste de los Parámetros del Controlador Neuronal.

Como se explicó en la sección 3.2.1 el primer paso para la implementación del controlador neuronal es la recolección de datos para el entrenamiento. Para el proceso del tanque de mezclado se crea en Simulink un modelo del proceso en lazo abierto, el cual se muestra en la figura 4.4. En ésta se identifican los siguientes componentes principales: el proceso con sus respectivas variables de entradas y salidas, la representación de las válvulas y los transmisores, los bloques generadores de pulsos y los recolectores de datos.

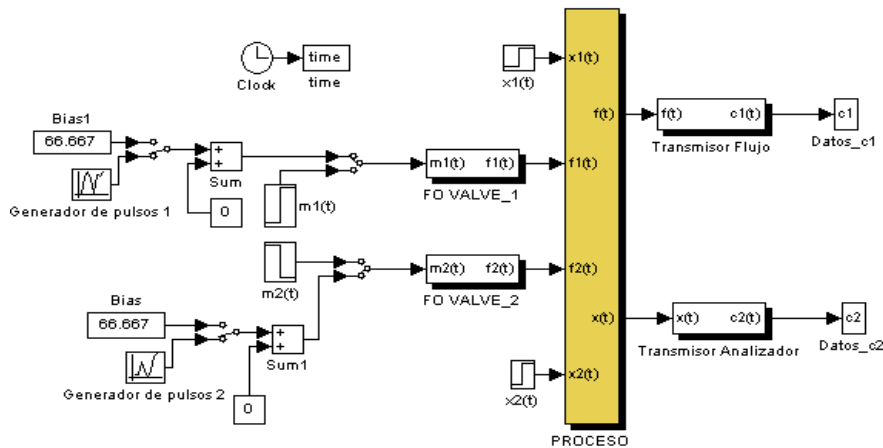


Figura 4.4 Bloques creados en Simulink para la recolección de datos.

Con el conjunto de datos obtenidos del modelo del proceso en lazo abierto se procede a entrenar la red de acuerdo a lo explicado en la sección 3.2.2. Para el proceso del tanque de mezclado, en la figura 4.5 se muestra la calidad de las predicciones hechas por la red al comparar las curvas de las salidas reales del proceso ( $c_1$  y  $c_2$ ) con las salidas predichas por la red ( $c_1\text{ pred}$  y  $c_2\text{ pred}$ ).

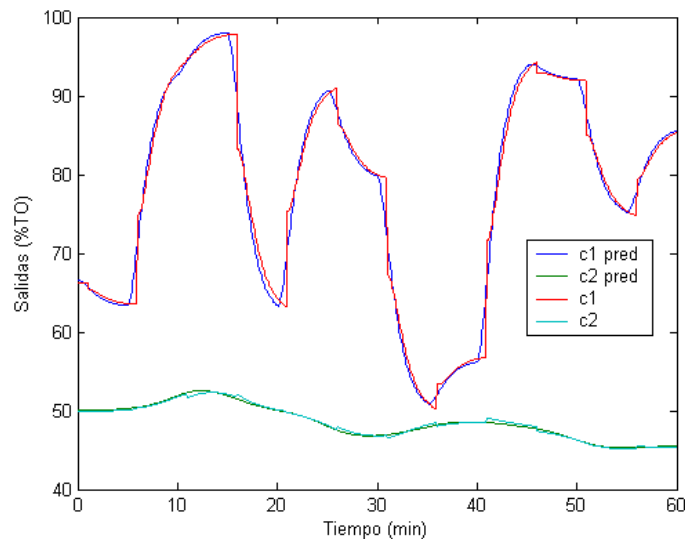


Figura 4.5 Comparación de las salidas reales del proceso con las salidas predichas por la red.

Se observa en la figura 4.5 que las curvas son muy similares, lo que indica que la red neuronal es un buen modelo del proceso y además, este hecho confirma la efectividad del método RAWN usado para el entrenamiento.

Luego del entrenamiento se procede a determinar los parámetros del controlador neuronal, que como se mencionó en la sección 3.1, son las variables que caracterizan su funcionamiento. Dichos parámetros son: factor de agresividad, tiempo de muestreo y número de predicciones. Para lograr el ajuste óptimo se realizan varios ejercicios de control, de tal manera que se comparan las respuestas del proceso, usando el controlador neuronal, ante cambios tipo escalón en los puntos de control de las variables controladas.

#### **4.1.8.1 Determinación del Factor de Agresividad**

Recordemos de la Sección 3.1.3 que el factor de agresividad es el que permite regular la agresividad del controlador neuronal para prevenir cambios súbitos que vayan en detrimento de la válvula u otro elemento final de control. Para determinar el factor de agresividad se efectúa un cambio en el set-point del flujo,  $f$ , en  $t=10$ , y se mantiene el valor del punto de control de la composición,  $x$ , en su valor de estado estable (20%). En la figura 4.6 se muestra las diferentes formas en que la variable controlada  $x$  responde, ante dicho cambio, para distintos valores de agresividad ( $\lambda$ ) en el controlador.

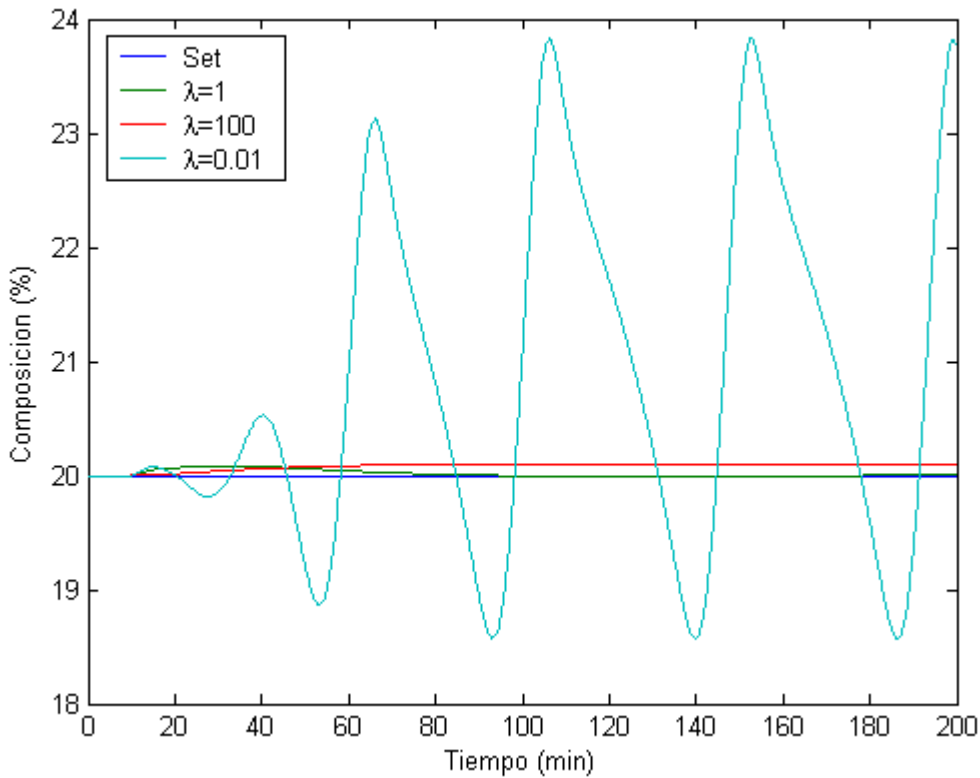


Figura 4.6 Comportamiento de la variable de salida  $x$ , para diferentes valores de agresividad en el controlador.

En la anterior figura se evidencia que la peor respuesta se obtiene con un valor de agresividad bajo,  $\lambda=0.01$ . Para  $\lambda=100$  se observa que el valor de la composición no regresa a su valor de set-point, aunque en este caso no se presenta oscilación. El mejor comportamiento se obtiene para un valor de agresividad igual a la unidad,  $\lambda=1$ .

#### 4.1.8.2 Determinación Del Tiempo De Muestreo

El tiempo de muestreo (TM) del controlador define cada cuanto tiempo el controlador debe calcular su salida óptima. En la figura 4.7 se muestran los



distintos tipos de respuesta, con diferentes tiempo de muestreo, de la variable controlada flujo ( $f$ ) cuando se presenta un cambio en su set-point de 100 a 90

$lb/min$

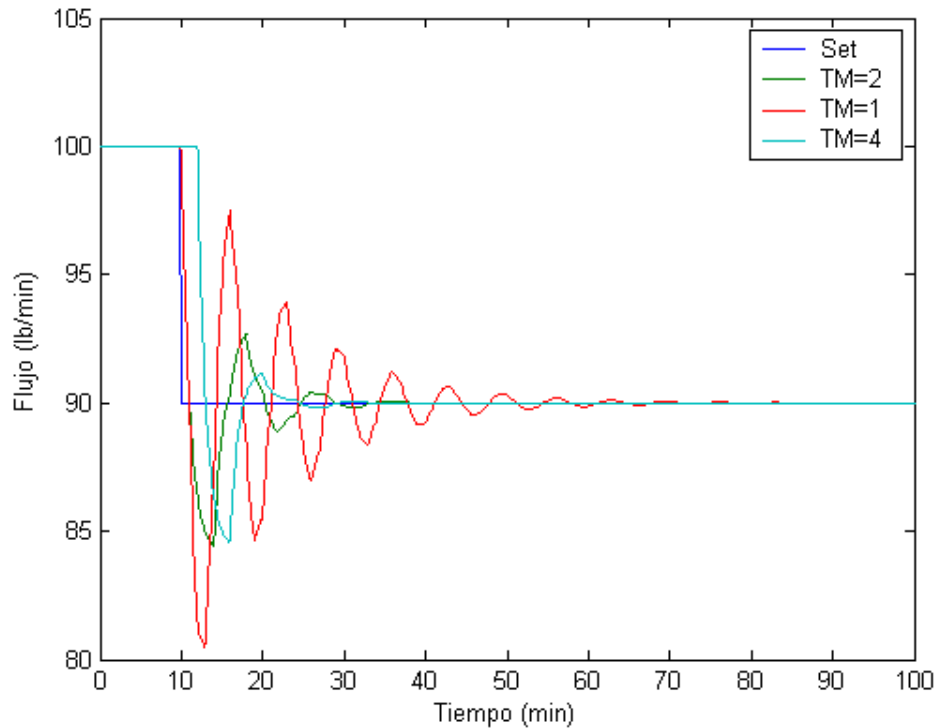


Figura 4.7 Comportamiento de la variable de salida  $f$ , para diferentes tiempos de muestreo en el controlador

Se observa que para un tiempo de muestreo igual a uno,  $TM=1$ , el flujo presenta un comportamiento oscilatorio y agresivo que sólo logra estabilizarse en un tiempo cercano a 80 minutos y presentando la mayor desviación (aproximadamente 10  $lb/min$ ) respecto al valor requerido. A medida que aumenta el tiempo de muestreo,  $TM=4$ , la salida del proceso pierde velocidad de reacción ante el cambio realizado. Para equilibrar los aspectos anteriores, menos agresividad sin oscilaciones fuertes ni prolongadas y buena velocidad de reacción, se opta por escoger un valor de tiempo de muestreo igual a dos,  $TM=2$ .

### 4.1.8.3 Determinación del Número de Predicciones

Como se mencionó en la sección 3.1.3 el controlador debe calcular  $n$  predicciones (horizonte de predicción) de la salida del proceso. Similar a lo hecho para determinar la agresividad y el tiempo de muestro, se procede a realizar un ejercicio de control y comparar los diversos tipos de respuestas de una de las variables controladas. En este caso se induce un cambio el set-point de la composición en  $t=200$  min, de 20 a 18%. En la figura 4.8 y se muestra el comportamiento de esta variable controlada.

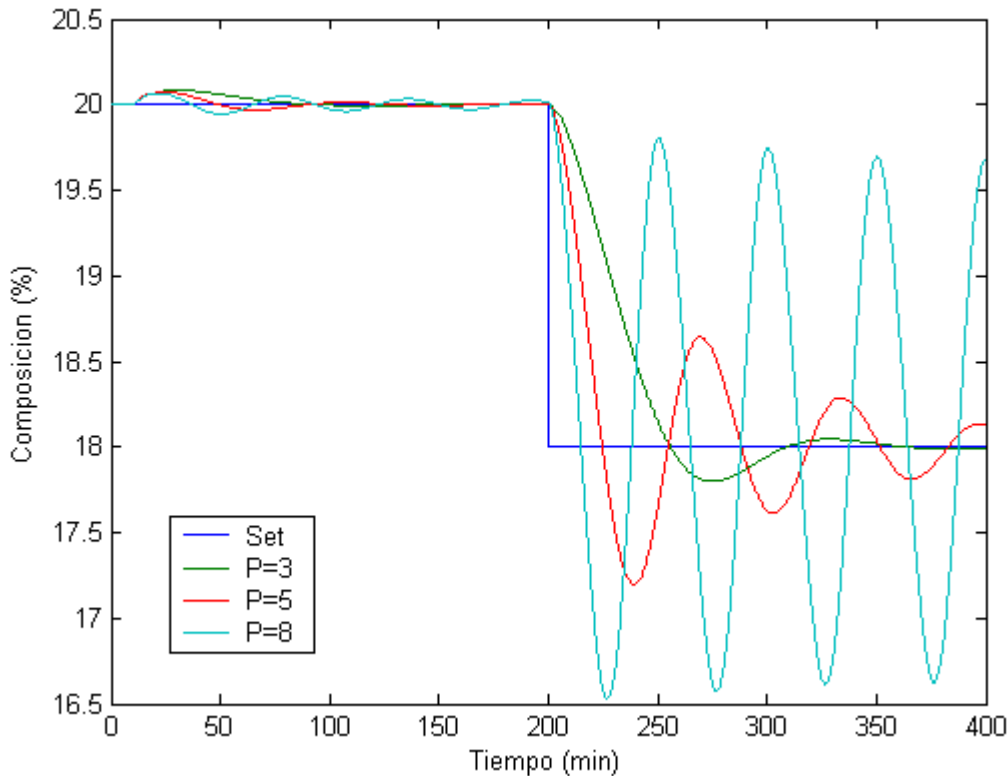


Figura 4.8 Comportamiento de la variable de salida  $x$ , para diferentes números de predicciones en el controlador

De la figura 4.8 se concluye que con tres predicciones,  $P=3$ , la variable de salida analizada alcanza óptimamente su nuevo punto de control y logra estabilizarse. A medida que se tiene un número mayor de predicciones,  $P=5$  y  $P=8$ , la respuesta del proceso es más oscilatoria y agresiva, además de que para estos casos no logra estabilizarse convenientemente. Debido a lo anterior se elige trabajar el controlador neuronal con un número de predicciones igual a tres.

## **4.2 PRUEBAS Y COMPARACIÓN DEL DESEMPEÑO SOBRE EL TANQUE DE MEZCLADO**

Ahora se procede a evaluar el desempeño del controlador neuronal, comparando su acción respecto a la ejercida por los controladores PID con desacopladores y sin desacopladores. Para evaluar el desempeño de cada uno de los controladores, es necesario tener datos cuantitativos para su comparación. Se han escogido tres parámetros comparativos para representar las ventajas y desventajas de los respectivos tipos de control. El primer parámetro utilizado corresponde al tiempo de establecimiento, obtenido por medio de las respuestas gráficas. El segundo parámetro es el índice de suavidad, que se evalúa tanto en la salida del proceso como en la señal de control, el cual permite conocer que tan bruscos son los cambios realizados durante el esfuerzo de control. El tercer parámetro corresponde a la integral del valor absoluto del error (IAE), el cual proporciona el área bajo las curvas entre el valor deseado y el valor obtenido en la salida del proceso. También se cometan ciertos criterios cualitativos como el efecto de la

presencia de grandes picos y fuertes oscilaciones sobre las válvulas y en las salidas del proceso.

Para comparar el desempeño de los diferentes tipos de control se realizaron dos pruebas de la siguiente manera: La primera prueba consiste en forzar un cambio escalón en el punto de control de la composición de la corriente de salida ( $x$ ), el cual cambia de 20% a 18% en  $t = 10$ , manteniendo fija la otra variable controlada, (el flujo  $f$ ), en su valor de diseño. La segunda prueba es cambiar el punto de control del flujo de 200 lb/min a 180 lb/min en  $t = 10$  y mantener el punto de control de la composición fijo.

En general las acciones de control se representan en las gráficas de la siguiente manera:

*Variable Set*: Es el punto de control o set-point de la variable respectiva

*Variable NN*: Comportamiento de la variable con el controlador neuronal

*Variable PID*: Comportamiento de la variable con los controladores PID sin desacoplar

*Variable PID-D*: Comportamiento de la variable con los controladores PID desacoplados

#### 4.2.1 Prueba 1. Cambio Escalón en el Set-point de Composición.

Se induce un cambio escalón en el set-point de la composición de la corriente de salida ( $x$ ), el cual cambia de 20% a 18% en  $t=10$ , manteniendo fija la otra variable controlada, (el flujo  $f$ ), en su valor de estado estable.

##### 4.2.1.1 Comparación de las Salidas del Proceso.

La respuesta del proceso ante el cambio realizado se muestra en las figuras 4.9 y 4.10. En éstas se superponen la acción del controlador neuronal (NN) y la acción de los controladores PID desacoplados (PID-D) y no desacoplados (PID) respecto al valor del punto de control de cada variable.

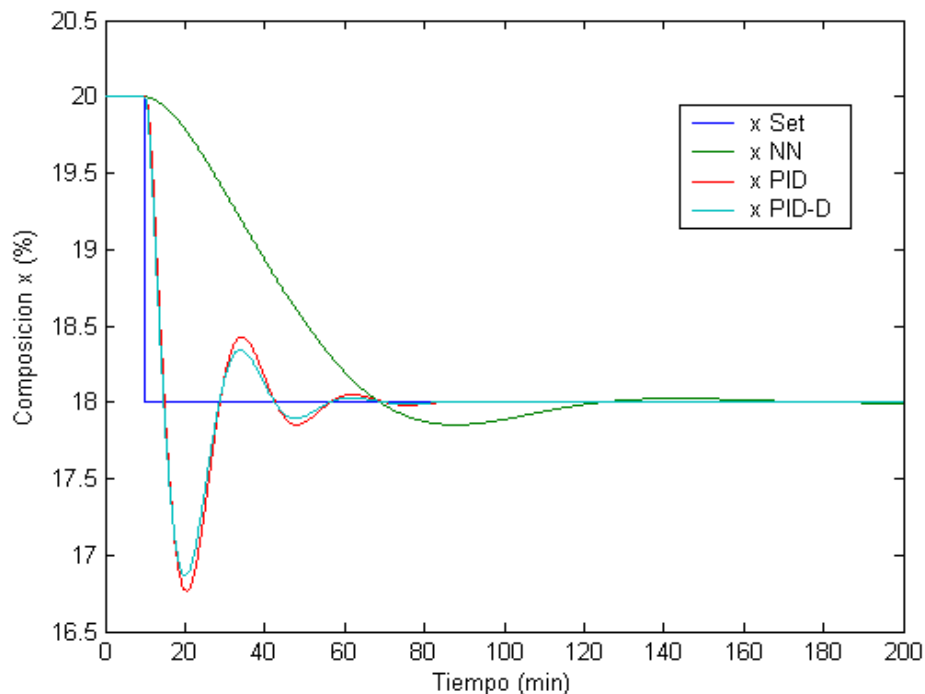


Figura 4.9 Respuesta de la variable de salida Composición ( $x$ ) ante un cambio escalón en su punto de control.

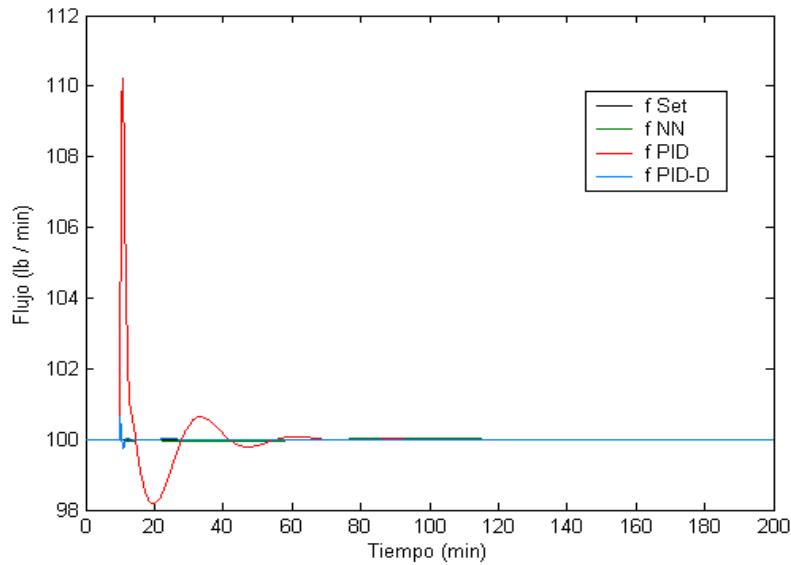


Figura 4.10 Respuesta de la variable de salida Flujo ( $f$ ) ante un cambio escalón en el punto de control de Composición.

En la gráfica 4.11 se ha omitido el control PID con el objeto de comparar sólo la acción de los controladores neuronal y PID desacoplado.

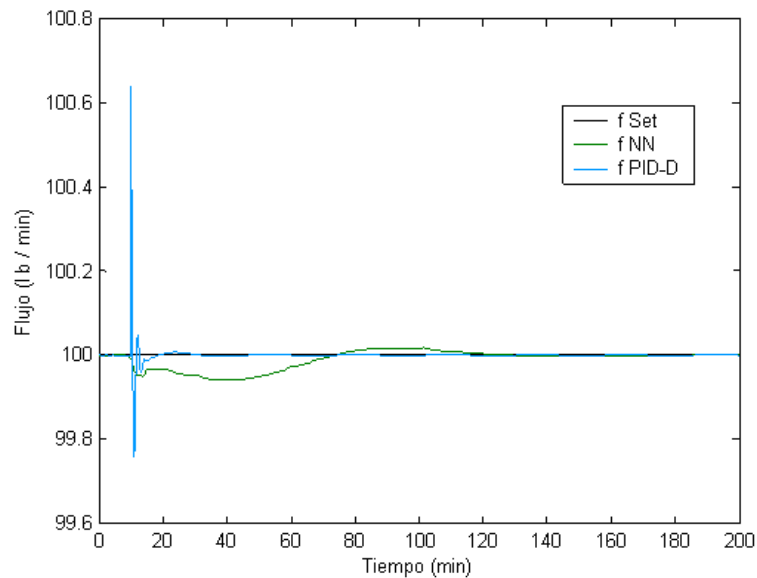


Figura 4.11 Respuesta de la variable de salida Flujo ( $f$ ) ante un cambio escalón en el set-point de Composición. Se omite el control PID.

En general los tres métodos de control son eficaces, sin embargo es notorio que el controlador neuronal tiene una respuesta más suave ante el cambio solicitado, lo que conlleva a proporcionar señales más adecuadas al elemento final de control. Además, en el instante en que se presenta el cambio en la composición se produce un pico alto en la variable flujo, tanto en el proceso controlado por los PID desacoplados como no desacoplados. La estabilización de las señales es similar, sin embargo, el control PID-D y PID presentan un *overshoot* (sobre impulso) no deseado antes de estabilizarse.

La composición, en el caso del control PID-D, llega a un pico de 16.7%, desviándose 1.3 unidades del valor solicitado (18%), mientras que en el control neuronal no hay pico y el valor más desviado del nuevo set-point es de 17.85%, presentando una desviación de tan sólo 0.15 unidades respecto del valor requerido. Algo similar sucede con el flujo, el cual tiene una desviación de 2 unidades respecto al set-point de 100 *lb/min* llegando por encima a 101.4 *lb/min* y por debajo a 99.4 *lb/min* con los controladores PID-D, a 110 *lb/min* por encima, desviación de 10 *lb/min*, y a 98.1 *lb/min* por debajo, desviación de 1.9 *lb/min*, con los PID, en tanto que con el controlador neuronal la desviación máxima es de sólo 0.1 unidades.

#### 4.2.1.2 Comparación de la Salida de los Controladores.

En las gráficas 4.12 y 4.13 se representan las señales de salida de los controladores (%CO Controller Output) neuronal (NN), PID sin desacoplar (PID) y PID desacoplado (PID-D) que llegan a los elementos finales de control (las válvulas).

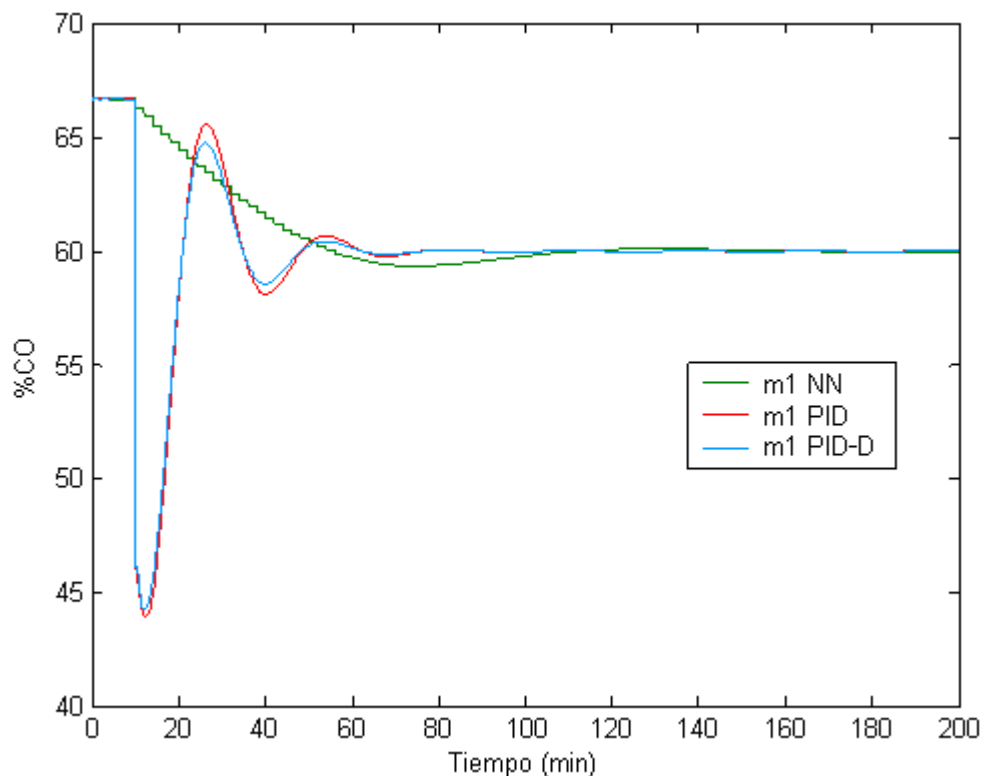


Figura 4.12 Respuestas de los controladores NN, PID y PID-D para la variable manipulada uno (m1).

Se observa en estas gráficas que las salidas de los controladores PID y PID-D presentan grandes picos y oscilaciones antes de lograr estabilizarse, a diferencia del controlador con redes neuronales, que además de proporcionar una respuesta muy eficaz, no presentan oscilaciones y logran una estabilización más suave.



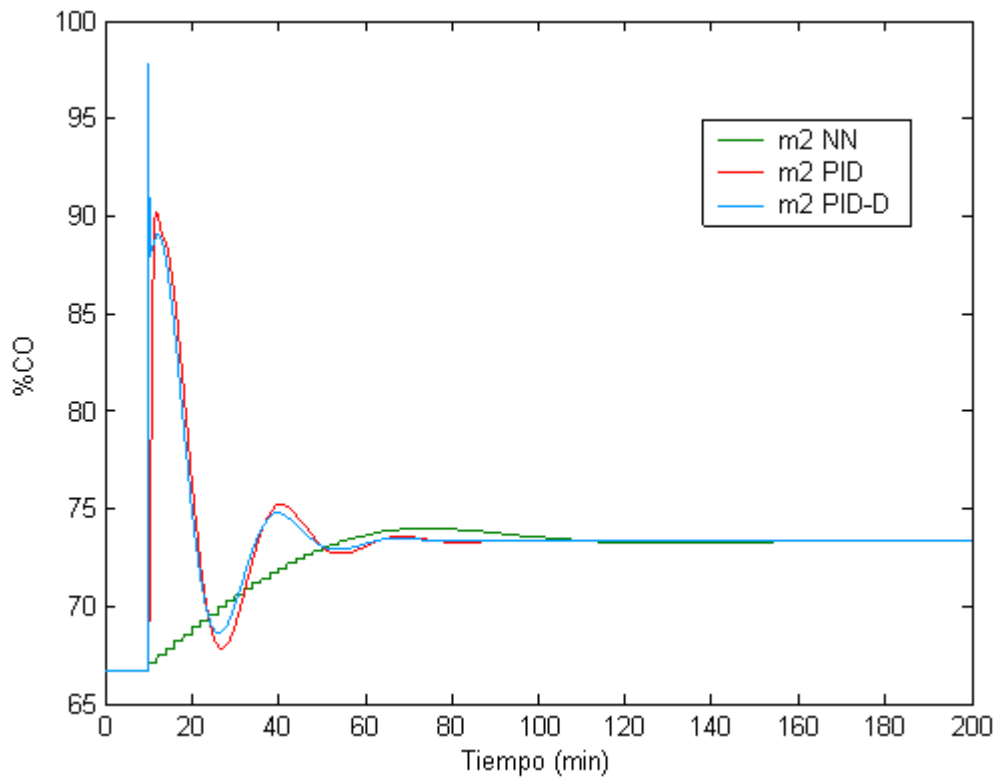


Figura 4.13 Respuestas de los controladores NN, PID y PID D para la variable manipulada dos (m2).

El resultado derivado de analizar las salidas de los diferentes controladores corrobora a los obtenidos cuando se consideraron y compararon las distintas respuestas de las variables de salida del proceso ante el cambio realizado: el control neuronal es mucho más suave que los otros dos.

Los resultados respecto a los criterios de comparación se muestran en la siguiente tabla:

Tabla 4.4

<b>PRUEBA 1 (Tanque de mezclado)</b>			
<b>IAE</b>			
<b>PRUEBA 1</b>	<b>TIPO DE CONTROL</b>		
	<b>PID</b>	<b>PID-D</b>	<b>NN</b>
<b>Lazo 1</b>	26	0.4	2.09
<b>Lazo 2</b>	70	64	244

<b>INDICE DE SUAVIDAD (Salida del proceso)</b>			
<b>PRUEBA 1</b>	<b>TIPO DE CONTROL</b>		
	<b>PID</b>	<b>PID-D</b>	<b>NN</b>
<b>Lazo 1 (%)</b>	1,3	1,2	0,2
<b>Lazo 2 (lb/min)</b>	10,3	0,7	0,05

<b>INDICE DE SUAVIDAD (Salida del controlador)</b>			
<b>PRUEBA 1</b>	<b>TIPO DE CONTROL</b>		
	<b>PID</b>	<b>PID-D</b>	<b>NN</b>
<b>m1 (%CO)</b>	17	16	1
<b>m2 (%CO)</b>	17	25	1

<b>TIEMPO DE ESTABLECIMIENTO (min)</b>			
<b>PRUEBA 1</b>	<b>TIPO DE CONTROL</b>		
	<b>PID</b>	<b>PID-D</b>	<b>NN</b>
<b>Lazo 1</b>	85	84	125
<b>Lazo 2</b>	70	35	60

Se verifica que para ambos lazos de control, el de flujo (Lazo 1) y el de composición (Lazo 2) el IAE del control PID-D es menor que el logrado por el control neuronal, esto es como consecuencia de un control más agresivo (pero no mejor) de los controladores PID-D. También resulta que el control neuronal presenta menores valores de IAE en el lazo de flujo que el control PID.

## 4.2.2 Prueba 2. Cambio Escalón en el Set-point de Flujo

La segunda prueba es cambiar el punto de control del flujo de 200 *lb/min* a 180 *lb/min* en  $t = 10$  y mantener el set-point de la composición fijo.

### 4.2.2.1 Comparación de las Salidas del Proceso.

De igual forma que en la prueba anterior se mostrarán los resultados obtenidos con el control neuronal (NN), el control PID con desacopladores (PID-D) y , además, los obtenidos usando un sistema de control PID sin la acción de los desacopladores (PID).

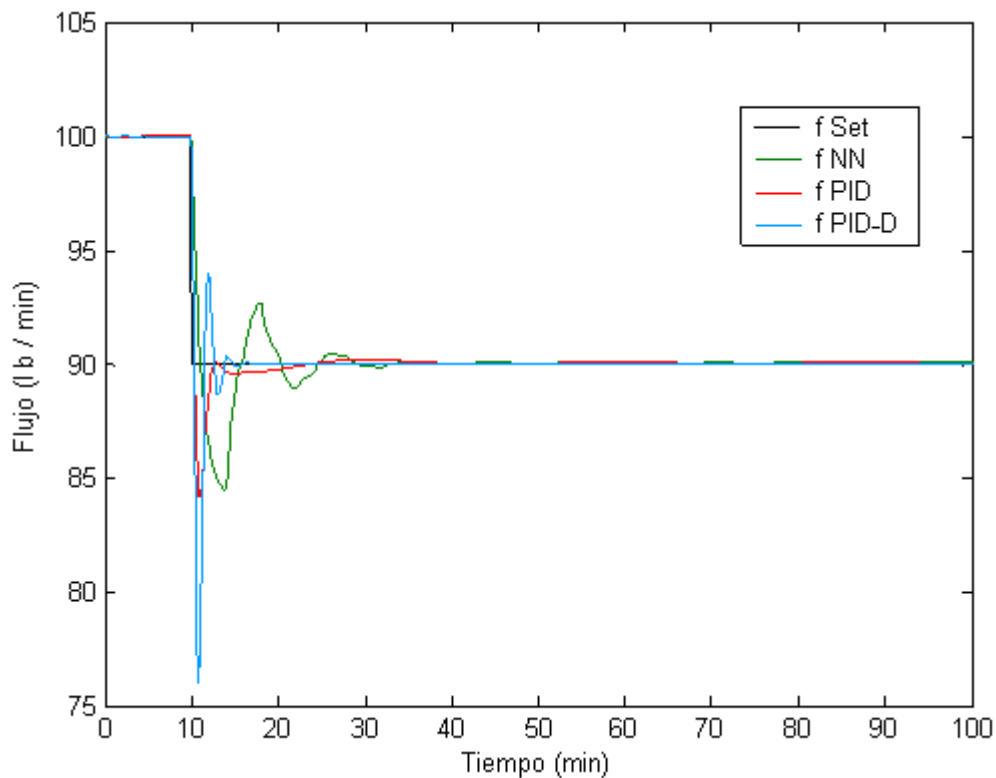


Figura 4.14 Respuesta de la variable de salida Flujo ( $F$ ) ante un cambio escalón en su set-point.

De la gráfica 4.14 se ve que la señal de flujo que sale del proceso oscila antes de alcanzar estabilizarse y que este comportamiento se presenta en las tres formas de control usadas. Sorprende que con el sistema desacoplado esta variable presente el mayor pico llegando a un valor de  $76 \text{ lb/min}$ , es decir una desviación de  $14 \text{ lb/min}$ . Con los otros dos tipos de control se presentan una desviación de tan sólo 4 unidades, control neuronal, y de 5 unidades, control PID sin desacoplar, respecto al nuevo set-point de  $90 \text{ lb/min}$ . Sin embargo, si se observa el tiempo de estabilización de la variable en su nuevo punto de control, se nota que con el control con PID desacoplado se alcanza dicho objetivo en el menor tiempo (18 min), seguido del control neuronal (31 min) y luego del control PID sin desacoplar (39 min).

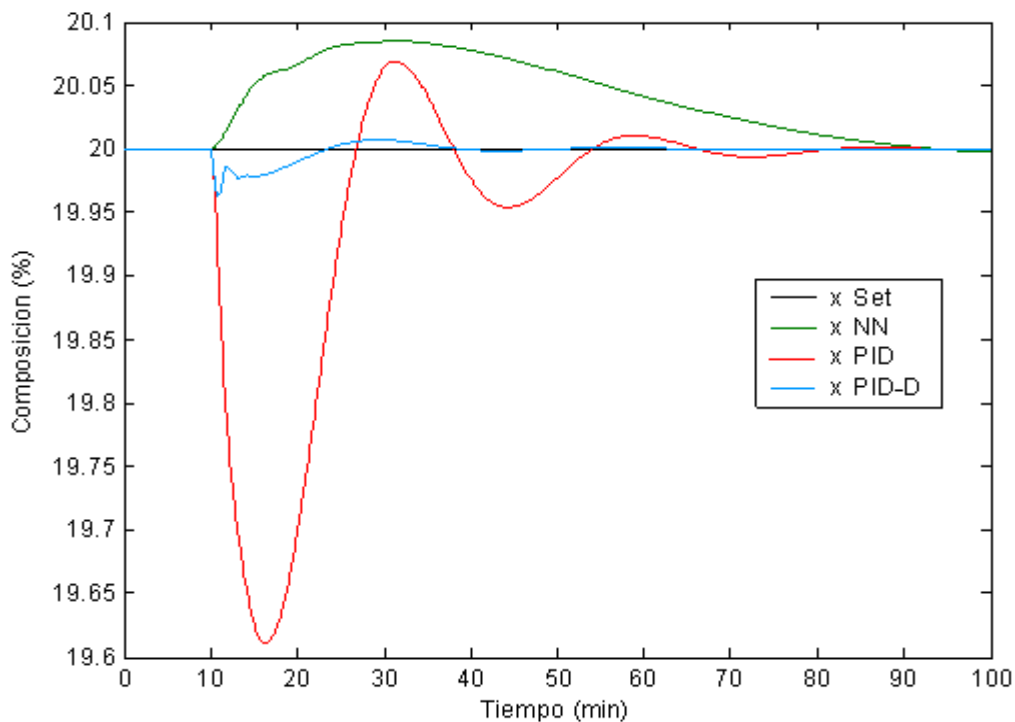


Figura 4.15 Respuesta de la variable de salida Composición ( $x$ ) ante un cambio escalón en el set-point de Flujo.

Se puede ver en la figura 4.15 que por medio de los tres métodos de control se logra regresar la variable composición a su set-point, sin embargo es notorio que el controlador neuronal tiene una respuesta más suave, sin picos y sin alejarse tanto, en comparación con la respuesta del PID, del valor del set-point. Además, en el instante en que se presenta el cambio en el flujo hay una notoria oscilación en la salida de composición del proceso que es controlado por los PID sin desacoplar. En cuanto a la estabilización de las señales se observa que se realiza en tiempos similares con el control NN y con el PID-D, sin embargo, el control PID presenta *overshoot*. En este último aspecto es notable la superioridad de la acción del control PID con desacopladores.

La composición, en el caso del control PID, llega a un pico de 19.62%, desviándose 0.38 unidades del valor deseado (20%), mientras que con el control NN no hay pico y el valor más desviado del nuevo punto de control es de 20.15%, presentando una desviación de tan sólo 0.15 unidades respecto del valor requerido. Cuando se aplican los desacopladores la desviación se reduce significativamente de 0.4 a 0.05 unidades.

#### **4.2.2.2 Comparación de la Salida de los Controladores.**

En las gráficas 4.16 y 4.17 se encuentran superpuestas las señales de salida de los controladores neuronal NN y PID desacoplado que llegan a las válvulas.

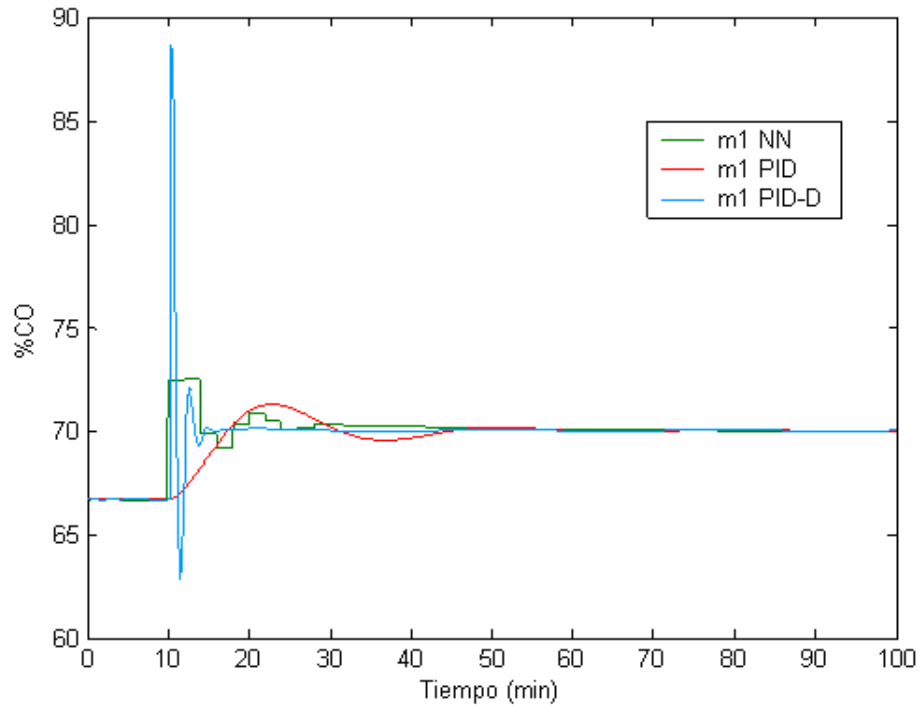


Figura 4.16 Respuestas de los controladores NN, PID y PID-D para la variable manipulada uno (m1).

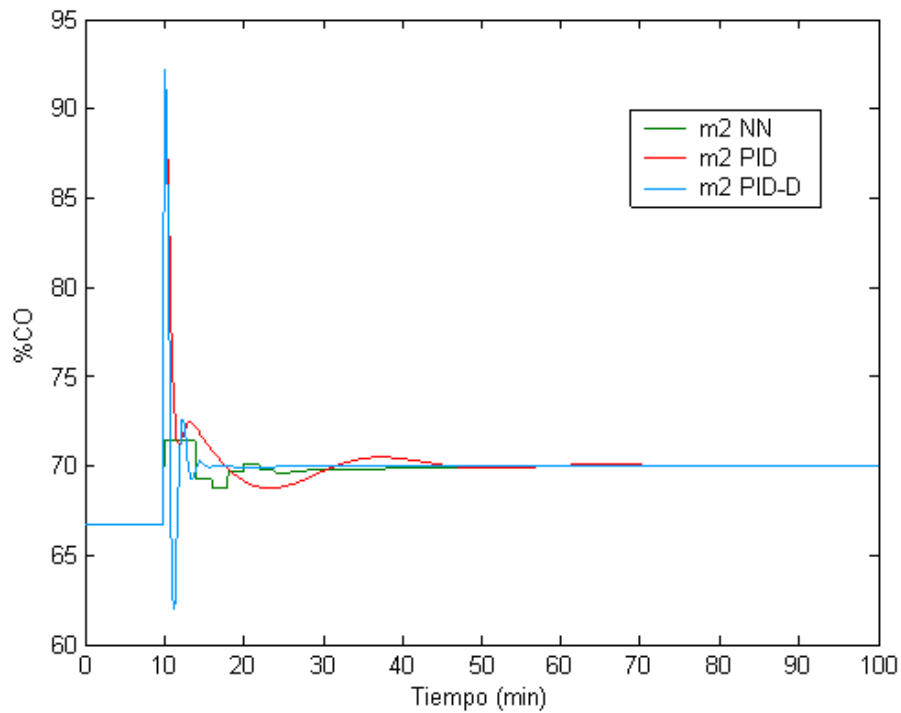


Figura 4.17 Respuestas de los controladores NN, PID y PID-D para la variable manipulada dos (m2).

En las figuras anteriores se manifiesta que nuevamente la salidas del controladores PID presenta grandes picos y oscilaciones antes de lograr estabilizarse, a diferencia de los controladores con redes neuronales, que además de proporcionar una buena acción de control, no presentan oscilaciones y logran una estabilización más suave y más rápida. Los resultados respecto a lo criterios comparativos se resumen en la siguiente tabla:

Tabla 4.5

<b>PRUEBA 2 (Tanque de mezclado)</b>			
<b>IAE</b>			
<b>PRUEBA 2</b>	<b>TIPO DE CONTROL</b>		
	<b>PID</b>	<b>PID-D</b>	<b>NN</b>
<b>Lazo 1</b>	9,4	6,2	21,3
<b>Lazo 2</b>	14	0,92	12,6

<b>INDICE DE SUAVIDAD (Salida del proceso)</b>			
<b>PRUEBA 2</b>	<b>TIPO DE CONTROL</b>		
	<b>PID</b>	<b>PID-D</b>	<b>NN</b>
<b>Lazo 1 (%)</b>	0,39	0,04	0,08
<b>Lazo 2 (lb/min)</b>	6	14	7

<b>INDICE DE SUAVIDAD (Salida del controlador)</b>			
<b>PRUEBA 2</b>	<b>TIPO DE CONTROL</b>		
	<b>PID</b>	<b>PID-D</b>	<b>NN</b>
<b>m1 (%CO)</b>	2	19	3
<b>m2 (%CO)</b>	17	6	2

<b>TIEMPO DE ESTABLECIMIENTO (min)</b>			
<b>PRUEBA 2</b>	<b>TIPO DE CONTROL</b>		
	<b>PID</b>	<b>PID-D</b>	<b>NN</b>
<b>Lazo 1</b>	90	40	90
<b>Lazo 2</b>	40	18	35

Se verifica nuevamente que con el control PID-D se presentan los menores valores de IAE. El control NN presenta valores menores de IAE respecto a los proporcionados con el control PID.

## 4.3 TANQUE DE CALENTAMIENTO

### 4.3.1 Descripción del Proceso

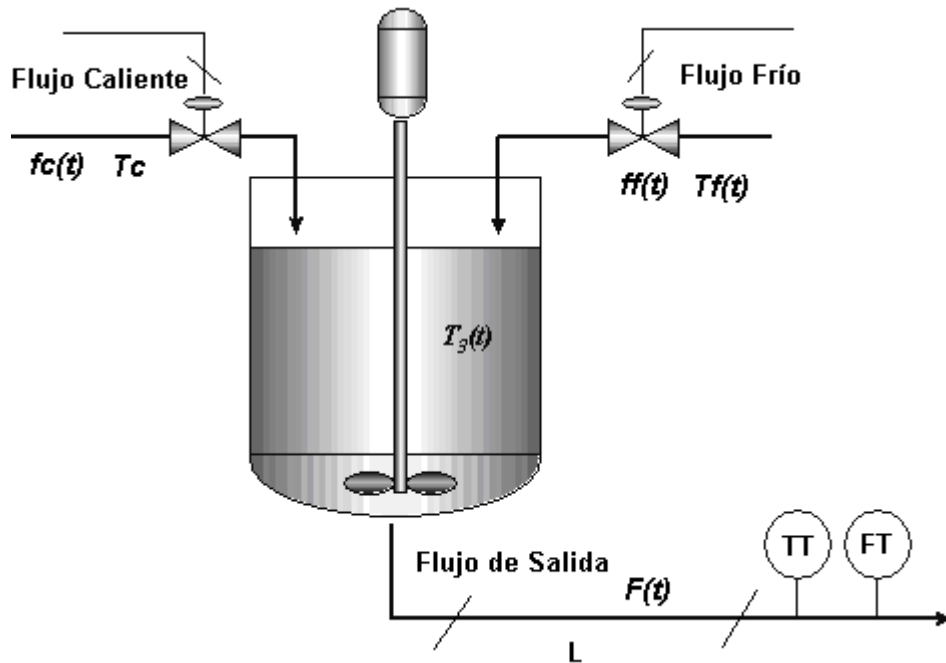


Figura 4.18 Tanque de calentamiento. (Tomado de [2])

La figura 4.18 muestra el proceso en el tanque de calentamiento. La corriente de flujo caliente ( $fc$ ) se mezcla con la corriente de flujo frío ( $ff$ ) en el tanque por medio de agitación continua. En este proceso se debe controlar la temperatura ( $T$ ) y el flujo de la corriente de salida ( $F$ ). La temperatura de salida se mide en un punto 120 pies en sentido descendiente del tanque. La temperatura del flujo caliente ( $T_c$ ) se mantiene constante y la temperatura del flujo frío ( $T_f$ ) se considera como una perturbación.

El modelamiento detallado de este proceso se muestra en el Anexo C.



### 4.3.2 Entradas y Salidas en Lazo Abierto

Entradas:  $f_c, ff, T_f$

Salidas:  $F, T_F$

### 4.3.3 Definición de Variables Controladas y Manipuladas.

Variables Controladas: Temperatura del flujo de salida  $T_F (c_1)$  y flujo de salida  $F (c_2)$ .

Variables Manipuladas: Flujo caliente  $f_c (m_1)$  y flujo frío  $ff (m_2)$

### 4.3.4 Funciones de Transferencia

En lazo abierto se realiza un paso escalón a cada una de las variables manipuladas y se registraron los datos en la tabla 4.6.

Tabla 4.6

Designación de variables		Cond. De Diseño	Prueba m1+	Prueba m2+
Flujo frío ff	m1 (%CO)	50	55	50
Flujo caliente fc	m2 (%CO)	50	50	55
Temperatura T	c1 (%TO)	45,8824	43,205	48,64
Flujo de salida F	c2 (%TO)	50	55,239	53,49

La representación gráfica de la respuesta del proceso se muestra en las figuras 4.19 y 4.20.

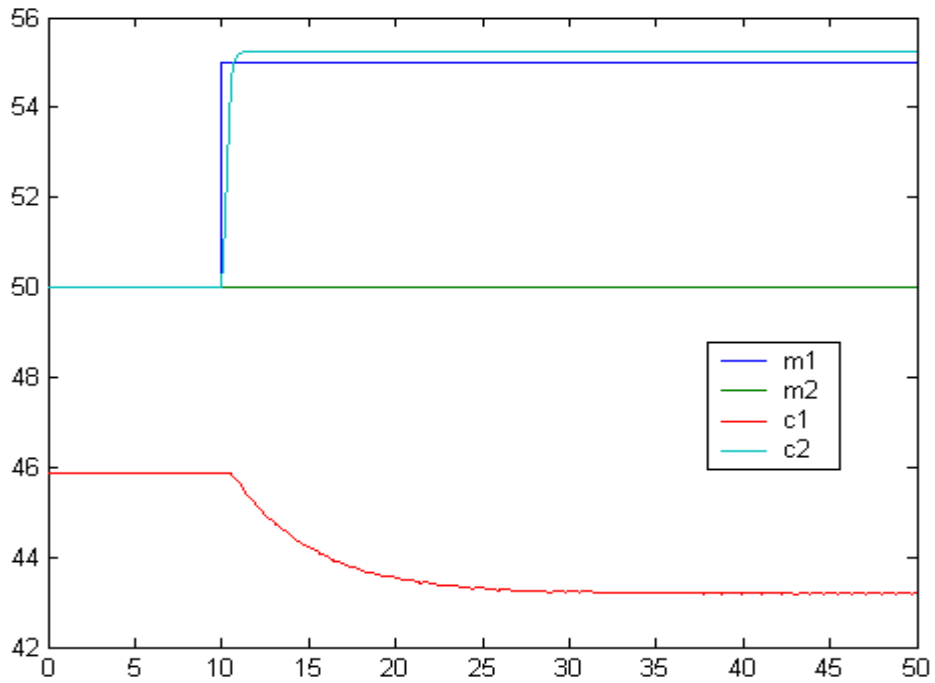


Figura 4.19 Respuesta del proceso ante un cambio escalón en m1

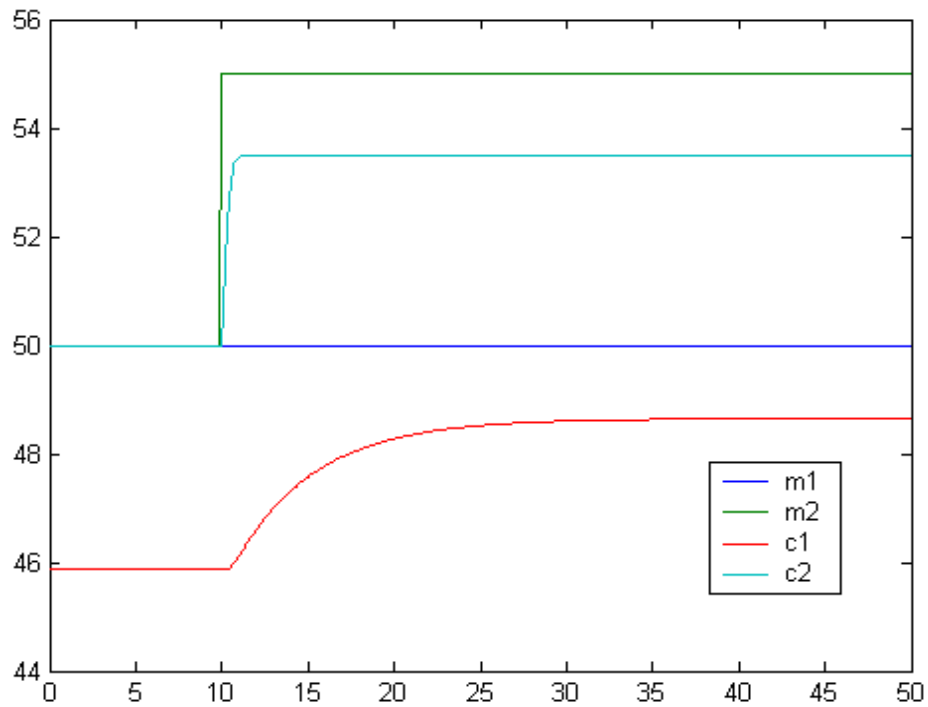


Figura 4.20 Respuesta del proceso ante un cambio escalón en m2

Se determinan las cuatro funciones de transferencia, cuyas ganancias de lazo abierto son:

Tabla 4.7

Ganancias de Lazo abierto		Ganancias Relativas	
K11	-0,54	U11	0,39
K12	0,55	U12	0,61
K21	1,05	U21	0,61
K22	0,70	U22	0,39

Las funciones de transferencia de las cuatro combinaciones de lazos de control son:

$$G_{(1,1)T-ff} = \frac{-0.54e^{-0.6s}}{4.5s + 1} ; G_{(1,2)T-fc} = \frac{0.55e^{-0.55s}}{4.7s + 1}$$

$$G_{(2,1)F-ff} = \frac{1.05e^{-0.6s}}{0.24s + 1} ; G_{(2,2)F-fc} = \frac{0.7e^{-0.55s}}{0.25s + 1}$$

#### 4.3.5 Apareamiento de Variables

Se calcula la matriz de ganancia relativa:

	<i>m1</i>	<i>m2</i>
<i>c1</i>	0,4	0,6
<i>c2</i>	0,6	0,4

El apareamiento es  $F - ff$  y  $T - fc$

De la matriz de ganancias relativas se observa que la temperatura  $T$  se debe controlar manipulando el flujo caliente  $fc$  y que el flujo de salida  $F$  se debe controlar manipulando el flujo frío  $ff$ .

#### 4.3.6 Sintonización de Controladores PID

Utilizando las ecuaciones 2.5 a 2.9 de la sección 2.1.5 calculamos los valores de los parámetros a sintonizar:

Lazo 1:  $T - fc$

$$K_{C1} = 4 ; T_{I1} = 5 ; T_{D1} = 0.275$$

Lazo 2:  $F - ff$

$$K_{C2} = 1.5 ; T_{I2} = 0.24 ; T_{D2} = 0.275$$

#### 4.3.7 Implementación de los Desacopladores

Las funciones de transferencia de los dos desacopladores, de acuerdo con las ecuaciones 2.12 y 2.13 son:

$$D_{(1,2)T-ff} = 0.98 \left( \frac{4.7s+1}{4.5s+1} \right) e^{-0.02} \quad \text{y} \quad D_{(2,1)F-fc} = -0.68 \left( \frac{0.26s+1}{0.24s+1} \right) e^{-0.05}$$

### 4.3.8 Entrenamiento de la Red y Ajuste de los Parámetros del Controlador Neuronal.

Con el objeto de entrenar la red y determinar los parámetros de ajuste del controlador neuronal relativos al proceso del tanque de mezclado, se procede de manera similar a lo realizado para el proceso del tanque de calentamiento (Sección 4.2.8).

Por medio del programa en Simulink llamado Lazo\_Abierto\_2X2 se generan y almacenan los datos que se usan para el entrenamiento de la red. De acuerdo a lo explicado en la sección 3.2.2 siguiendo el método RAWN, para el proceso del tanque de calentamiento, la figura 4.21 muestra los valores del error,  $e$ , cometidos por la red neuronal durante el proceso de entrenamiento, donde, recordemos,  $e = [e1, e2] = [(c_1 - c_1 pred), (c_2 - c_2 pred)]$ .

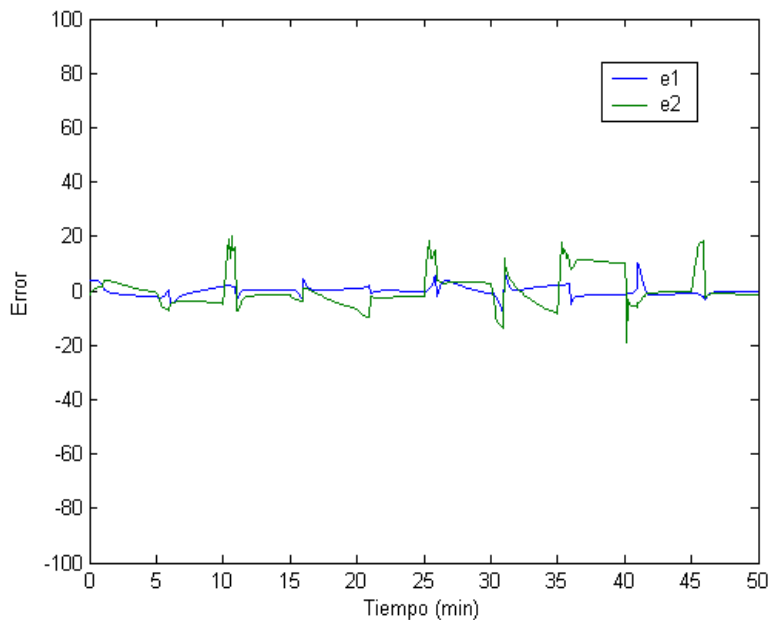


Figura 4.21 Error correspondiente a la diferencia entre las salidas reales y las predichas por la red

La calidad de las predicciones hechas por la red al comparar las curvas de las salidas reales del proceso ( $c_1$  y  $c_2$ ) con las salidas predichas por la red ( $c_1pred$  y  $c_2pred$ ) se muestran en la figura 4.22. Se observa que son muy similares, lo que indica que la red neuronal es un buen modelo del proceso.

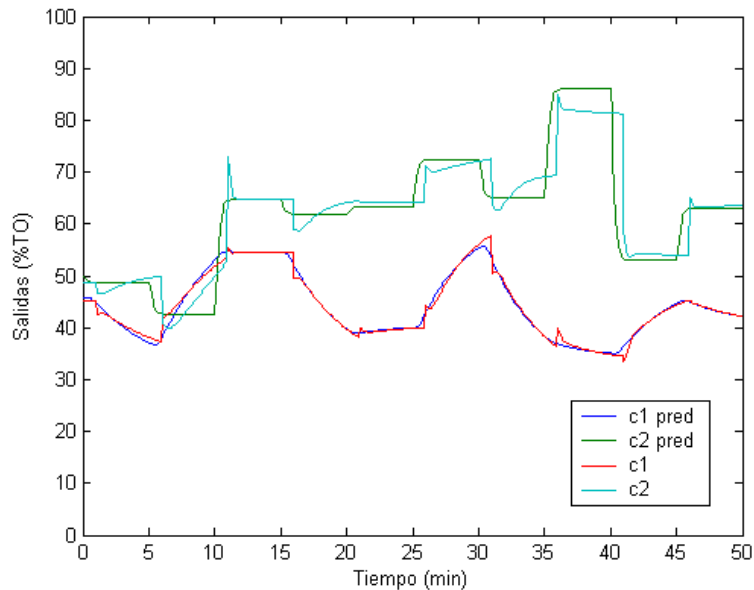


Figura 4.22 Comparación entre las salidas predichas por la red y las salidas reales del proceso.

Luego del entrenamiento se procede a determinar los parámetros de agresividad, tiempo de muestreo y número de predicciones para el controlador neuronal.

#### 4.3.8.1 Determinación del Factor de Agresividad

Para determinar el factor de agresividad se efectúa un cambio en el set-point de la temperatura,  $T$ , en  $t=10$ , y se mantiene el valor del punto de control del flujo,  $F$ , en su valor de estado estable. En la figura 4.23 se muestra las diferentes formas

en que la variable controlada  $T$  responde, ante dicho cambio, para distintos valores de agresividad ( $\lambda$ ) en el controlador.

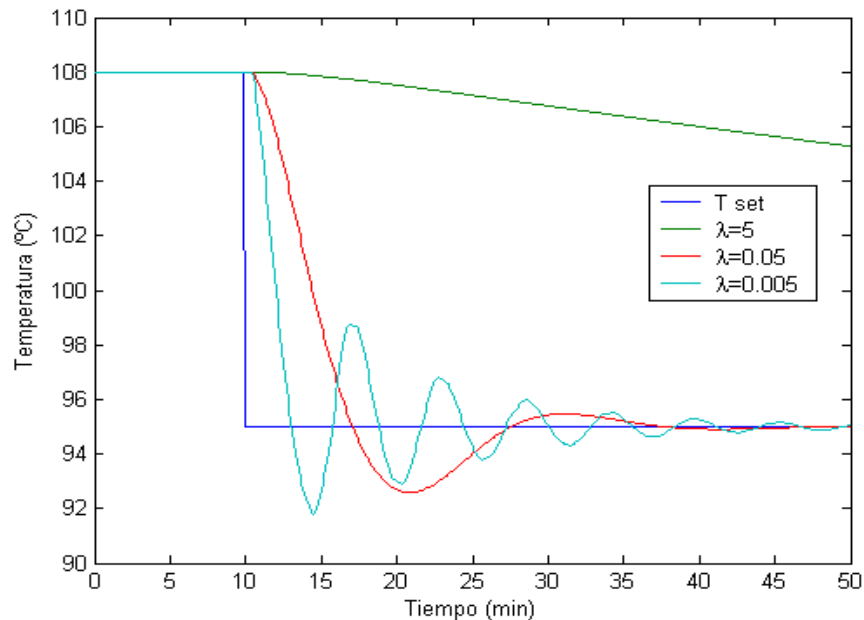


Figura 4.23 Comportamiento de la variable de salida  $T$ , para diferentes valores de agresividad en el controlador

En la figura 4.23 se evidencia que la peor respuesta se obtiene con el valor de agresividad más alto,  $\lambda=5$ . Para  $\lambda=0.005$  se observa que la temperatura presenta oscilaciones y tarda en estabilizarse. El mejor comportamiento se obtiene para un valor de agresividad igual a la unidad,  $\lambda=0.05$ .

#### 4.3.8.2 Determinación del Tiempo de Muestreo

En la figura 4.24 se muestran los distintos tipos de respuesta, con diferentes tiempo de muestreo, de la variable controlada temperatura ( $T$ ) cuando se presenta un cambio en su setpoint de 108 a 115 °C.

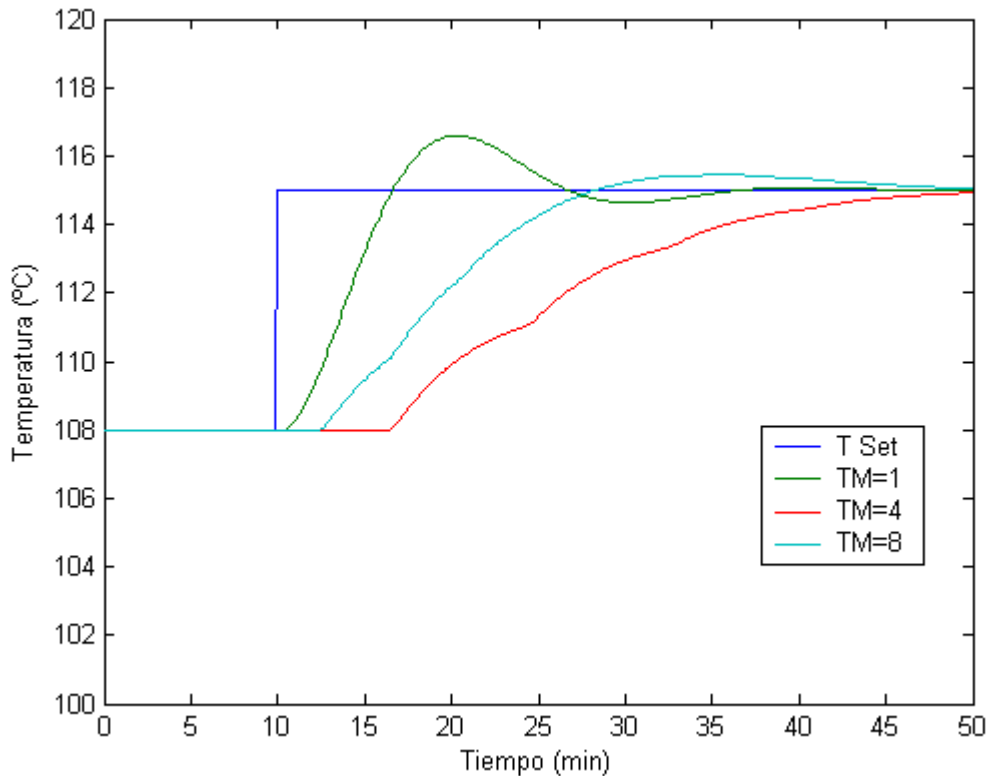


Figura 4.24 Comportamiento de la variable de salida  $T$ , para diferentes tiempos de muestreo en el controlador

Se observa que para un tiempo de muestreo igual a uno,  $TM=1$ , el flujo presenta un comportamiento óptimo. A medida que aumenta el tiempo de muestreo,  $TM=4$  y  $TM=8$ , la salida del proceso pierde velocidad de reacción ante el cambio realizado. La acción de control es lenta e inconvenientemente poco agresiva. Se opta, entonces, por escoger un valor de tiempo de muestro igual a dos,  $TM=1$ .

#### 4.3.8.3 Determinación del Número de Predicciones

Se procede a realizar un ejercicio de control y comparar los diversos tipos de respuestas de las variables controladas. En este caso se induce un cambio el



setpoint de la temperatura en  $t=10$  min, de 108 a 95 °C. En la figura 4.25 y 4.26 se muestra el comportamiento la temperatura y el flujo, respectivamente, para  $P=3$ ,  $P=4$  y  $P=15$  números de predicciones.

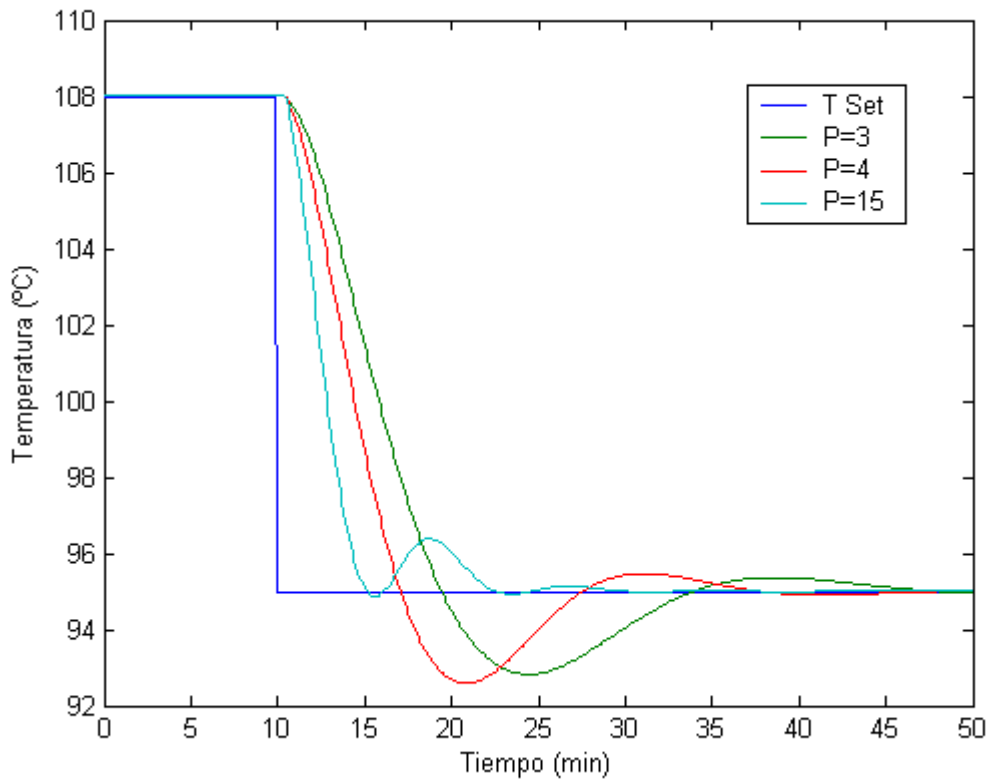


Figura 4.25 Comportamiento de la variable de salida  $T$ , para diferentes números de predicciones en el controlador

Para el caso de la temperatura, en la figura 4.25, se nota que a medida que aumentamos el número de predicciones, el control es mejor; sin embargo en la figura 4.26, la variable de salida flujo tiene un comportamiento inverso y entre mayor es el número de predicciones, tanto mas pobre es el control. Para equilibrar esta situación se opta por escoger un número de predicciones igual a 4 ( $P=4$ ).

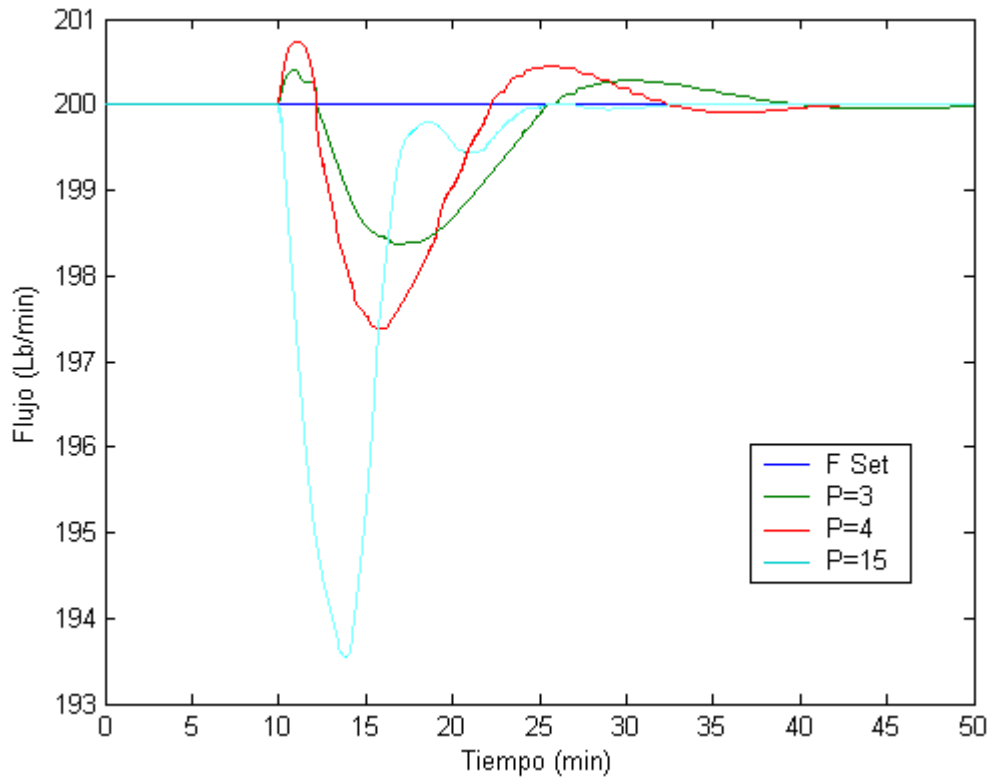


Figura 4.26 Comportamiento de la variable de salida  $F$  , para diferentes números de predicciones en el controlador

#### 4.4 PRUEBAS Y COMPARACIÓN DEL DESEMPEÑO SOBRE EL TANQUE DE CALENTAMIENTO

Para evaluar el desempeño del controlador neuronal, respecto al control PID y PID-D en el proceso del tanque de mezclado, se procede a realizar tres pruebas, en las dos primeras se realiza un cambio escalón en el set-point de las variables manipuladas y, en la tercera se introduce una perturbación al proceso con el fin de analizar las respuestas de los controladores ante dicho evento.

En la prueba 1 cambiamos el punto de control de temperatura de 180 a 95 °C en  $t = 10\text{min}$ , manteniendo el set-point de flujo fijo. En la prueba 2 cambiamos el set-point de flujo de 200 a 220  $lb/\text{min}$  en  $t = 10\text{min}$ , manteniendo el punto de control de temperatura fijo. En la prueba 3 se aplican dos cambios sucesivos de set-point y posteriormente se agrega una perturbación.

#### 4.4.1 Prueba 1. Cambio Escalón en el Set-point de Temperatura.

Para realizar esta prueba forzamos un cambio de tipo escalón en el set-point de temperatura de 108 a 95 °C en  $t = 10\text{min}$ , manteniendo el punto de control de flujo en su valor de estado estacionario.

##### 4.4.1.1 Comparación de las Salidas del Proceso.

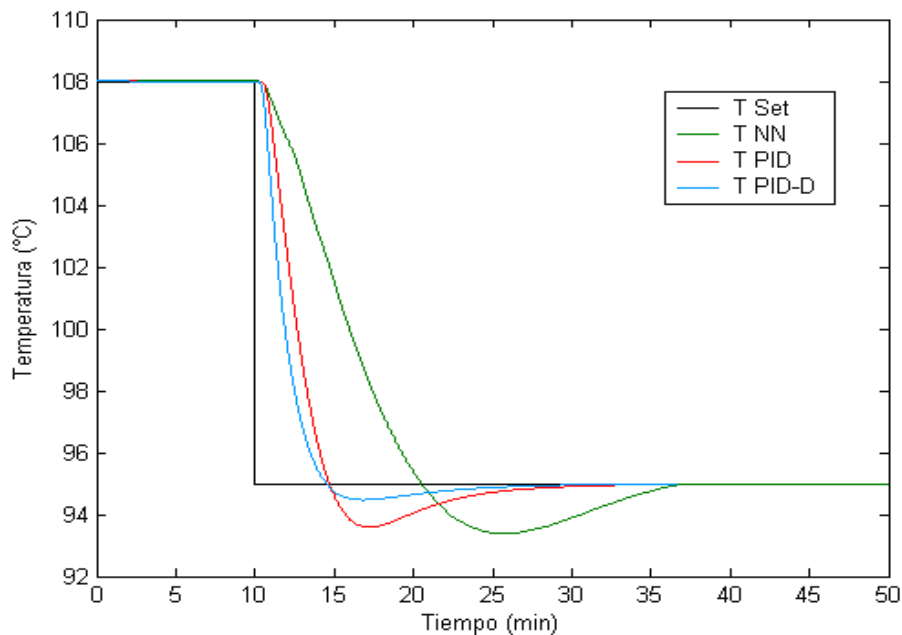


Figura 4.27 Respuesta de la variable de salida Temperatura ( $T$ ) ante un cambio escalón en su set-point.

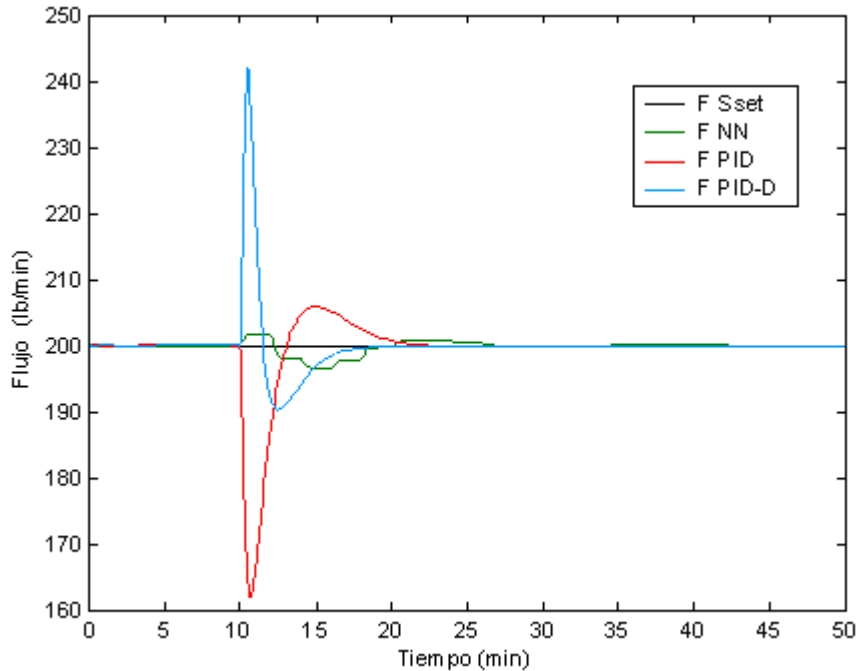


Figura 4.28 Respuesta de la variable de salida Flujo ( $F$ ) ante un cambio escalón en el set-point de la Temperatura.

De la gráfica 4.27 se observa que con todos los controladores se alcanza al llevar a la variable controlada a su nuevo valor requerido. En lo referente al comportamiento del flujo ante el cambio en la temperatura, se muestra en la figura 4.28 que con el control PID y PID-D se generan oscilaciones con grandes picos. Con los PID-D se produce uno por encima del valor solicitado de valor máximo  $243 \text{ lb/min}$ , desviándose  $43 \text{ lb/min}$ , y con el control PID se produce un pico por debajo del valor requerido de  $162 \text{ lb/min}$ . Por el contrario, con el controlador neuronal el flujo no presenta oscilación aunque sí alcanza a desviarse del valor del set-point llegando a  $196 \text{ lb/min}$ , con una desviación máxima de sólo  $4 \text{ lb/min}$ . En este punto podemos mencionar también que el controlador neuronal tarda un poco más de tiempo en estabilizar el flujo que el controlador PID y PID-D. El

controlador neuronal estabiliza el flujo en  $t = 27\text{min}$ , el control PID en  $t = 24\text{min}$  y el controlador PID-D lo estabiliza en  $t = 20\text{min}$ .

#### 4.4.1.2 Comparación de la Salida de los Controladores

En las gráficas 4.29 y 4.30 se ilustran las salidas del proceso. En éstas hay que resaltar que con el control neuronal se obtienen mejores señales desde el punto de vista de forzamiento de las válvulas de control, ya que son mucho más suaves, sin picos grandes ni oscilaciones prolongadas, sin embargo con el control PID y PID-D estas señales se logran estabilizar más rápidamente.

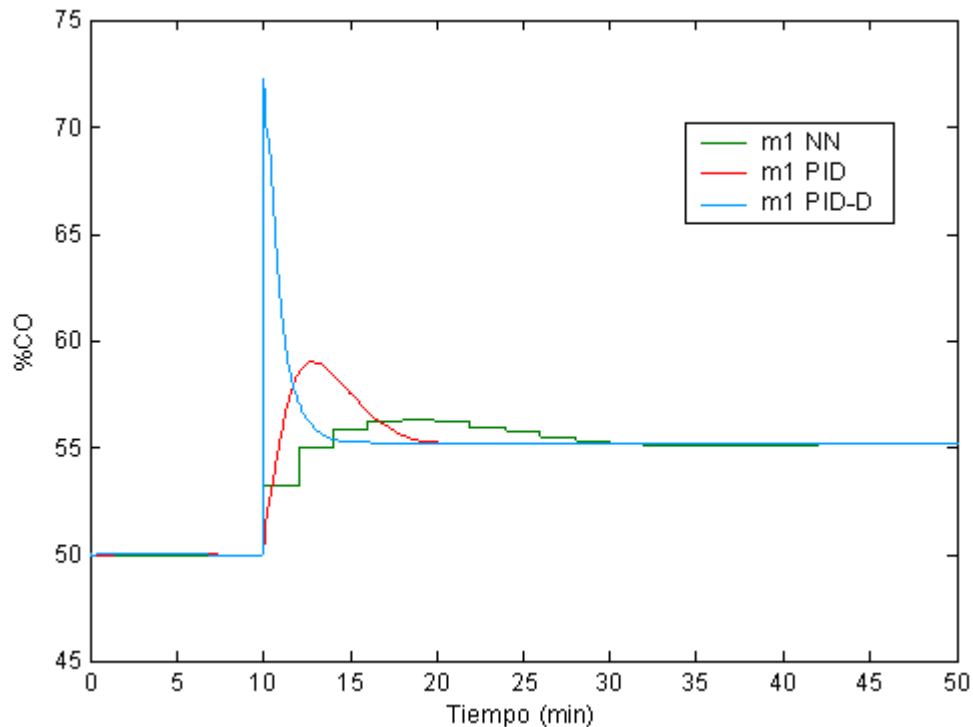


Figura 4.29 Respuestas de los controladores NN, PID y PID D para la variable manipulada uno (m1).

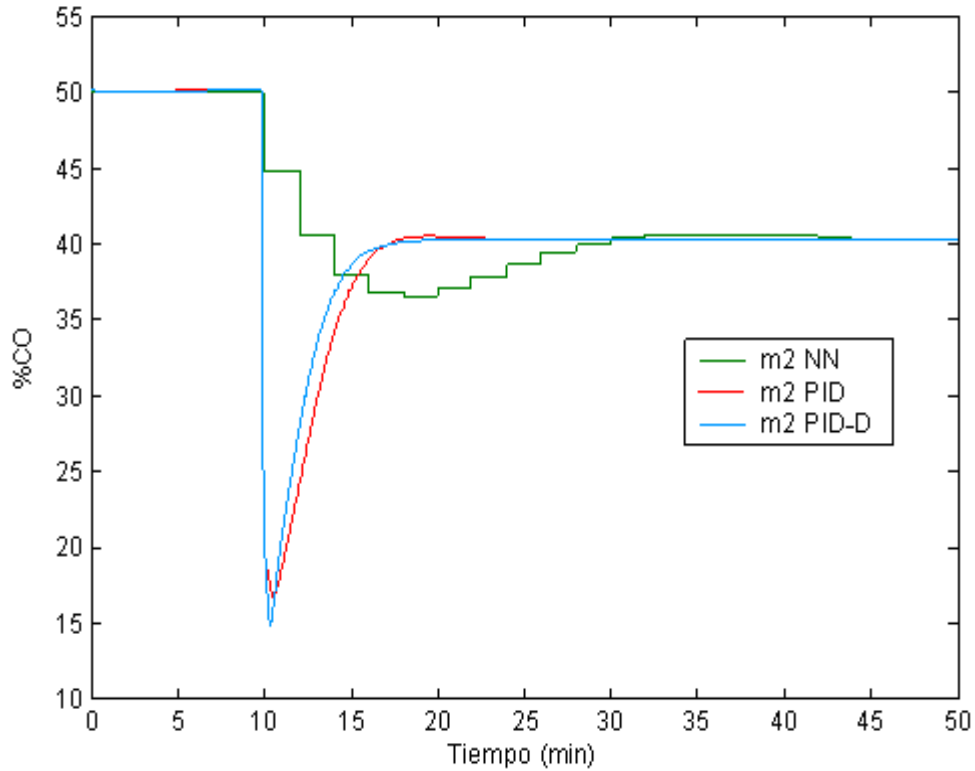


Figura 4.30 Respuestas de los controladores NN, PID y PID D para la variable manipulada dos (m2).

En la tabla 4.9 se muestran los valores de los parámetros comparativos logrados por cada tipo de control.

Tabla 4.9

<b>PRUEBA 1 (Tanque de Calentamiento)</b>			
<b>PRUEBA 1</b>	<b>IAE</b>		
	<b>TIPO DE CONTROL</b>		
	<b>PID</b>	<b>PID-D</b>	<b>NN</b>
<b>Lazo 1</b>	24,1	16,5	49
<b>Lazo 2</b>	20,8	16	3,8

<b>INDICE DE SUAVIDAD (Salida del proceso)</b>			
<b>PRUEBA 1</b>	<b>TIPO DE CONTROL</b>		
	<b>PID</b>	<b>PID-D</b>	<b>NN</b>
	<b>Lazo 1 (°C)</b>	2	1
<b>Lazo 2 (lb/min)</b>	39	43	3

INDICE DE SUAVIDAD (Salida del controlador)			
PRUEBA 1	TIPO DE CONTROL		
	PID	PID-D	NN
m1 (%CO)	3	18	1
m2 (%CO)	23	25	3

TIEMPO DE ESTABLECIMIENTO (min)			
PRUEBA 1	TIPO DE CONTROL		
	PID	PID-D	NN
Lazo 1	33	33	36
Lazo 2	23	18	27

Para el lazo de temperatura, lazo 1, el control neuronal arroja mayores valores de IAE; en cambio para el lazo de flujo, lazo 2, arroja menores valores. Estos resultados concuerdan con los vistos en las gráficas 4.27 y 4.28.

#### 4.4.2 Prueba 2. Cambio Escalón en el Set-point de Flujo.

Procedemos a cambiar el set-point de Flujo de 200 a 220 *lb/min* en  $t = 10\text{min}$ , manteniendo el punto de control de temperatura fijo.

##### 4.4.2.1 Comparación de las Salidas del Proceso.

La respuesta del proceso ante este cambio se muestra en las figuras 4.31 y 4.32. En ellas se observa que los diferentes métodos de control logran estabilizar las variables controladas del proceso ante el cambio efectuado.

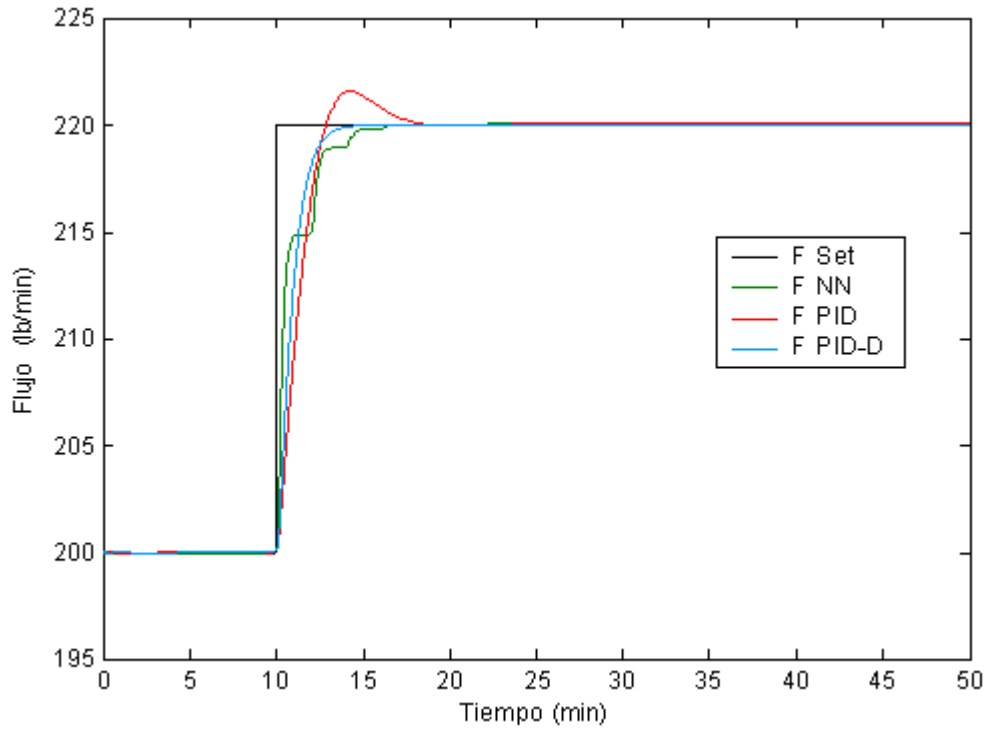


Figura 4.31 Respuesta de la variable de salida Flujo ( $F$ ) ante un cambio escalón en su set-point.

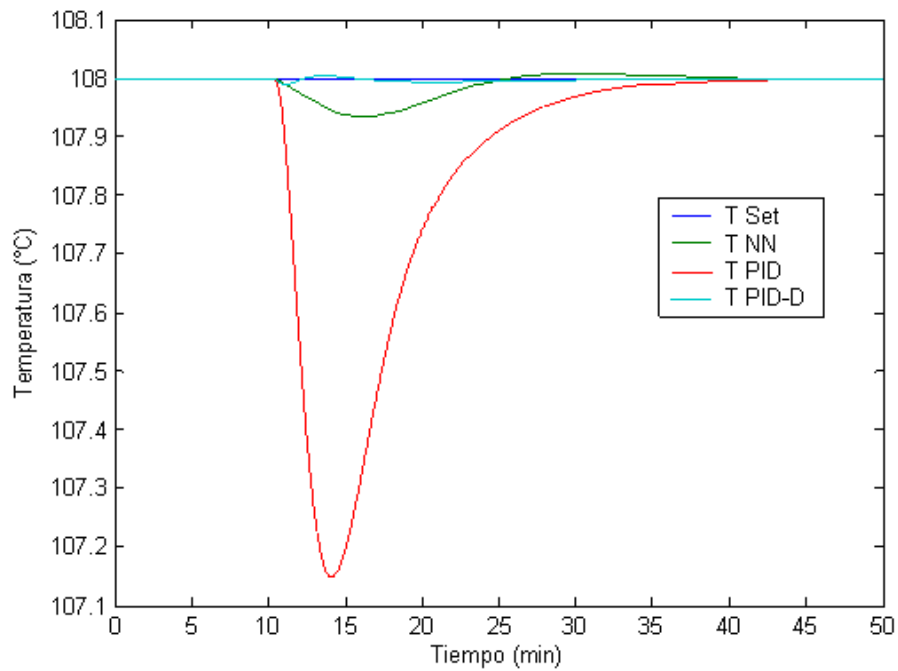


Figura 4.32 Respuesta de la variable de salida Temperatura ( $T$ ) ante un cambio escalón en el set-point de Flujo



En el caso del control de flujo las salidas del proceso son bastantes similares para el caso NN y PID-D, en cuanto a la forma en que se alcanza el objetivo de control y en el tiempo de establecimiento de las señales. No se presentan ni picos ni oscilaciones para esta variable de salida en ninguno de los dos anteriores tipos de control. Cuando se utiliza el control PID se producen picos tanto en el flujo como en la temperatura antes de alcanzar la estabilización de estas variables. En lo referente a la temperatura se observa que con el control PID-D se produce una respuesta casi desapercibible al cambio en el set-point de flujo, con el controlador neuronal la temperatura se alcanza a desviar  $0.05\text{ }^{\circ}\text{C}$  y con el control PID la oscilación producida lleva a una desviación de  $0.8\text{ }^{\circ}\text{C}$  de su valor requerido.

#### 4.4.2.2 Comparación de la Salida de los Controladores

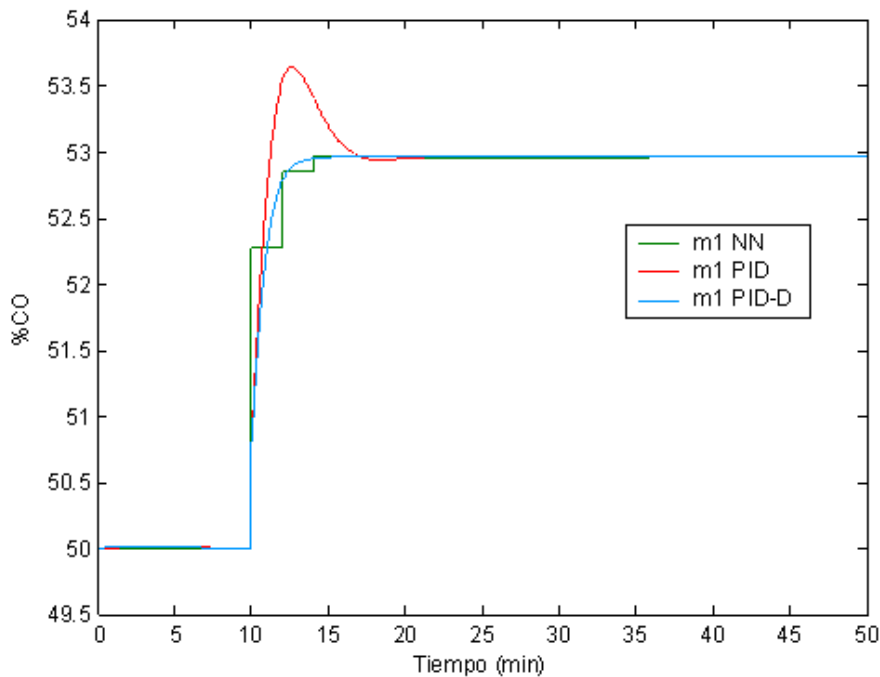


Figura 4.33 Respuestas de los controladores NN, PID y PID-D para la variable manipulada uno (m1).

En las gráficas 4.33 y 4.34 se muestran las señales que salen de los respectivos controladores.

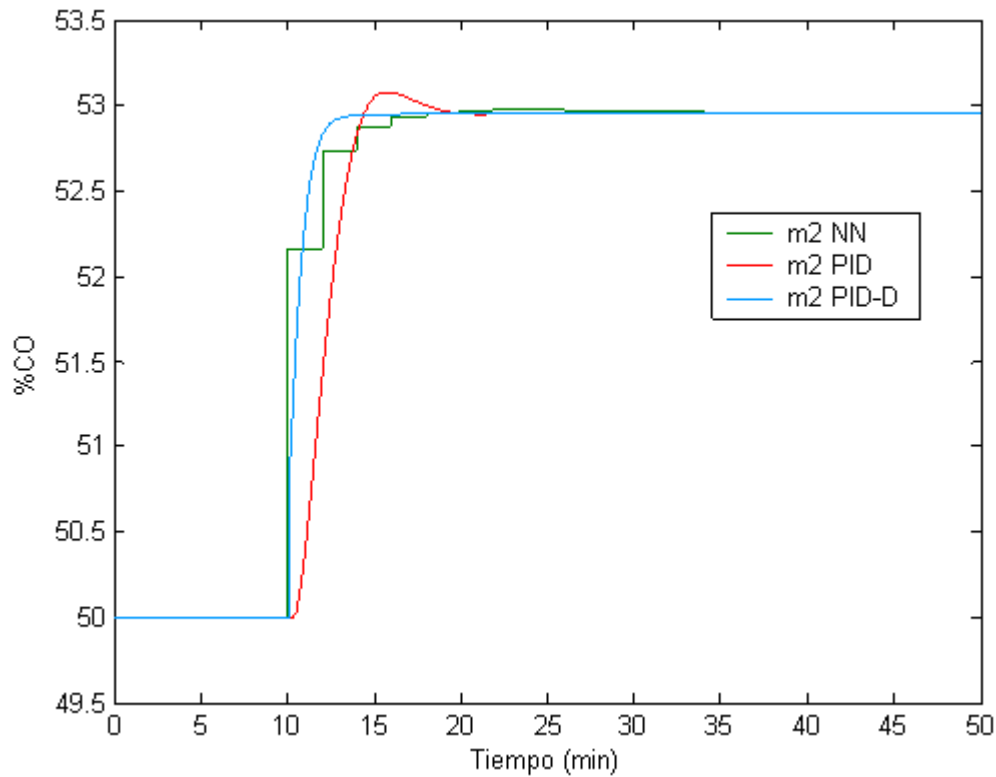


Figura 4.34 Respuestas de los controladores NN, PID y PID-D para la variable manipulada dos (m2).

En este caso el comportamiento de las señales es similar para los procesos controlados por los PID-D y NN, ambos tipos de control, sin picos ni grandes oscilaciones. Con el control PID se presenta overshoot en las salidas de los controladores. Además, la acción del control NN y PID-D es más rápida que la proporcionada por el control PID ya este último tarda más en conseguir estabilizar su señal de salida.

En cuanto a los valores de IAE que se obtuvieron en esta prueba, los resultados se muestran en la tabla 4.10, y de ellos se verifica que nuevamente para el lazo de temperatura, lazo 1, el control NN arroja mayores valores de IAE respecto al control PID-D y menores valores respecto al PID; en cambio para el lazo de flujo, lazo 2, el control neuronal arroja los menores valores. Estos resultados concuerdan con los vistos en las gráficas 4.31 y 4.32

Tabla 4.10

<b>PRUEBA 2 (Tanque de Calentamiento)</b>			
<b>IAE</b>			
<b>PRUEBA 2</b>	<b>TIPO DE CONTROL</b>		
	<b>PID</b>	<b>PID-D</b>	<b>NN</b>
<b>Lazo 1</b>	3,7	0,06	0,38
<b>Lazo 2</b>	7,1	5,5	4,6

<b>INDICE DE SUAVIDAD (Salida del proceso)</b>			
<b>PRUEBA 2</b>	<b>TIPO DE CONTROL</b>		
	<b>PID</b>	<b>PID-D</b>	<b>NN</b>
<b>Lazo 1 (°C)</b>	0,85	0,01	0,06
<b>Lazo 2 (lb/min)</b>	2	0,01	0,01

<b>INDICE DE SUAVIDAD (Salida del controlador)</b>			
<b>PRUEBA 2</b>	<b>TIPO DE CONTROL</b>		
	<b>PID</b>	<b>PID-D</b>	<b>NN</b>
<b>m1 (%CO)</b>	0,7	0,01	0,01
<b>m2 (%CO)</b>	0,2	0,01	0,01

<b>TIEMPO DE ESTABLECIMIENTO (min)</b>			
<b>PRUEBA 2</b>	<b>TIPO DE CONTROL</b>		
	<b>PID</b>	<b>PID-D</b>	<b>NN</b>
<b>Lazo 1</b>	45	22	40
<b>Lazo 2</b>	18	14	16

Se observa un desempeño superior del control neuronal y PID-D respecto al control PID, sobre todo más notable en cuánto a los índices de suavidad.

#### 4.4.3 Prueba 3. Cambios Sucesivos de Set-point más Perturbación

En esta prueba se realiza un paso escalón en el set-point de flujo en  $t = 5$ , luego un paso escalón en el punto de control de temperatura en  $t = 15$ , y posteriormente se induce un cambio escalón en la temperatura del flujo frío (perturbación), la cual pasa de 60 a 66 °C en  $t = 25$ . Los resultados se muestran en la gráfica 4.36.

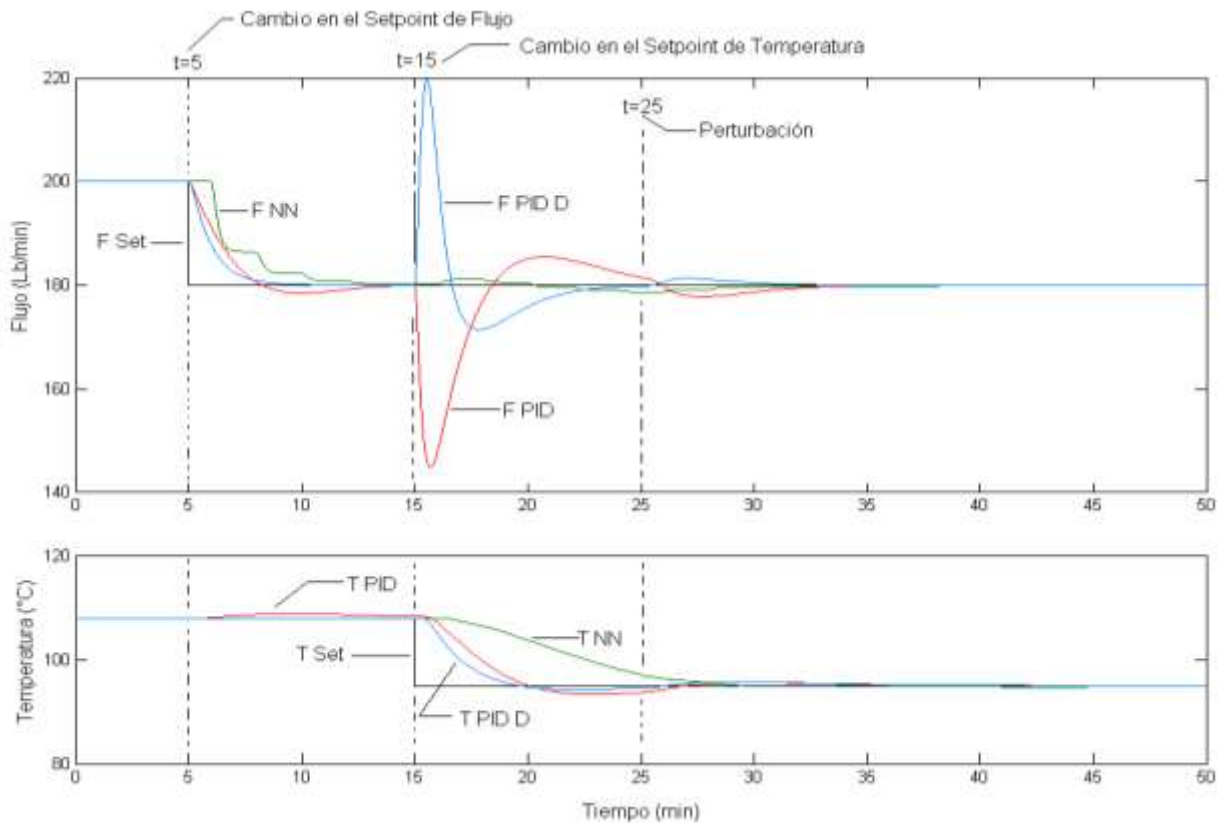


Figura 4.36 Salidas del proceso ante los cambios de set-points y perturbación realizados.

Lo más sobresaliente de los resultados mostrados en la gráfica 4.36 son los picos que alcanza el Flujo, en  $t = 15$ , con el control PID y PID-D cuando se presenta el cambio en el set-point de temperatura. A diferencia de lo anterior, las variables

que son controladas por el sistema neuronal alcanzan su valor requerido sin picos ni oscilaciones. También es importante anotar que todos los métodos de control tienen un buen desempeño cuando al proceso ingresa la perturbación realizada, logrando cada uno estabilizar de manera adecuada la salida del proceso.

#### 4.4.4 Prueba 4. Cambios Simultáneos de Set-point más Perturbación y Presencia de Ruido

Los factores que causan que una característica funcional, como por ejemplo, la señal de salida de un proceso, se desvíe de su valor objetivo, se llaman factores de ruido. Los factores de ruido causan variación y pérdida de calidad, además de convertirse en una dificultad más al momento de realizar un control de proceso. En general a todos los factores incontrolables se les denomina factores de ruido. Ruido puede definirse como cualquier fenómeno que causa a una determinada característica desviarse de su valor objetivo. La figura 4.37 muestra el efecto del ruido sobre la salida de la temperatura en el proceso del tanque de mezclado.

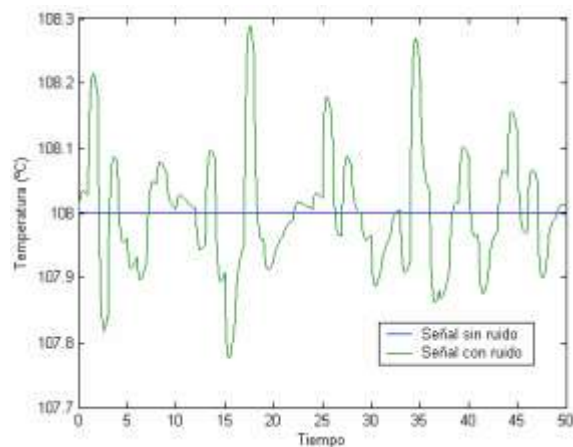


Figura 4.37 Efecto del ruido sobre la variable de salida Temperatura.

Para esta última prueba se realizará el siguiente ejercicio de control: cambio en set-point del flujo: 200-210 lb/min, cambio en el set-point de temperatura: 108-98 °C y entrada de perturbación Tf: 60-54 °C , todo en el tiempo t=10 minutos.

Con el objeto de observar el comportamiento de la red ante la presencia de ruido se ha optado por realizar un nuevo entrenamiento, esta vez se entrena a la red con datos que incluyen el efecto del ruido. En las figuras 4.38 y 4.39 se ilustran los diferentes tipos de respuestas obtenidos por la red entrenada con ruido y la no entrenada con ruido.

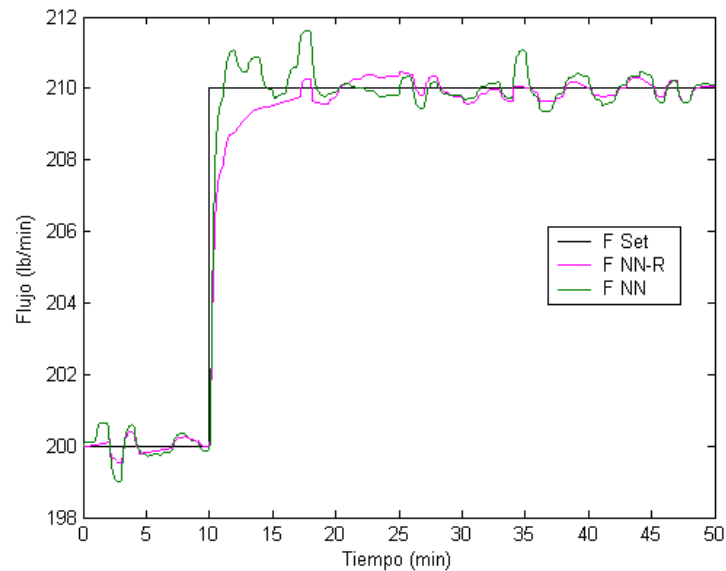


Figura 4.38 Respuesta de la variable de salida Flujo ante los cambios en los puntos de control y perturbación realizados.

Se observa en ambas figuras que la red entrenada con datos que contenían ruido tiene una mejor respuesta de control ante la prueba realizada.

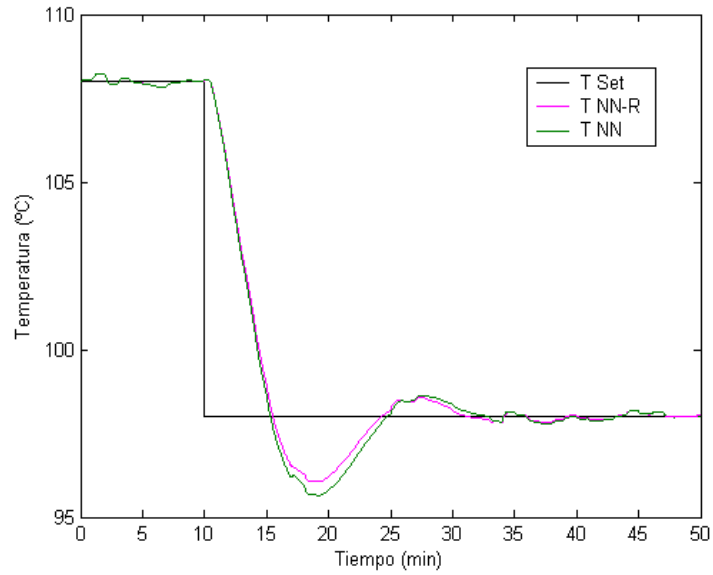


Figura 4.39 Respuesta de la variable de salida Temperatura ante los cambios en los puntos de control y perturbación realizados.

En las figuras 4.40 y 4.41 se incluyen las respuestas de los controladores PID ante los cambios realizados en esta prueba y se comparan con los obtenidos por la red entrenada con ruido.

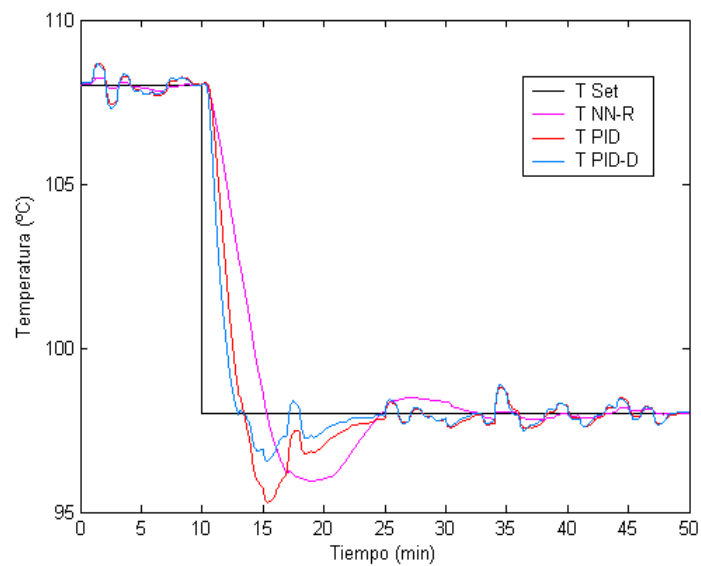


Figura 4.40 Respuesta de la variable de salida Temperatura ante los cambios en los puntos de control y perturbación realizados.

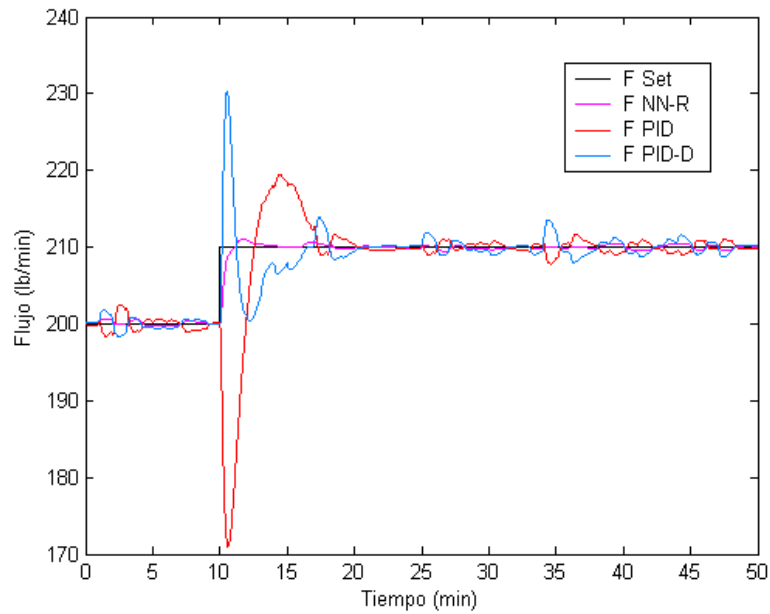


Figura 4.41 Respuesta de la variable de salida Flujo ante los cambios en los puntos de control y perturbación realizados.

De la prueba anterior se observa un comportamiento superior del controlador neuronal, ya que la red entrenada con ruido está preparada para enfrentar mejor dicho fenómeno. Entre los controladores PID se tiene que los desacopladores ejercen un efecto de filtro en las señales de salida, atenuando mejor los efectos del ruido en comparación con el control PID no desacoplado.



## 5. CONCLUSIONES

El controlador utilizado en el esquema propuesto, trabaja con base en el algoritmo de optimización (SQP) [21] para minimizar la función objetivo definida como el error cuadrático medio entre la señal de referencia y la salida del modelo neuronal del proceso, razón por la cual, el modelo neuronal que representa la dinámica de la planta, debe ser lo más exacto posible. El sistema de control una vez implementado, fue probado mediante ensayos de simulación controlando sistemas de dos entradas y dos salidas. Como entradas de referencia se aplicaron diferentes señales escalón pudiéndose observar que se logra eliminar los problemas producidos por la interacción de variables que se dan típicamente en procesos multivariantes proporcionando un buen desacoplo entre los pares de variables manipuladas y controladas.

Se demostró que la estrategia de control predictivo usando redes neuronales con entrenamiento rápido para sistemas  $2 \times 2$ , provee buenos resultados, tan buenos como los obtenidos por un sistema PID desacoplado y ampliamente superiores a los producidos por un sistema controlado por PID sin desacoplar.

Prácticamente las señales de salida del proceso y las de referencias logran estabilizarse adecuadamente, eliminando absolutamente los problemas de interacción de las variables y el uso de desacopladores, problemas que se producen en sistemas multivariables que hacen que los criterios clásicos de control utilizados en sistemas SISO no sean de aplicación inmediata en sistemas MIMO. El problema de interacción de variables se traduce en que, las variaciones de una variable de control, provocan variación en las otras, lo que obliga a tomar nuevas acciones sobre el proceso y todo ello en forma encadenada.

Fue posible comprobar mediante diferentes tipos de pruebas, que el controlador neuronal tiene un rendimiento superior o igual a dos controladores PID desacoplados y absolutamente superior a dos controladores PID sin desacoplar; haciéndose más visible esta superioridad cuando se comparan la calidad de las señales enviadas a las válvulas. Precisamente el controlador neuronal ejerce una acción de control eficaz, pero minimizando el impacto de los cambios abruptos en la salida del controlador, hecho que evita un detrimento del elemento final de control.

La robustez del controlador neuronal fue comprobada al ser puesto en funcionamiento sobre dos procesos diferentes, donde se controlaron variables de diferente tipo como temperatura, flujo y concentración, sin la necesidad de hacer ningún cambio en la estructura básica del controlador. Los cambios realizados para cada proceso en particular se limitan a recalcular el horizonte de predicción y el factor de agresividad que produce la mejor respuesta posible.

Además, el hecho de utilizar el método RAWN para entrenar la red neuronal, permite al controlador realizar una labor de control mediante un procesamiento rápido, ya que si se hubiese usado backpropagation, como método de entrenamiento, la fase de reentrenamiento hubiese implicado una complejidad aún mayor.

El entrenamiento realizado con el método de generación aleatoria de los pesos de activación RAWN, fue satisfactorio y permitió que la red pudiera ser implementada en un esquema predictivo de control, como un predictor neuronal capaz de hacer predicciones con un mínimo margen de error. Lo que indica que el método RAWN es el indicado si se desea hacer uso de redes neuronales que requieran de un entrenamiento rápido.

Puede concluirse a partir de las pruebas realizadas, que como el control neuronal tiene en muchos casos desempeños muy parecidos al control PID desacoplado y en varias ocasiones mejor desempeño, es posible llevar a una mayor implementación de este tipo de sistemas de control, al que puede verse como una alternativa efectiva para aquellos procesos en los cuales el control PID no es satisfactorio. Además con la implementación de la prueba con ruido en los procesos se comprobó que la red neuronal entrenada con datos que incluían dicho fenómeno presentó un comportamiento superior que los PID.

## 6. APORTES Y RECOMENDACIONES

### 6.1 APORTES

Los resultados más importantes y novedosos que se pueden destacar producto del presente trabajo, se pueden resumir en los siguientes aspectos:

La innovación y la principal diferencia de esta investigación frente a estudios previos consiste en brindar una alternativa distinta de los tradicionales PID, eficaz y novedosa, que permite controlar procesos multivariables no lineales, sin requerir la implementación de desacopladores, y que hace uso de un sistema de control basado en redes neuronales.

El procedimiento descrito en la sección 2.2 acerca del diseño de desacopladores para sistemas  $2 \times 2$  es simple, pero éste no puede extenderse para sistemas de más de dos objetivos de control que interactúan. El diseño de desacopladores para sistemas de orden superior no es sencillo debido al elevado número de desacopladores requeridos. Por ejemplo, para un sistema  $4 \times 4$ , se requiere diseñar e implementar 20 desacopladores, y para uno de  $5 \times 5$  se necesitan 30. Debido a esto se han diseñado modernos computadores para el control de

sistemas multivariables que trabajan por diferentes métodos como el DMC (Dynamic Matriz Control), y el IdCom (Identificación Command). En este punto el presente trabajo cobra relevancia ya que se logra diseñar un sistema de control  $2 \times 2$  que puede ser fácilmente extensible a sistemas de orden superior, y el cual no necesita de la aplicación de desacopladores.

También destacamos que para sistemas de orden mayor al  $2 \times 2$  aún no se han determinado las ecuaciones de sintonización adaptadas para los controladores PID que se pretendan implementar. Esto coloca un punto de interés adicional al presente trabajo ya que en éste, se logra desarrollar un sistema de control predictivo robusto, óptimo y eficiente que iguala o mejora al sistema que usa controladores PID con desacopladores y que además, logra superar ampliamente al control PID cuando no se implementan tales desacopladores.

Otro aspecto importante para destacar, es que el sistema de control predictivo desarrollado, trabaja en base a un modelo neuronal de predicción del proceso. Este modelo neuronal es entrenado fuera de línea con el método RAWN, un método rápido y eficaz.

## **6.2 RECOMENDACIONES**

Al realizar esta investigación, se puede percibir que no hay un solo camino para la obtención de mejores resultados, sin embargo luego de realizar este estudio sugerimos para estudios posteriores las siguientes recomendaciones.

Una buena practica si alguien esta interesado es realizar pruebas acerca de los valores de los parámetros de agresividad y horizonte de predicción, encontrados en el proceso de optimización, es posible obtener mejores resultados.

Es recomendable evaluar bien, para qué aplicación se decide implementar un controlador usando redes neuronales, ya que si bien las redes neuronales son una buena opción para controlar procesos industriales, en especial los que se caracterizan por su comportamiento no lineal, también es cierto que existen aplicaciones donde otras técnicas probadas han sido exitosas, siendo innecesario hacer sistemas más complejos como las redes neuronales.

Una buena alternativa para hacer pruebas en estudios futuros, es dedicar esfuerzos en observar si el desempeño del controlador neuronal mejora si los parámetros de agresividad y horizonte de predicción dejan de ser fijos, es posible que esto permita que el controlador sea más versátil, siendo capaz de ser poco o muy agresivo de acuerdo a la situación y, de tener el mismo comportamiento frente a todos los cambios.

Si se desea llevar a cabo una implementación en hardware de este tipo de controlador, recomendamos usar dispositivos como DSP's (Procesadores Digitales de Señales) con suficiente capacidad de procesamiento para poder hacer los cálculos necesarios en un tiempo menor a un período de muestreo del proceso que se intente controlar.

## REFERENCIAS

- [1] Dorf, Richard y Bishop, Robert. Sistemas de Control Moderno. Pearson. España, 2005.
- [2] Smith, Carlos y Corripio, Armando. Principles and Practice of Automatic Proccess Control, John Wiley & Sons, New York, 1997.
- [3] Eronini Umez-Eronini. Dinámica de Sistemas y Control. Thomson Learning. Mexico, 2001.
- [4] Medina, Maximino. 1991, "A Tuning Technique For Decentralized Multivariable Control".
- [5] [Chen, Qing](#) ; [Li, Shao-Yuan](#) and [Xi, Yu-Geng](#). Decentralized closed-loop identification and controller design for a kind of cascade systems. Source: Zidonghua Xuebao/Acta Automatica Sinica, v 31, n 3, May, 2005, p 335-342

[6] [Lin, C.-H.](#) ; [Han, K.-W.](#) Prediction of limit-cycles in nonlinear two-input two-output control systems. Source: IEE Proceedings: Control Theory and Applications, v 146, n 3, May, 1999, p 253-258

[7] Rojas, Raúl. Neural Networks. A Systematic Introduction. Springer. Germany, 1996.

[8] Kung, Sun Yuan. Digital Neural Networks. Prentice-Hall. USA, 1993.

[9] Ospino, Efren y Saucedo, Juan, Controlador Con Redes Neuronales De Entrenamiento Rápido Basado En Identificación Del Proceso Como Un Sistema De Primer Orden Más Tiempo Muerto. Tesis de pregrado, Universidad del Norte, 2003

[10] Isasi, Pedro y Galván, Inés. Redes Neuronales Artificiales. Un Enfoque Práctico, Pearson Prentice Hall, Madrid, 2004.

[11] [Cybenko, G.](#), 1989, "Approximation By Superpositions Of A Sigmoidal Function", Mathematics of Control, Signals, and Systems, v 2, n 4, 1989, p303-314  
CODEN:MCSYE8

[12] Freeman, James y Skapura, David. Redes Neuronales-Algoritmos, Aplicaciones y Técnicas de Programación.1993.



[13] [Hopfield, John J.](#), 1988, "Artificial neural networks" Source: IEEE Circuits and Devices Magazine, v 4, n 5, p 3-10. ISSN: 8755-3996 CODEN: ICDMEN

[14] Te Braake, Hubert y Van Straten, Gerrit., 1994, "Random Activation Weight Neural Net (RAWN) for Fast Non-iterative Training".

[15] Perez, Rafael, 2000. Neural Networks Class Notes. University of South Florida. Computer Science Engineering Department, 2000.

[16] Quin, S., Badgwell, T., 2003, "A Survey of Industrial Model Predictive Control Technology", Control Engineering Practice, 11(7), 733-764.

[17] [Airikka, Pasi.](#), 2004, "Advanced Control Methods For Industrial Process Control" Source: IEE Computing and Control Engineering, v 15, n 3, p18-23 ISSN: 0956-3385 CODEN:CCEJEL

[18] Bordons, Carlos., 1998, "Generalized Predictive Controller for a Wide Class of Industrial Processes" Source: *IEEE Transactions on Control Systems Technology*, v 6, n 3, May, 1998, p 372-387

[19] Sanjuán, Marco, 2005. "Neural-Network Based On-Line Adaptation Of Model Predictive Controller For Dynamic Systems With Uncertain Behavior". Tomado de ASME International Mechanical Engineering Congress And Exposition, IMECE2005-82609.

[20] Simulink Model-Based and System-Based Design. Writing S-Functions, Versión 5. The Mathworks Inc., 2004

[21] Optimization Toolbox for use with MATLAB®. User's Guide, Versión 2. The Mathworks Inc., 2003

## ANEXO A

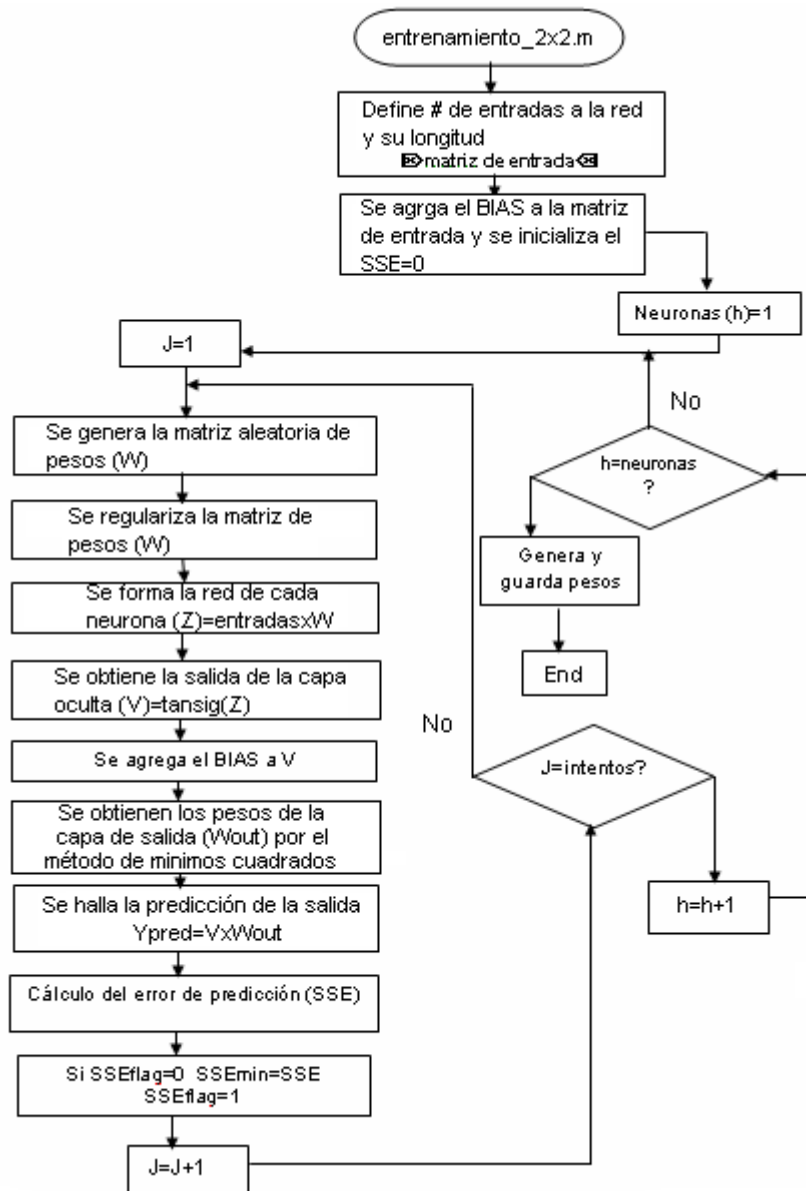
### CÓDIGOS DE FUENTE DEL ALGORITMO DE CONTROL

#### Entrenamiento\_2X2.m

```
function [Wh,Bh,Wo,Bo,neuronas,SSERROR,Y,Ymin,e]=entrenamiento_2X2(entradas,salidas);
%entradas=[md,cd] ; salida=[c1,c2]; md=[m1d1,m2d1] ; cd=[c1d1,c2d1]
Ni=min(size(entradas)); %numero de entradas
Ne=length(entradas);
Y=salidas; %Salida del proceso c(n)
entradas(1:Ne,Ni+1)=ones(Ne,1); %adicion del bias a la matriz de entrada
%Iniciando La suma del cuadrado del error.
SSEflag=0;
for h=1:8; %numero maximo de neuronas con que se desea evaluar la red
for j=1:16 %numero de pruebas con matrices aleatorias que hace la red para realizar las predicciones con el
minimo SSE posible
W=zeros(Ni+1,h); %Iniciando la matriz de pesos
R=randn(Ni+1,h); %Matriz aleatoria
W=sqrt(1/(max(sum(entradas.^2))))*R; %proceso de regularizacion
Z=entradas*W; %red de cada neurona
V=tansig(Z); %salida de cada neurona
V(:,h+1)=ones; %adicion del bias de entrada
Wout=inv(V'*V)*V'*Y; %metodo de los m´inimos cuadrados
Ypred=V*Wout;
SSE=sumsq(Y-Ypred); %calculo del SSE
if SSEflag==0
SSEmin=SSE; %Initializing SSEmin
SSEflag=1; %blocking
end
%separacion de los pesos y bias
if SSE<= SSEmin
SSEmin=SSE; %minimo SSE
Ymin=Ypred;
Wh=W(1:Ni,:);
Bh=W(Ni+1,:);
Wo=Wout(1:h,:);
Bo=Wout(h+1,:);
neuronas=h; %numero optimo de neuronas
e=Y-Ypred;
end
```

end  
end  
SSERROR=SSEmin  
neuronas=neuronas

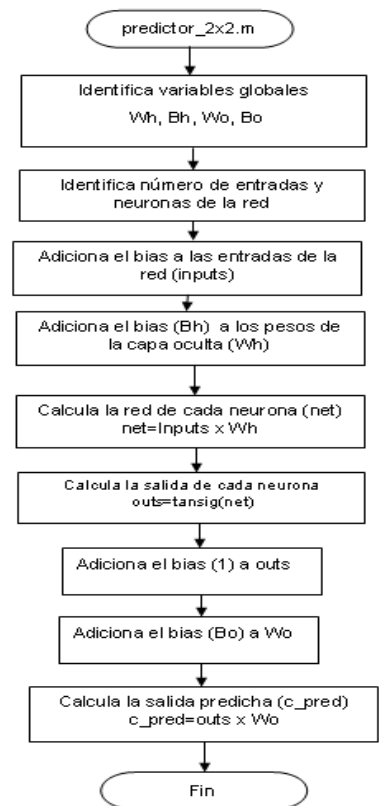
### DIAGRAMA DE FLUJO DE entrenamiento\_2X2



## predictor\_2X2.m

```
% este algoritmo realiza las predicciones
function [c_pred]=predictor_2X2(inputs);
% inputs=[md,cd]; md=[m1d1,m2d1] ; cd=[c1d1,c2d1]
global Wh Bh Wo Bo
load Wh
load Wo
load Bh
load Bo
Ni=min(size(Wh)); % numero de elementos de entrada; Ni=4
nodes=max(size(Wh)); % numero de neuronas en la capa intermedia; nodes=8
g=length(inputs); % g=501
inputs(:,Ni+1)=ones; % suma del bias; 501X5
Wh(Ni+1,:)=Bh; % matriz de pesos de la capa intermedia; 5X8
net=inputs*Wh; % red de cada neurona, 501X8
outs=tansig(net); % salida de cada neurona; 501X8
outs(:,nodes+1)=ones; % 501X9
Wo(nodes+1,:)=Bo; % matriz de pesos de la capa de salida
c_pred=outs*Wo;% salida predicha, 501X2
```

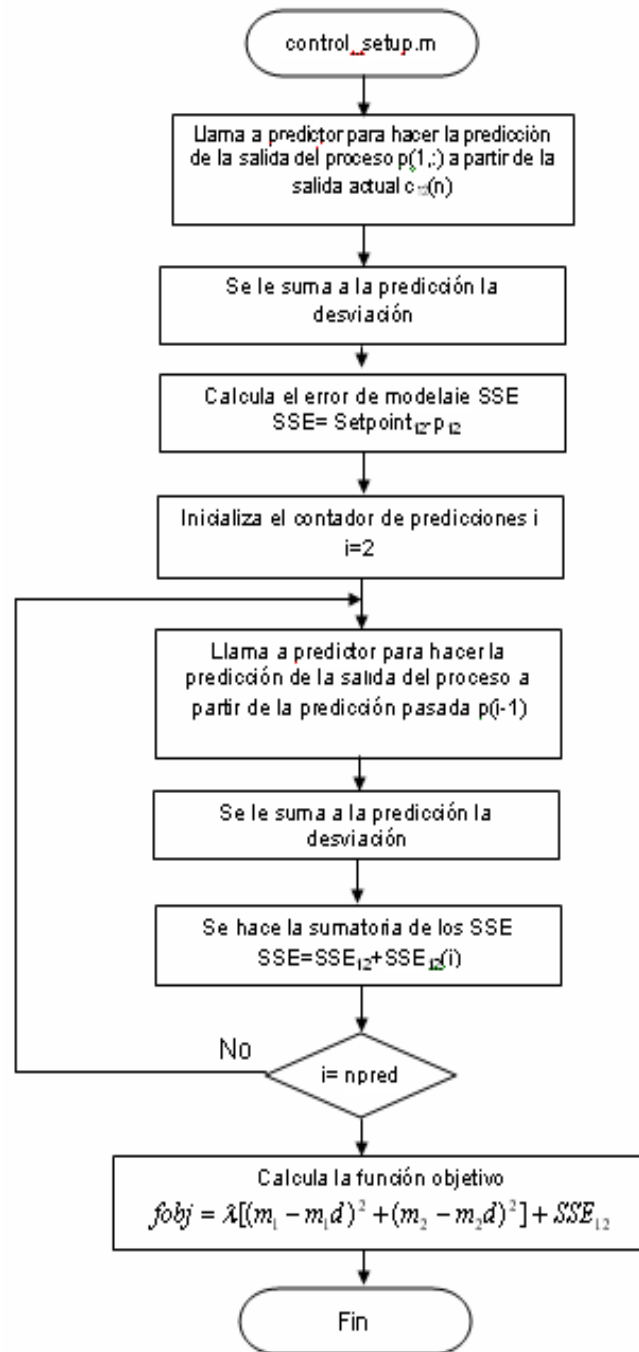
### DIAGRAMA DE FLUJO DE predictor\_2X2



## control\_setup.m

```
function fobj=control_setup_2X2(M,fin);
%fin=[m2d1;c1;c1d1;e1;SP1;m1d1;c2;c2d1;e2;SP2]
%M=[m1;m2]
m2d1=fin(1);
c1=fin(2);
c1d1=fin(3);
d1=fin(4);
SP1=fin(5);
m1d1=fin(6);
c2=fin(7);
c2d1=fin(8);
d2=fin(9);
SP2=fin(10);
m1=M(1);
m2=M(2);
% se predice la salidad del proceso npred pasos adelante y se calcula el SSE
d=[d1,d2]; %NiX2
fin61=[fin(6),fin(1)];
fin27=[fin(2),fin(7)];
p(1,:)=predictor_2X2([fin61,fin27]) + d;
SSE=[(SP1-p(1,1))^2,(SP2-p(1,2))^2]; %NiX2
for i=2:3; % Numero de predicciones a realizar
    g1=[m1,m2];
    g2=[p(i-1,1),p(i-1,2)];
    p(i,:)=predictor_2X2([g1,g2]) + d;
    SSE=SSE+[(SP1-p(i,1))^2,(SP2-p(i,2))^2]; %calculo del SSE
end
fobj=1*[(m1-m1d1)^2 + (m2-m2d1)^2] + 0.5*SSE(:,2) + 5*SSE(:,1);
```

## DIAGRAMA DE FLUJO DE control\_setup

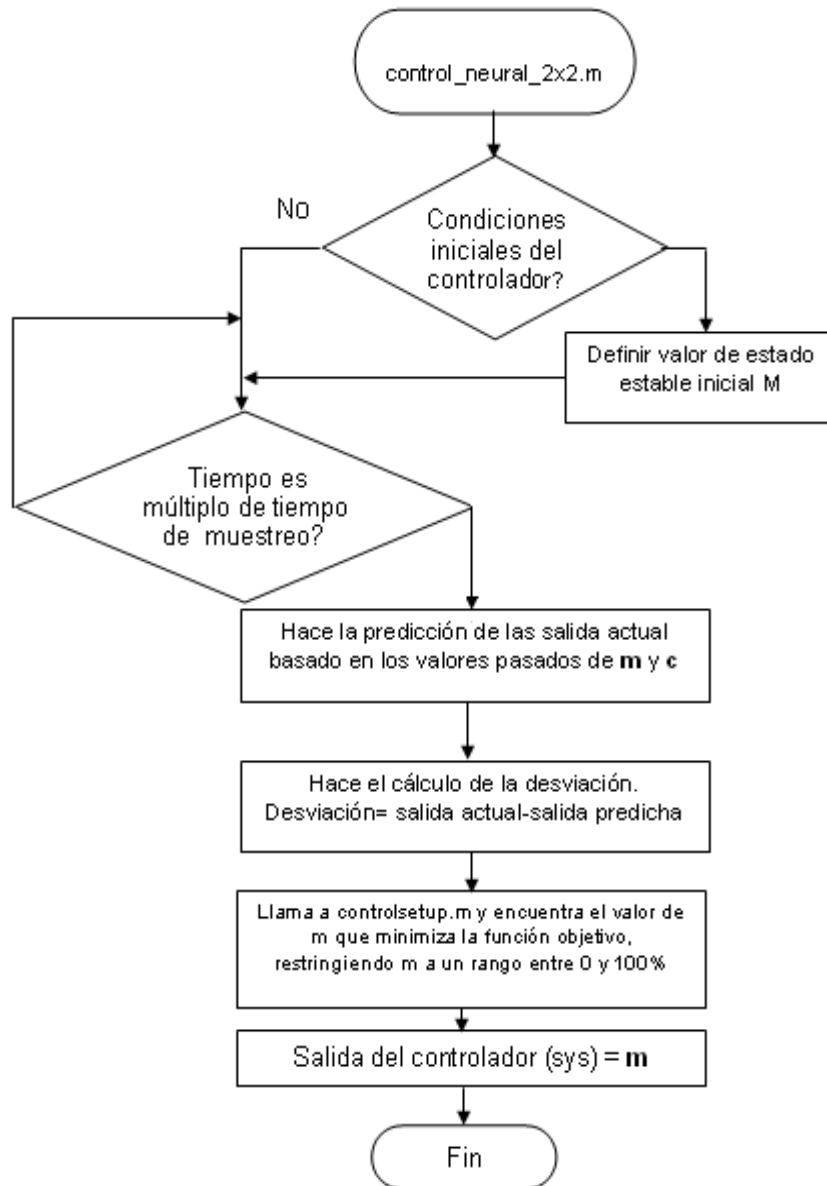


## control\_neural\_2X2.m

```
function [sys, x0] =control_neural_2X2(t,x,u,flag)
global Wh Bh Wo Bo tu M Em i
if nargin~=4
if nargin==0
flag = 0;
else
error('numero errado de argumentos de entrada');
end
end
if flag == 1 % retorna estados continuos
sys = 0;
elseif flag == 3 % retorna las salidas
%u = [m2d1;c1;c1d1;SP1;m1d1;c2;c2d1;SP2]
Ts=2;% Define tiempo de muestreo
if t>=tu+Ts,
u51=[u(5),u(1)]; %[m1d1,m2d2]
u37=[u(3),u(7)]; %[c1d1,c2d2]
TOtnn = predictor_2X2([u51, u37]) % TOtnn es una matriz de NiX2; calcula la prediccion de la salida
e1 = u(2) - TOtnn(:,1); %calcula el error de prediccion: c1-columna 1 de TOtnn
e2 = u(6) - TOtnn(:,2); %calcula el error de prediccion: c2-columna 2 de TOtnn
e=[e1,e2];
fin = [u(1:3);e1;u(4);u(5:7);e2;u(8)]; %fin=[m2d1;c1;c1d1;e1;SP1;m1d1;c2;c2d1;e2;SP2]
M=fmincon(@control_setup_2X2,[66;66],[],[],[],[],[0;0],[100;100],[],optimset('Display','iter','LargeScale','off'),fin);
tu=t
Em(i,:)=e;
i=i+1;
end;
sys=M; %salida de los controladores, ahora es una matriz
elseif flag == 0, %condiciones iniciales
sys=[1,0,2,8,0,1];
%sys=sizes.NumContStates=1;sizes.NumDiscStates=0;sizes.NumOutputs=2;sizes.NumInputs=8;sizes.DirFee
dthrough=0;sizes.NumSampleTimes=1;
x0=66.6667;
M=[66.6667;66.6667]
tu=0;
i=1;
else
sys = [];
end
```

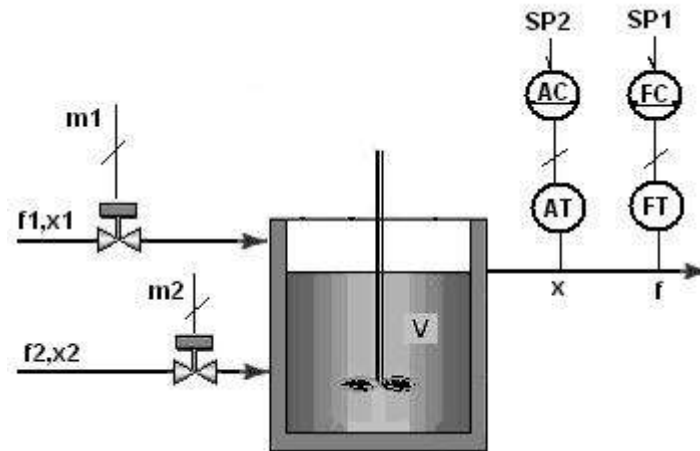


## DIAGRAMA DE FLUJO DE control\_neural.m



## ANEXO B

### Modelamiento del proceso del Tanque de Mezclado



Tanque de mezclado. Control de Flujo y Composición.

El modelo incluye la dinámica de las válvulas y de los transmisores, así como las relaciones hidráulicas para el flujo de salida en el tanque. Se asume mezclado perfecto; además el volumen y las densidades se toman constantes. Las ecuaciones diferenciales son:

- a) Balance de masa en el tanque

$$f_1(t) + f_2(t) - f(t) = \frac{d(\rho V)}{dt} = 0 \quad [1 \text{ ecuación; } 1 \text{ incógnita } \rho];$$

- b) Balance de masa en  $x$ :

$$x_1 f_1(t) + x_2 f_2(t) - x(t) f(t) = \frac{d(\rho V x(t))}{dt} \quad [2 \text{ ecuación; 1 incógnita } x];$$

Las ecuaciones en estado estacionario son:

$$f = f_1 + f_2$$

$$f x = f_1 x_1 + f_2 x_2$$

De las ecuaciones de estado estacionario y con los datos de las condiciones de diseño podemos obtener los valores nominales del sistema que se muestran en la tabla B.1

TABLA B.1

$f_1 = 50 \text{ lb/min}$
$f_2 = 50 \text{ lb/min}$
$f = 100 \text{ lb/min}$
$V = 1000 \text{ lb}$
$x_1 = 10\%$ Fracción de masa
$x_2 = 30\%$ Fracción de masa
$x = 20\%$ Fracción de masa

## Implementación de las válvulas de control

La ecuación de la válvula falla abierta proporcional es:

$$\tau_{V1,2} \frac{df_{1,2}(t)}{dt} + f_{1,2}(t) = K_{V1,2} [100 - m_{1,2}(t)] \quad (\text{Ecuación B.1}) [2]$$

Esta ecuación en estado en estado estacionario tiene la siguiente forma:

$$f_{1,2} = \frac{f_{1,2}^{MAX}}{100} (100 - m_{1,2}) \quad (\text{Ecuación B.2})$$

de la cual podemos obtener el valor de  $m_{1,2}$  dado el valor de flujo máximo de 150lb/min. Para nuestro caso  $m1 = m2 = 66.667$

## Implementación de los sensores / transmisores

La ecuación en estado dinámico de este instrumento de control es:

$$\tau_{T1,2} \frac{dc_{1,2}(t)}{dt} + c_{1,2}(t) = K_{T1,2} [P_{1,2}(t) - VP_{MIN}] \quad (\text{Ecuación B.3}) [2]$$

donde 
$$K_{T1,2} = \frac{100}{SPAN}$$

Las ecuaciones en estado estacionario son:

Para el transmisor de flujo  $\bar{c}1 = 100/150 * \bar{f} - 0$

Para el transmisor analizador  $\bar{c}2 = 100 / (0.35 - 0.06) * \bar{x} - 0.06$

## ANEXO C

### Modelamiento del proceso del Tanque de Calentamiento

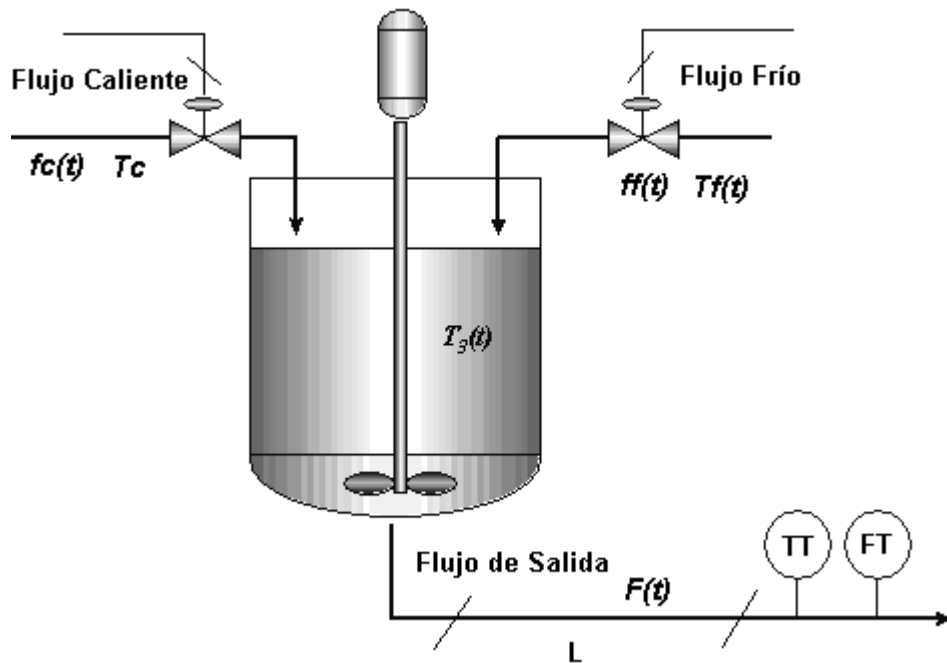


Figura 4.18 Tanque de calentamiento.

Balance de Masas en Estado no Estacionario

$$ff(t) + fc(t) - F(t) = 0 \quad (\text{Ecuación C.1})$$

Balance de Energía en Estado no Estacionario

$$ff(t)T_f(t) + fc(t)T_c - F(t)T_0(t) = V\rho \frac{d}{dt}T_0(t) \quad (\text{Ecuación C.2})$$

## Retardo de Propagación (Transport Delay)

$$T_F(t) = T_0'(t - t_0(t)) \quad (\text{Ecuación C.3})$$

$$t_0(t) = \frac{L A \rho}{F(t)} \quad (\text{Ecuación C.4})$$

Los valores de estado estable del proceso se muestran en la siguiente tabla:

Tabla C.1

$f_f = 120 \text{ lb/min}$
$f_c = 80 \text{ lb/min}$
$T_f = 60^\circ F$
$T_c = 180^\circ F$
$T_F = 108^\circ F$

## Implementación de las válvulas de control

La ecuación de la válvula falla cerrada isoporcentual es:

$$\tau_{V1,2} \frac{df_{1,2}(t)}{dt} + f_{1,2}(t) = f_{MAX} \alpha^{\frac{m(t)}{100}-1} \quad (\text{Ecuación C.5}) [2]$$

Esta ecuación en estado en estado estacionario tiene la siguiente forma:

$$f_{1,2} = f_{MAX} \alpha^{\frac{m}{100}-1} \quad (\text{Ecuación C.6})$$

de la cual podemos obtener el valor de  $m_{1,2}$  dado un valor adecuado de flujo máximo. Para nuestro caso  $f_{MAX} = 600 \text{ lbm/min}$  y  $m1 = m2 = 50$

### Implementación de los sensores / transmisores

La ecuación en estado dinámico del instrumento usado es:

$$\tau_{T1,2} \frac{dc_{1,2}(t)}{dt} + c_{1,2}(t) = K_{T1,2} \sqrt{|P_{1,2}(t) - VP_{MIN}|} \quad (\text{Ecuación C.7}) [2]$$

donde 
$$K_{T1,2} = \frac{100}{SPAN}$$

Las ecuaciones en estado estacionario para los transmisores son:

Transmisor de temperatura:  $\bar{c}_1 = 100/170 * \bar{T} - 30$

Transmisor de flujo:  $\bar{c}_2 = 100/400 * \bar{F} - 0$