

**CENTRO DE INVESTIGACIÓN Y DE ESTUDIOS AVANZADOS  
DEL INSTITUTO POLITÉCNICO NACIONAL**

DEPARTAMENTO DE CONTROL AUTOMÁTICO

## **Modelación y Optimización del Mezclado de Petróleo Crudo con Redes Neuronales**

TESIS QUE PRESENTA EL:  
**Ing. Rubio Avila José de Jesús**

PARA OBTENER EL GRADO DE  
MAESTRO EN CIENCIAS

EN LA ESPECIALIDAD DE  
CONTROL AUTOMÁTICO

DIRECTORES DE TESIS:  
Dr. Wen Yu Liu  
Dra. América Morales Díaz

México, D.F.

Febrero del 2004.

# AGRADECIMIENTOS

Al CONACyT por haberme otorgado una beca económica para la realización de este documento.

Al Departamento de Control Automático del Centro de Investigación y de Estudios Avanzados del Instituto Politécnico Nacional: muy en especial al Departamento de Control Automático por darme la oportunidad de pertenecer al programa de Maestría en Ciencias y por albergarme en estos últimos años, en los cuales se me ha preparado para ser un profesional comprometido con el desarrollo de México. A los profesores por su indeleble labor dentro y fuera de las aulas y a mis condiscipulos con quienes compartí momentos agradables de estudio, de trabajo, de esparcimiento, pero realmente por un compañerismo desinteresado.

A mi asesor Wen Yu Liu por su confianza, por su comprensión, por su apoyo y por brindarme su asesoramiento y dirección para organizar y mejorar el presente trabajo, sin usted no habría podido desarrollar el presente trabajo ya que me guió desde el principio hasta el final, para adquirir y entender mejor los conceptos.

A mi asesora del Instituto Mexicano del Petróleo América Morales Díaz por sus consejos y apoyo moral durante el desarrollo de este trabajo.

A los compañeros tanto contratados como becarios del Departamento en el Instituto Mexicano del Petróleo por brindarme su amistad desinteresada.

A mis ex profesores Moisés Bonilla Estrada, Alexander S. Pozniak Gorbach, Rubén Alejandro Garrido Moctezuma y Ieroham Salomón Barouh por los conocimientos brindados los cuales fueron fundamentales para no partir de cero en el desarrollo de éste documento.

A mis padres Ignacio Rubio Morales y Rosalía Avila Palomo, para quienes sus hijos son la esencia de su vida, su dicha. Quienes consideran que su esfuerzo y sacrificio no ha sido infructuoso. A ustedes que además de privarse de muchas cosas para otorgarlas a nosotros nos han proporcionado algo más importante: su amor, su ternura, su comprensión, e inolvidables momentos de felicidad, nos han hecho sentir sus deseos de superación, su firmeza, su humildad, pero sobre todo a sentirnos comprometidos con nosotros mismos y con mis objetivos. A quienes la ilusión de su existencia ha sido vernos convertidos en personas de provecho, para ellos nuestros logros, tengo la fortuna de ser su hijo.

A mis hermanos Ángel y Alma Sonia con quienes he tenido la dicha de compartir mi vida, los momentos más agradables y los más difíciles, a ustedes que han soportado a veces mi conducta hostil, a ustedes que me han apoyado, por su confianza, por su ejemplo.

A mis amigos Andrés, José y Agustín por nuestra enorme amistad, sincera, leal y desinteresada, por la aventura de habernos conocido, por que siempre han estado para apoyarme en los momentos más difíciles de mi existencia.

A todos mis familiares.

A las personas mencionadas Gracias.

# Índice general

<b>1. Introducción</b>	<b>1</b>
1.1. Motivación del tema de tesis . . . . .	2
1.2. Objetivo de la tesis . . . . .	4
1.3. Contenido de la tesis . . . . .	7
<b>2. Redes neuronales y optimización</b>	<b>11</b>
2.1. Introducción . . . . .	11
2.2. Redes neuronales . . . . .	11
2.2.1. Modelos de las redes neuronales . . . . .	12
2.2.2. Redes neuronales en identificación . . . . .	16
2.2.3. Algoritmo de propagación hacia atrás (Backpropagation) . . . . .	16
2.3. Técnicas de optimización . . . . .	19
2.3.1. Problemas de programación . . . . .	19
2.3.2. Optimización sin restricciones (Definiciones y teoremas) . . . . .	21
2.3.3. Optimización con restricciones de igualdad . . . . .	22
2.3.4. Optimización con restricciones de desigualdad . . . . .	27
<b>3. Modelos matemáticos del proceso de mezclado de petróleo crudo</b>	<b>33</b>
3.1. Introducción . . . . .	33
3.2. Modelos de las propiedades estáticas de mezclado . . . . .	34
3.2.1. Relación de propiedad para sistemas de composición variable . . . . .	34

3.2.2.	Propiedades molares parciales . . . . .	35
3.2.3.	Soluciones ideales . . . . .	36
3.2.4.	Cambio de las propiedades de una mezcla (actividad) . . . . .	37
3.3.	Modelos de las propiedades dinámicas de mezclado . . . . .	40
3.3.1.	Modelo 1. En función de la altura en el tanque . . . . .	40
3.3.2.	Modelo 2. En función de la densidad y del volumen en el tanque . . . . .	45
3.4.	Modelos del mezclado de petróleo crudo . . . . .	48
3.4.1.	Regla de mezclado para petróleo crudo . . . . .	48
3.4.2.	Proceso de mezclado que incluye diferentes tipos de nodos . . . . .	51
3.4.3.	Modelo distribuido . . . . .	53
3.4.4.	Modelo agregado . . . . .	54
<b>4.</b>	<b>Modelación del mezclado de petróleo crudo vía redes neuronales y datos reales</b>	<b>57</b>
4.1.	Introducción . . . . .	57
4.2.	Least square (LS) para identificar los parámetros del mezclado de petróleo crudo . . . . .	58
4.3.	Redes neuronales para identificar los parámetros del mezclado de petróleo crudo	61
4.4.	Obtención de los datos desde una base de datos reales . . . . .	66
4.5.	Modelación del modelo distribuido . . . . .	72
4.5.1.	Identificación de parámetros vía least square . . . . .	72
4.5.2.	Identificación vía redes neuronales para el modelo completo . . . . .	77
4.5.3.	Identificación vía redes neuronales para la parte no lineal . . . . .	84
4.6.	Modelación del modelo agregado . . . . .	90
4.6.1.	Identificación de parámetros vía least square . . . . .	90
4.6.2.	Identificación vía redes neuronales para el modelo completo . . . . .	92
4.6.3.	Identificación vía redes neuronales para la parte no lineal . . . . .	96
4.7.	Conclusión . . . . .	100

<b>5. Optimización de petróleo crudo vía redes neuronales y actualización de datos (bias updating)</b>	<b>103</b>
5.1. Introducción . . . . .	103
5.2. Optimización del mezclado de petróleo crudo con redes neuronales y un esquema de actualización (bias updating) . . . . .	104
5.2.1. Optimización del mezclado petróleo crudo con un esquema de actualización . . . . .	104
5.2.2. Optimización fuera de línea del mezclado de petróleo crudo con redes neuronales . . . . .	107
5.3. Análisis de robustez del controlador con redes neuronales para mezclado . . .	110
5.3.1. Análisis de robustez del controlador . . . . .	110
5.3.2. Análisis de robustez del controlador optimizante con redes neuronales	119
5.4. Simulación de la optimización fuera de línea del mezclado de crudo con redes neuronales . . . . .	120
5.4.1. Obtención de los datos y algoritmo empleado . . . . .	120
5.4.2. Optimización del modelo distribuido . . . . .	123
5.4.3. Optimización del modelo agregado . . . . .	132
5.5. Conclusión . . . . .	139
<b>6. Conclusiones</b>	<b>141</b>
<b>A. Simulación con GUI</b>	<b>145</b>
A.1. Estructura del GUI . . . . .	145
A.1.1. Entrada . . . . .	145
A.1.2. Salida . . . . .	147
A.1.3. Simulador de las redes neuronales y del controlador optimizante . . .	147
<b>B. Método Simplex</b>	<b>151</b>
B.1. Forma estándar de un problema de programación lineal . . . . .	151
B.2. Algoritmo simplex . . . . .	154

B.3. Identificando un punto óptimo . . . . .	155
B.3.1. Mejorando una solución factible básica no óptima . . . . .	155
B.4. Las dos fases del método simplex . . . . .	157
<b>C. Modelos matemáticos del proceso de mezclado de gasolina</b>	<b>161</b>
C.1. Número de octanos . . . . .	161
C.1.1. Método ideal del número de octanos mezclados . . . . .	162
C.1.2. Método Ethil RT-70 . . . . .	162
C.1.3. Método Steward . . . . .	163
C.1.4. Método de Iteración . . . . .	164
C.1.5. Método en exceso . . . . .	166
C.1.6. Método Zahed . . . . .	167
C.2. Presión de Vapor Reid . . . . .	168
C.2.1. Método de Iteración . . . . .	168
C.2.2. Método de mezcla indexada . . . . .	169
C.3. Destilación ASTM . . . . .	169
C.3.1. Método de Iteración . . . . .	169
C.3.2. Modelos empíricos . . . . .	170

# Índice de figuras

1.1. Controlador optimizante comercial para el mezclado de petróleo crudo . . . .	5
1.2. Controlador optimizante con redes neuronales para el mezclado de petróleo crudo . . . . .	6
1.3. Diagrama a bloques de los tópicos del documento . . . . .	9
2.1. Modelo no-lineal de una neurona . . . . .	13
2.2. Perceptrón multicapa . . . . .	15
3.1. Propiedades dinámicas del flujo en el tanque . . . . .	40
3.2. Altura y su derivada en el tanque . . . . .	44
3.3. Densidad y Volumen en el buque tanque . . . . .	47
3.4. Mezcla homogénea de m componentes . . . . .	49
3.5. Etapa de mezclado de la industria petrolera nacional . . . . .	51
3.6. Representación distribuida del proceso de mezclado . . . . .	54
3.7. Representación agregada de la etapa de mezclado . . . . .	55
4.1. Modelos de identificación del mezclado de petróleo crudo . . . . .	58
4.2. Comunicación Excel-Matlab . . . . .	69
4.3. Densidad a la salida del mezclador 1 . . . . .	71
4.4. Densidad de salida del modelo agregado . . . . .	71
4.5. Least square para el mezclador 1 . . . . .	74
4.6. Least square para el mezclador 2 . . . . .	75

4.7. Least square para el mezclador 3 . . . . .	76
4.8. Red neuronal para el modelo completo del mezclador 1 . . . . .	80
4.9. Red neuronal para el modelo completo del mezclador 2 . . . . .	81
4.10. Red neuronal para el modelo completo del mezclador 3 . . . . .	83
4.11. Red neuronal para la no linealidad del mezclador 1 . . . . .	86
4.12. Red neuronal para la no linealidad del mezclador 2 . . . . .	87
4.13. Red neuronal para la no linealidad del mezclador 3 . . . . .	89
4.14. Least square para el modelo agregado . . . . .	91
4.15. Red neuronal para el modelo completo del modelo agregado . . . . .	95
4.16. Red neuronal para la no linealidad del modelo agregado . . . . .	98
5.1. Diagrama de la optimización fuera de línea del mezclado de crudo con nn . . .	108
5.2. Optimización para el modelo completo del Mezclador 1 . . . . .	125
5.3. Optimización para el modelo completo del Mezclador 2 . . . . .	126
5.4. Optimización para el modelo completo del Mezclador 3 . . . . .	128
5.5. Optimización para la no linealidad del Mezclador 1 . . . . .	130
5.6. Optimización para la no linealidad del Mezclador 2 . . . . .	131
5.7. Optimización para la no linealidad del Mezclador 3 . . . . .	133
5.8. Optimización para el modelo completo del Modelo Agregado . . . . .	135
5.9. Optimización para la no linealidad del Modelo Agregado . . . . .	137
A.1. GUI en la opción de la red neuronal para la no linealidad del mezclador 2 . .	149
A.2. GUI en la opción del controlador optimizante para el modelo completo del modelo agregado . . . . .	150

# Capítulo 1

## Introducción

Las operaciones de mezclado son muy comunes en las industrias petroquímicas, cementeras, de refinación, etc. La operación consiste en obtener una mezcla de dos o más componentes con diferentes propiedades a una especificación requerida como son temperatura, densidad, flujo, etc. El control de estas operaciones se considera un punto muy importante dentro de este tipo de industrias, ya que su rentabilidad depende de estas operaciones. Por ejemplo, en la industria petrolera nacional se ha visto que realizar de manera constante las operaciones de mezclado de petróleo crudo con técnicas de control avanzadas podría garantizar condiciones favorables y reportar beneficios económicos muy importantes, como en el caso del crudo ligero de exportación [1].

Es muy común que las propiedades de los componentes a mezclar varíen de una manera indeterminada y que las reglas de mezclado utilizadas contengan términos no lineales sin modelar, llevando a un modelo inadecuado del proceso y como consecuencia que el controlador no encuentre el punto de operación óptimo. Para garantizar las especificaciones requeridas en una operación de mezclado y manejar incertidumbres en el modelo se ha propuesto utilizar un controlador retroalimentado llamado controlador optimizante [1].

## 1.1. Motivación del tema de tesis

El mezclado de petróleo crudo es una operación de optimización basada en una gran cantidad de conocimiento y experiencia. Las propiedades de crudo óptimas son proporcionadas por los modelos de refinería que definen el empleo óptimo de las unidades de torrente para varios tipos de crudo.

Se han desarrollado muchos controladores optimizantes de petróleo crudo basados sobre la técnica de programación lineal [4]. En general, la regla de mezclado es no lineal, un método alternativo es adicionar un término de corrección (incertidumbre) a la regla de mezclado lineal. Un esquema de actualización (bias updating) es una herramienta que resuelve el problema de la optimización en tiempo real, la principal desventaja del esquema de actualización es que requiere de la obtención de las propiedades de mezclado a través de la medición de los datos en línea, pero, algunas veces se necesita la optimización fuera de línea, por ejemplo cuando se espera encontrar la fracción de flujo óptima a la entrada basada sobre datos históricos, el control fuera de línea es útil para supervisión y control.

Se han desarrollado una gran cantidad de algoritmos numéricos para resolver el problema de optimización usando la computadora, tales como el Toolbox de Optimización de Matlab, la principal desventaja de éstos algoritmos es que generalmente convergen lentamente. Un método alternativo es emplear un circuito eléctrico dedicado el cual simule la función objetivo y las restricciones.

Se han propuesto recientemente muchas técnicas de control avanzado para mejorar las propiedades de mezclado de petróleo crudo, todos éstos modelos requieren de un buen modelo de la operación de mezclado, pero es muy complejo manejar analíticamente el modelo matemático exacto del mezclado de petróleo crudo y por lo tanto se hicieron muchos intentos para introducir un modelo simplificado. Un método común de aproximar la operación de mezclado es usar un modelo lineal (ideal) el cual tiene una pequeña incertidumbre ya que en la práctica la mayoría de los modelos de mezclado ideales son no lineales.

Para los ingenieros es muy importante un modelo real para el mezclado de petróleo crudo basado sobre los datos de operación. Resultados recientes muestran que la técnica de redes

neuronales es muy efectiva para modelar gran cantidad de sistemas no lineales cuando no se tiene información completa del modelo, pero, debido a la carencia de un modelo matemático de la planta se tiene la necesidad de probar la estabilidad del aprendizaje del modelo neuro, por esta razón es muy importante tener un resultado teórico que asegure de la estabilidad del modelo neuro.

Foxboro propone un controlador optimizante comercial con un control PID para el mezclado de petróleo crudo, (ver Figura 1.1) [4].

Hay varios problemas para el esquema de control propuesto en la Figura 1.1 entre los cuales destacan los siguientes:

1. Para producir los puntos de operación emplea el algoritmo de programación lineal el cual considera datos lineales, pero el modelo real tiene una no linealidad de los datos que varía entre el 10% y el 20%.
2. El controlador optimizante de Foxboro tiene una única etapa de mezclado, éste tiene que considerar los tres procesos de mezclado en serie de la Figura 1.1 como un solo mezclador.
3. El algoritmo del controlador optimizante de Foxboro es un analizador NMR (de resonancia magnética nuclear) el cual no es flexible y por lo tanto no se puede modificar.

En este proyecto se propone un nuevo esquema de control mostrado en la Figura 1.2.

El controlador optimizante empleado está compuesto por un conjunto de subsistemas, como se muestra en la Figura 1.2 y su forma de operar se describe en los siguientes pasos:

1. Datos. En este paso el controlador toma lecturas de los datos en un intervalo de tiempo para actualizar el modelo que lo representa. Es muy común que los datos se analicen para evitar la presencia de datos erróneos.
2. Modelo. Se pueden emplear el modelo Least Square o el modelo de Redes Neuronales para una estimación del comportamiento del proceso, con esto se evita el usar un modelo matemático ya que el modelo matemático puede introducir mas errores que el emplear uno de los modelo citados [6][8].

3. Actualización del modelo. Los datos obtenidos se utilizan para actualizar los parámetros en el modelo del proceso. Se utilizan técnicas de estimación para los parámetros que no están disponibles, las técnicas son Least Square o Redes Neuronales.
4. Optimización. Una vez actualizado el modelo del proceso. Si se requiere, se utilizan técnicas de estimación para obtener las nuevas referencias que utilizarán los controladores. Los principales algoritmos de optimización utilizados son los algoritmos de programación lineal (LP) y no lineal (NLP) [11][20]. En algunas ocasiones, los resultados de la optimización son examinados antes de enviarlos a los controladores para una operación óptima del proceso.
5. Control. Los resultados de la optimización son utilizados por los controladores para aplicarlos al proceso. Los controladores pueden ser lineales, PI, PID o incluso técnicas avanzadas como el control predictivo [19].

## 1.2. Objetivo de la tesis

El objetivo de esta tesis es mostrar que el proceso de mezclado de petróleo crudo de la industria petrolera nacional se pueden mejorar llevando a cabo el siguiente desarrollo:

1. Realizando un análisis de los modelos distribuido y agregado del proceso de mezclado de petróleo crudo.
2. Realizando la identificación vía redes neuronales para los modelos distribuido y agregado del proceso de mezclado de petróleo crudo.
3. Aplicando un controlador optimizante con redes neuronales al proceso de mezclado de petróleo crudo.

Con los puntos anteriores se pretende encontrar una combinación adecuada de la materia prima a utilizar tal que se produzca la mezcla de interés a un costo mínimo y al mismo

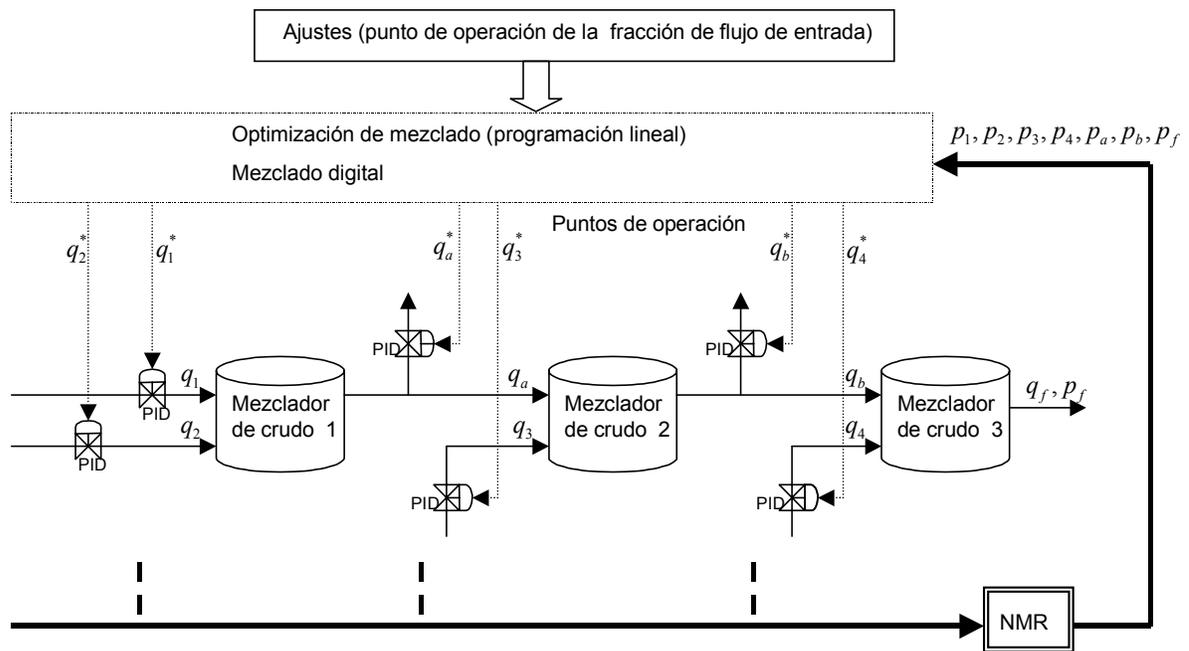


Figura 1.1: Controlador optimizante comercial para el mezclado de petróleo crudo

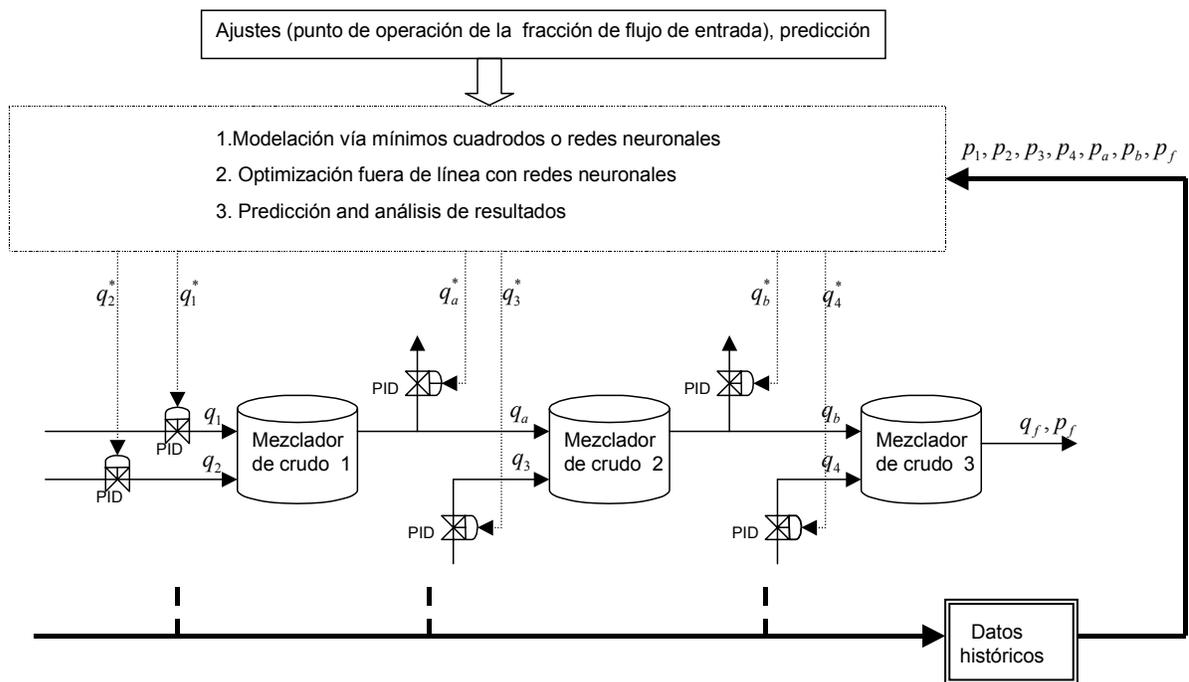


Figura 1.2: Controlador optimizante con redes neuronales para el mezclado de petróleo crudo

tiempo, que cumpla con las condiciones de calidad requeridas, junto con las restricciones físicas del proceso.

### 1.3. Contenido de la tesis

- En este capítulo se presenta en que consisten las operaciones de mezclado y su importancia dentro de las industrias. Se describe la estructura y la forma típica de operar de los controladores que comúnmente se utilizan para el control de las operaciones de mezclado.
- El capítulo 2 presenta un panorama general de las redes neuronales como la descripción de los modelos más comunes de las redes neuronales, las redes neuronales en identificación y el algoritmo de Backpropagation que es el algoritmo mayormente empleado en identificación y también es el más simple [8][9]. Posteriormente se presenta un panorama de la optimización en donde lo más importante son la optimización con restricciones de igualdad donde se describe la solución por el método de multiplicadores de Lagrange y la optimización con restricciones de desigualdad donde se describen las condiciones de Kuhn-Tucker [20].
- En el capítulo 3 se presentan los modelos de la propiedades estáticas del mezclado de petróleo crudo [18]. Mas adelante se presentan dos modelos para el tanque, el modelo dependiente de la altura y el modelo dependiente de la densidad y del volumen en el tanque [1][25]. Posteriormente se analizan los procesos de mezclado que contienen más de un nodo de mezclado, para los cuales se utilizan el modelo distribuido y el modelo agregado [1]. En el modelo distribuido, el modelo de mezclado se representa por puntos de mezclado independientes con un controlador para cada uno, teniendo solo información local del proceso. En el modelo agregado, el proceso de mezclado se representa como un solo punto de mezclado utilizando solo un controlador el cual tiene información global del proceso. Al final se presenta como se modelan cada uno de estos modelos los cuales representan en el proceso de mezclado.

- En el capítulo 4 se presenta como se obtienen los datos reales que se tienen una base de datos reales en un programa de Excel, se establece la comunicación para extraer los datos de Excel con una sentencia de Matlab [22], posteriormente estos datos se emplean para desarrollar la estimación del comportamiento del proceso de mezclado para el modelo agregado y para el modelo distribuido, vía Least Square, vía Redes neuronales y se presentan los resultados para cada uno de los casos, por último se presenta una tabla comparativa que muestra los resultados para cada caso.
- En el capítulo 5 se presenta como se aplica el controlador optimizante con redes neuronales al proceso de mezclado, para ello en primer lugar se demuestra la robustez del controlador optimizante [12], después se demuestra la robustez del controlador optimizante con redes neuronales [26], posteriormente se realiza una comparación entre el controlador optimizante que opera fuera de línea al cual se le introducen datos reales y el controlador optimizante con redes neuronales que opera fuera de línea, por último se presenta una tabla comparativa que muestra los resultados para cada caso.
- En el capítulo 6 se presentan las conclusiones finales.

En la Figura 1.3 se muestra un diagrama a bloques de la relación entre los tópicos de este documento.

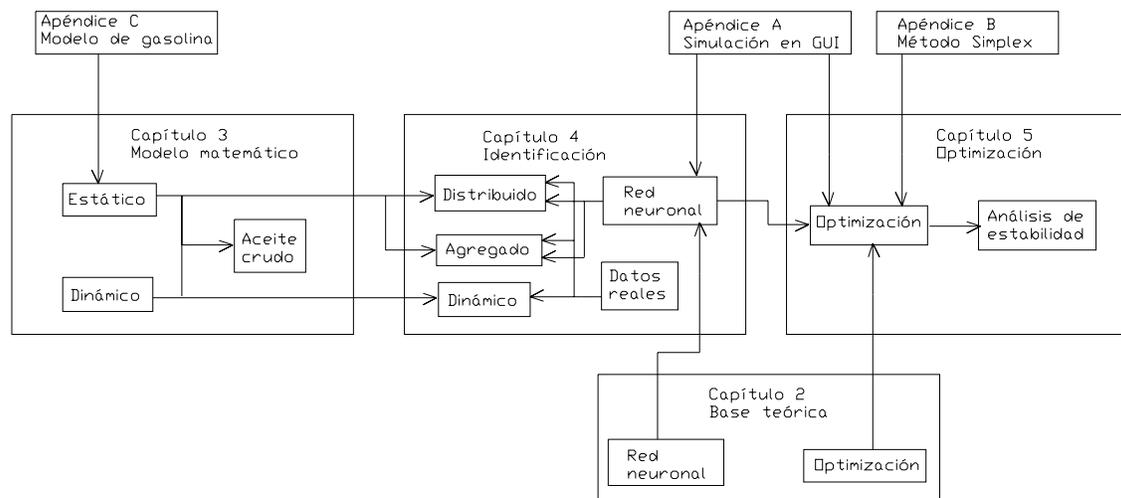


Figura 1.3: Diagrama a bloques de los tópicos del documento



# Capítulo 2

## Redes neuronales y optimización

### 2.1. Introducción

Este capítulo presenta un panorama general de las redes neuronales como la descripción de los modelos más comunes de las redes neuronales, las redes neuronales en identificación y el algoritmo de Backpropagation que es el algoritmo mayormente empleado en identificación y también es el más simple [8][9]. Posteriormente se presenta un panorama de la optimización en donde lo más importante son la optimización con restricciones de igualdad donde se describe la solución por el método de multiplicadores de Lagrange y la optimización con restricciones de desigualdad donde se describen las condiciones de Kuhn-Tucker [20].

### 2.2. Redes neuronales

Un sistema de control tiene la habilidad de aprender si adquiere la información durante la operación, acerca de comportamientos desconocidos de la planta y su ambiente tal que se mejora la ejecución completa. Con este requerimiento del controlador, es posible expandir la región de operación y finalmente la implementación de sistemas autónomos. Una clase de modelos que tiene la potencialidad de implementar este aprendizaje son las redes neuronales artificiales. Ciertamente, la morfología neuronal del sistema nervioso es mucho más compleja,

no obstante, se puede desarrollar una analogía simplificada, la cual se podría utilizar en aplicaciones de ingeniería. Basándose en esta analogía, se pueden desarrollar las estructuras de las redes neuronales artificiales [8].

### 2.2.1. Modelos de las redes neuronales

Una red neuronal artificial (RNA) es un elemento capaz de procesar gran cantidad de información de forma paralela y distribuida, inspirada de las redes neuronales biológicas, las cuales pueden almacenar conocimiento y tenerlo disponible para su uso [13]. Esta tiene algunas similitudes con el cerebro, como lo son:

- El conocimiento es adquirido a través del proceso de aprendizaje.
- La conectividad entre neuronas es llamada pesos sinópticos y son utilizados para almacenar el conocimiento.

La función del proceso de aprendizaje es modificar los pesos sinópticos de las redes neuronales con el objetivo de minimizar una función de costo. La modificación de los pesos sinópticos es el método tradicional para el diseño e implementación de las redes neuronales.

La neurona, es la unidad fundamental para la operación de la red neuronal. La Figura 2.1 muestra el esquema de una red neuronal, el cual esta compuesto por:

1. Un conjunto de uniones sinópticas, con cada elemento caracterizado por su propio peso.
2. Un sumador el cual suma los componentes de la señal de entrada, multiplicados por su respectivo peso sinóptico.
3. Una función de activación no-lineal que transforma la salida del sumador en la entrada de la neurona siguiente .

Al esquema de la red neuronal también se aplica un umbral para reducir la entrada a la función de activación. En términos matemáticos, la  $i$ -ésima neurona se puede describir como:

$$u_i = \sum_{j=1}^n \omega_{ij} x_j$$

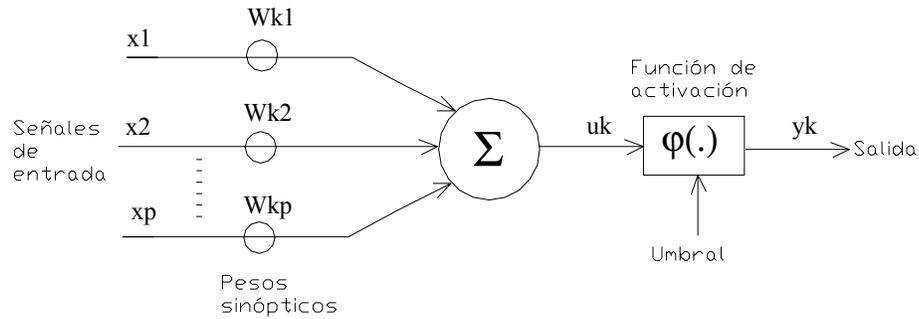


Figura 2.1: Modelo no-lineal de una neurona

$$y_i = \rho(u_i - \rho_i) \quad (2.1)$$

donde:

$x_j$  j-ésimo componente de la entrada.

$\omega_{ij}$  peso de la conexión entre la j-ésima componente de la entrada y la i-ésima neurona.

$u_i$  salida del sumador.

$\rho_i$  umbral.

$\rho(\cdot)$  función de activación no lineal.

$y_i$  salida de la i-ésima neurona.

La función de activación no-lineal denotada por  $g(\cdot)$ , recibe como entrada  $x_i$  y genera el elemento de la salida  $y_i$  como se describe en la siguiente ecuación:

$$y_i = g(x_i) \quad (2.2)$$

Una clasificación de este tipo de función es:

1. Diferenciable y no diferenciable.

2. Tipo pulso y tipo escalón.
3. Positiva y de promedio cero.

La clasificación 1 se distingue por tener funciones suaves y discontinuas. Las funciones suaves son necesarias para algunos algoritmos de adaptación como el de propagación hacia atrás, mientras que las funciones discontinuas (por ejemplo las funciones de umbral) se emplean para generar una salida binaria.

La clasificación 2 se distingue por tener funciones, las cuales solo tienen un valor significativo de la salida cuando las entradas están cerca del cero, dado que las funciones solo cambian significativamente alrededor del cero.

La clasificación 3 se refiere a las funciones tipo escalón. Las funciones positivas que cambian de 0 en  $-\infty$  a 1 en  $\infty$ . Las funciones de promedio cero cambian de -1 en  $-\infty$  a 1 en  $\infty$ .

La forma como se interconectan las neuronas de una red neuronal determina su estructura. Para propósitos de identificación y control, las estructuras más usadas son:

1. Redes con conexión hacia adelante de una capa.
2. Redes con conexión hacia adelante multicapa.
3. Redes de funciones radiales básicas.

Las redes neuronales anteriores se les conoce como redes neuronales con conexión hacia adelante o redes neuronales estáticas.

### **Redes neuronales con conexión hacia adelante multicapa**

Estas se distinguen por la presencia de una o más capas ocultas, (ver Figura 2.2), cuyos nodos computacionales se llaman neuronas ocultas. Típicamente, las neuronas en cada capa tienen como señales de entrada, las señales de salida de la capa precedente. Si cada neurona en cada capa es conectada con todas las neuronas de las capas adyacentes, entonces la

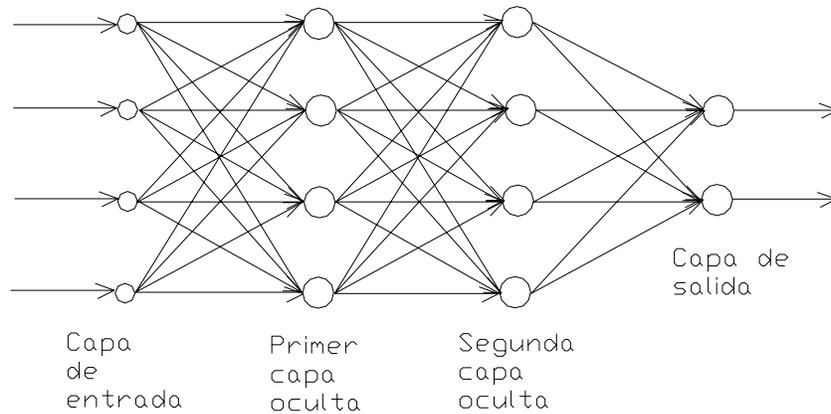


Figura 2.2: Perceptrón multicapa

red neuronal es llamada totalmente conectada, en el caso opuesto es llamada parcialmente conectada [8][9].

El perceptrón multicapa tiene las siguientes características:

1. La función de activación para cada neurona es suave a diferencia a la de límites extremos usada en el perceptrón de una sola capa. Comúnmente, la función no-lineal empleada es el sigmoide, definida como: (ver Figura 2.2)

$$e_i(v_i) = \frac{1}{1 + e^{-v_i}}$$

2. La red esta compuesta por una o más capas de neuronas ocultas.
3. La red presenta un alto grado de conectividad.

El perceptrón multicapa obtiene su poder computacional a través de la combinación de estas características y su habilidad de aprender de la experiencia. Sin embargo, la presencia de no-linealidades distribuidas y la alta conectividad de la red complican el análisis teórico.

### 2.2.2. Redes neuronales en identificación

Las redes neuronales tienen el potencial de ser aplicadas para modelar sistemas no-lineales. Una pregunta importante es si el sistema es identificable, i.e., si la red neuronal puede modelar el comportamiento del sistema sin ser necesario dar un modelo estructurado particular. La identificabilidad en redes neuronales está relacionada a la unicidad de los pesos y si dos redes con diferentes parámetros pueden producir un comportamiento idéntico entrada/salida [13].

Para representar los sistemas no-lineales con redes neuronales, una aproximación correcta es incrementar las entradas de las redes con señales correspondientes a sus entradas y salidas. Asumiendo que el sistema no-lineal está descrito por:

$$y(k+1) = g[y(k), y(k+1), \dots, y(k-n); u(k), u(k+1), \dots, u(k-m)] \quad (2.3)$$

$$y, u \in \mathfrak{R}, m \leq n$$

Este modelo no considera las perturbaciones directamente.

Una aproximación obvia del método del sistema es seleccionar la estructura de entrada/salida de la red neuronal para que sea la misma que tiene el sistema.

### 2.2.3. Algoritmo de propagación hacia atrás (Backpropagation)

El algoritmo de aprendizaje usado para ajustar los pesos sinópticos del perceptrón multicapa estático es conocido como backpropagation. El algoritmo provee un método eficiente para el entrenamiento del perceptrón multi-capas [8][9].

El error en la salida de la  $j$ -ésima neurona de la capa de salida, está dado por:

$$e_j(k) = d_j(k) - y_j(k) \quad (2.4)$$

donde:

$d_j$  es la salida deseada.

$y_j$  es la neurona de salida.

$k$  indica la  $k$ -ésima muestra.

La suma instantánea del cuadrado de los errores de salida está dado por:

$$\varepsilon(k) = \frac{1}{2} \sum_{j=1}^l e_j^2(k) \quad (2.5)$$

donde  $l$  es el número de neuronas de la capa de salida.

El error medio cuadrático se obtiene al sumar  $\varepsilon(k)$  para todas las muestras (en una iteración) y normalizar con respecto al tamaño de la iteración.

$$\varepsilon(k) = \frac{1}{N} \sum_{k=1}^N \varepsilon(k) \quad (2.6)$$

con  $N$  como el número de muestras, que forman cada iteración.

Usando el algoritmo del gradiente descendente, los pesos de las conexiones entre  $i$ -ésima neurona y la  $j$ -ésima neurona se actualiza como:

$$\Delta\omega_{ji}(k) = \omega_{ji}(k+1) - \omega_{ji}(k) = -\eta \frac{\partial \varepsilon(k)}{\partial \omega_{ji}(k)} \quad (2.7)$$

La corrección que realiza  $\Delta\omega_{ji}(k)$  es conocida como la regla delta, donde el término  $\frac{\partial \varepsilon(k)}{\partial \omega_{ji}(k)}$  se puede calcular como:

$$\frac{\partial \varepsilon(k)}{\partial \omega_{ji}(k)} = \frac{\partial \varepsilon(k)}{\partial e_j(k)} \frac{\partial e_j(k)}{\partial y_j(k)} \frac{\partial y_j(k)}{\partial v_j(k)} \frac{\partial v_j(k)}{\partial \omega_{ji}(k)} \quad (2.8)$$

Las derivadas parciales están dadas por:

$$\begin{aligned} \frac{\partial \varepsilon(k)}{\partial e_j(k)} &= e_j(k) & \frac{\partial e_j(k)}{\partial y_i(k)} &= -1 \\ \frac{\partial y_j(k)}{\partial v_j(k)} &= \varrho_j'(v_j(k)) & \text{con } \varrho_j'(v_j(k)) &= \frac{\partial \varrho_j(v_j(k))}{\partial v_j(k)} \\ & & \frac{\partial v_j(k)}{\partial \omega_{ji}(k)} &= y_i(k) \end{aligned} \quad (2.9)$$

Así, la regla delta puede ser reescrita como:

$$\Delta\omega_{ji}(k) = \eta \delta_j(k) y_i(k) \quad (2.10)$$

donde el gradiente local  $\delta_j(k)$  se define de la siguiente manera:

$$\delta_j(k) = \frac{\partial e_j(k)}{\partial y_j(k)} \frac{\partial y_j(k)}{\partial v_j(k)} = e_j(k) \rho_j'(v_j(k)) \quad (2.11)$$

Para el gradiente local  $\delta_j(k)$  se distinguen dos casos: la  $j$ -ésima neurona está en la capa de salida o se encuentra en una capa oculta.

Caso1: Si la  $j$ -ésima neurona está localizada en la capa de salida es correcto calcular  $\delta_j(k)$  como se define en la ecuación (2.11).

Caso 2: Si la  $j$ -ésima neurona está localizada en la capa oculta, no hay una respuesta deseada para esta neurona. Así, la señal del error se tiene que propagar desde la salida hacia la neurona de la capa oculta donde se requiera el valor del gradiente local  $\delta_j(k)$ . En este caso es posible establecer la siguiente ecuación:

$$\delta_j(k) = \frac{\partial y_j(k)}{\partial v_j(k)} \sum_{n=1}^m \delta_n(k) \omega_{nj}(k) \quad (2.12)$$

donde  $n$  indica la conexión de la  $n$ -ésima a la  $j$ -ésima neurona y  $m$  es el número total de estas neuronas.

Este algoritmo se ha convertido en el más popular para el entrenamiento del perceptrón multicapa. Es muy eficiente y tiene la capacidad de clasificar información que debe ser separada de forma no-lineal. El algoritmo es una técnica de gradiente que implementa la minimización en una sola dirección, la cual puede ser la óptima, debido a esto, no es posible demostrar la convergencia del algoritmo a un mínimo global.

Las redes neuronales permiten obtener un mapeo entrada-salida del comportamiento del sistema, con lo cual no es necesario obtener un modelo matemático del mismo, esto, puede ser una ventaja al tratar con sistemas muy complejos de los cuales es muy difícil obtener su modelo matemático. Por otra parte las redes neuronales nos permiten implementar un controlador para un sistema a partir de la información obtenida de la red.

## 2.3. Técnicas de optimización

La optimización es el acto de obtener el mejor resultado bajo determinadas circunstancias. La meta final de todas las decisiones es minimizar el esfuerzo requerido. La minimización se define como encontrar las condiciones que nos dan el máximo o el mínimo de un valor de una función. No existe un método individual para resolver por completo el problema de la optimización de manera eficiente. Se han desarrollado diferentes métodos para resolver diferentes problemas de optimización. Existen tres métodos de programación los cuales son: técnicas de programación matemática, técnicas de procesos estocásticos y métodos estadísticos. Para el propósito de este desarrollo solo interesan las técnicas de programación matemática y en especial, la programación lineal. Las técnicas de programación matemática son útiles para encontrar el mínimo de una función de varias variables bajo un espacio prescrito de restricciones [20].

### 2.3.1. Problemas de programación

Un problema de control óptimo (OC) usualmente describe mediante dos tipos de variables, nombradas, variables de control (diseño) y las variables de estado. Las variables de control gobiernan la evolución del sistema desde una etapa a la siguiente y las variables de estado describen el comportamiento del sistema en cualquier etapa. Explícitamente, el problema de control óptimo es un problema de programación matemática envolviendo un número de etapas, donde cada etapa esta envuelta desde la etapa previa en una manera prescrita, el problema es encontrar un espacio de variables de control o variables de diseño tales que se minimice la función objetivo total sobre un número de etapas sujeto a ciertas restricciones sobre las variables de control y las variables de estado [20]. Este se puede

enunciar como sigue:

$$\left\{ \begin{array}{l} \text{Encontrar } X \text{ el cual minimize:} \\ f(X) = \sum_{i=1}^l f_i(x_i, y_i) \\ \text{sujeto a las siguientes restricciones:} \\ q_i(x_i, y_i) + y_i = y_{i+1}, \quad i = 1, 2, \dots, l \\ g_j(x_j) \leq 0, \quad j = 1, 2, \dots, l \\ h_k(x_k) \leq 0, \quad k = 1, 2, \dots, l \end{array} \right. \quad (2.13)$$

donde  $x_i$  es la  $i$ -ésima variable de control,  $y_i$  es la  $i$ -ésima variable de estado y  $f_i$  es la contribución de la  $i$ -ésima etapa para la función objetivo total;  $g_i$ ,  $h_k$  y  $q_i$  son funciones de  $x_j$ ,  $y_k$ ,  $x_i$  y  $y_i$  respectivamente.

### Problema de programación lineal

Si la función objetivo y todas las restricciones en ecuación (2.13) son funciones lineales de las variables de control, al problema de programación matemática se le llama problema de programación lineal (LP). Un problema de programación lineal se enuncia de la siguiente forma estándar:

Encontrar:

$$X = \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix}$$

El cual minimize la función:

$$f(X) = \sum_{i=1}^n c_i x_i \quad (2.14)$$

sujeto a las siguientes restricciones:

$$\sum_{i=1}^n a_{jk} x_k = b_j, \quad j = 1, 2, \dots, m$$

y

$$x_i \geq 0, \quad i = 1, 2, \dots, n$$

donde  $c_i$ ,  $a_{ij}$  y  $b_j$  son constantes.

### Problema de programación no lineal

Si cualquiera de las funciones entre las funciones objetivo y las funciones de restricción en la ecuación (2.13) es no lineal, a este se le llama problema de la programación no lineal (NLP). Este es el problema de programación más general y todos los demás problemas se pueden considerar casos especiales del problema NLP.

Los métodos clásicos de optimización son útiles para encontrar el óptimo de funciones diferenciales y continuas. Estos métodos son analíticos y hacen uso de las técnicas de los cálculos diferenciales en la localización de los puntos óptimos. En vista de que algunos problemas prácticos envuelven funciones objetivo que no son continuas ni diferenciables, las técnicas de optimización clásica están limitadas por las aplicaciones prácticas.

#### 2.3.2. Optimización sin restricciones (Definiciones y teoremas)

**La r-ésima diferencial de  $f$ .** Si todas las derivadas parciales de la función  $f$  de orden completo  $r \geq 1$  existen y son continuas en el punto  $X^*$ , entonces el polinomio:

$$d^r f(X^*) = \sum_{i=1}^n \sum_{j=1}^n \cdots \sum_{k=1}^n h_i h_j \cdots h_k \frac{\partial^r f(X^*)}{\partial x_i \partial x_j \cdots \partial x_k} \quad (2.15)$$

se le llama la r-ésima diferencial de  $X^*$ . Note que hay  $r$  sumatorias y un  $h_i$  esta asociado con cada sumatoria en la ecuación (2.15).

**Teorema 2.1** Si  $f(X)$  tiene un punto extremo (un mínimo o un máximo) en  $X = X^*$  y si las primeras derivadas parciales de  $f(X)$  existen en  $X^*$ , entonces:

$$\frac{\partial f}{\partial x_1}(X^*) = \frac{\partial f}{\partial x_2}(X^*) = \cdots = \frac{\partial f}{\partial x_n}(X^*) \quad (2.16)$$

**Teorema 2.2** Una condición suficiente para que el punto estacionario  $X^*$  sea un punto extremo es que la matriz de las segundas derivadas parciales (matriz Hessian) de  $f(x)$  evaluada en  $X^*$  sea: (i) positiva definida cuando  $X^*$  es un punto mínimo, y (ii) negativa definida cuando  $X^*$  es un punto máximo.

Nota: Una matriz  $A$  será positiva definida si todos sus eigenvalores son positivos, i.e. todos los valores de  $\lambda$  los cuales satisfagan el determinante:

$$|A - \lambda I| = 0$$

deben ser positivos.

Otro examen que se puede aplicar para encontrar si la matriz  $A$  de orden  $n$  es positiva definida es la evaluación de los determinantes:

$$A_1 = \begin{vmatrix} a_{11} \end{vmatrix}, A_2 = \begin{vmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{vmatrix}, A_3 = \begin{vmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{vmatrix}, \dots,$$

$$A_n = \begin{vmatrix} a_{11} & a_{12} & a_{13} & \dots & a_{1n} \\ a_{21} & a_{22} & a_{23} & \dots & a_{2n} \\ a_{31} & a_{32} & a_{33} & \dots & a_{3n} \\ \vdots & & & & \\ a_{n1} & a_{n2} & a_{n3} & \dots & a_{nn} \end{vmatrix}$$

La matriz  $A$  será positiva definida si y solamente si todos los valores  $A_1, A_2, \dots, A_n$  son positivos. La matriz  $A$  será negativa definida si el signo de  $A_j$  es  $(-1)^j$  para  $j = 1, 2, \dots, n$ . Si algunos valores son positivos y algunos son cero, la matriz  $A$  será positiva semidefinida.

### 2.3.3. Optimización con restricciones de igualdad

Se considera la optimización de funciones continuas sujetas a las siguientes restricciones:

$$\text{minimiza } f = f(X) \text{ sujeto a } g_i(X) = 0, \quad j = 1, 2, \dots, m \quad (2.17)$$

donde:

$$X = \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix}$$

Aquí,  $m \leq n$ ; en el otro caso (si  $m > n$ ), el problema se convierte en sobre-definido, y, en general, no habrá solución. Hay varios métodos para solucionar este problema, los métodos de la sustitución directa, variación restringida y multiplicadores de Lagrange son algunos de ellos [20].

### **Solución por el método de sustitución directa**

En el problema no restringido, hay  $n$  variables independientes, y la función objetivo se puede evaluar para cualquier espacio de  $n$  números. De cualquier manera, en el problema restringido, al menos una variable independiente pierde su arbitrariedad con la adición de cada restricción de igualdad. Así, el problema con  $m$  restricciones en  $n$  variables tendrá solamente  $(n - m)$  variables independientes. Si se selecciona cualquier espacio de  $n - m$  variables, los valores de las variables restantes están determinados por  $m$  restricciones de igualdad.

Así, teóricamente es posible resolver simultáneamente las  $m$  restricciones de igualdad y expresar cualquiera  $m$  variables en términos de las  $n - m$  variables restantes. Cuando se substituyen estas expresiones dentro de la función objetivo original, resulta una nueva función objetivo que considera solamente  $n - m$  variables. La nueva función objetivo no esta sujeta a ninguna restricción y por lo tanto se puede encontrar su óptimo con las técnicas de optimización no restringidas clásicas.

Este método de la sustitución directa, aunque parece en teoría simple, no es conveniente desde el punto de vista práctico. La razón es que las restricciones serán no lineales para la mayoría de los problemas prácticos y se hace imposible el resolver el expresar  $m$  variables en términos de las  $n - m$  variable restantes.

### Solución por el método de variaciones restringidas

La idea básica empleada en el método de variaciones restringidas es encontrar una expresión en forma cerrada para la diferencial de primer orden de  $f(df)$  de todos los puntos en los cuales se satisfacen las restricciones  $g_j(X) = 0$ ,  $j = 1, 2, \dots, m$ . Los puntos óptimos deseados se obtienen por la solución de la ecuación diferencial en cero.

Un problema con  $n = 2$  y  $m = 1$ . Considere el problema:

Minimizar:

$$f(x_1, x_2) \quad (2.18)$$

Sujeto a:

$$g(x_1, x_2) = 0 \quad (2.19)$$

Para resolver la restricción  $g(x_1, x_2) = 0$  obtenemos  $x_2$  como:

$$x_2 = h(x_1) \quad (2.20)$$

Por la sustitución de (2.20), la solución objetivo se convierte en una función de una sola variable como  $f = f[x_1, h(x_1)]$ . Una condición necesaria para que  $f$  tenga un mínimo en algún punto  $(x_1^*, x_2^*)$  es que la derivada total de  $f(x_1, x_2)$  con respecto a  $x_1$  sea cero en  $(x_1^*, x_2^*)$ . La ecuación diferencial total  $f(x_1, x_2)$  se puede escribir como sigue

$$df(x_1, x_2) = \frac{\partial f}{\partial x_1} dx_1 + \frac{\partial f}{\partial x_2} dx_2$$

Cuando esta ecuación es igual a cero nosotros obtenemos:

$$df(x_1, x_2) = \frac{\partial f}{\partial x_1} dx_1 + \frac{\partial f}{\partial x_2} dx_2 = 0 \quad (2.21)$$

Desde que  $g(x_1^*, x_2^*) = 0$  en el punto mínimo, cualquier variación  $dx_1$  y  $dx_2$  que se tome alrededor del punto  $(x_1^*, x_2^*)$  se llama variación admisible proporcionada por la relación:

$$g(x_1^* + dx_1, x_2^* + dx_2) = 0 \quad (2.22)$$

La expansión en series de Taylor de la función en la ecuación (2.22) alrededor del punto  $(x_1^*, x_2^*)$  es:

$$g(x_1^* + dx_1, x_2^* + dx_2) \simeq g(x_1^*, x_2^*) + \frac{\partial g}{\partial x_1}(x_1^*, x_2^*) dx_1 + \frac{\partial g}{\partial x_2}(x_1^*, x_2^*) dx_2 = 0 \quad (2.23)$$

donde se consideran  $dx_1$  y  $dx_2$  pequeñas. Desde que  $g(x_1^*, x_2^*) = 0$ , la ecuación (2.23) se reduce a:

$$\frac{\partial g}{\partial x_1} dx_1 + \frac{\partial g}{\partial x_2} dx_2 = 0 \text{ en } (x_1^*, x_2^*) \quad (2.24)$$

Así, la ecuación (2.24) se debe satisfacer para todas las variaciones admisibles.

Considerando que, la ecuación (2.24) se escribe como:

$$dx_2 = -\frac{(\partial g/\partial x_1)}{(\partial g/\partial x_2)}(x_1^*, x_2^*) dx_1 = 0 \quad (2.25)$$

Esta relación nos indica que una vez que se escoge una variación arbitraria en  $x_1(dx_1)$ , la variación  $x_2(dx_2)$  de fija automáticamente desde que  $dx_1$  y  $dx_2$  son las únicas variaciones admisibles. Por la sustitución de la ecuación (2.25) en ecuación (2.21) se tiene:

$$df = \left[ \frac{\partial f}{\partial x_1} - \frac{(\partial g/\partial x_1)}{(\partial g/\partial x_2)} \frac{\partial f}{\partial x_2} \right] \Big|_{(x_1^*, x_2^*)} dx_1 = 0 \quad (2.26)$$

Note que la ecuación (2.26) se tiene que satisfacer para todos los valores de  $dx_1$ . Desde que  $x_1$  se puede elegir arbitrariamente, la ecuación (2.26) es equivalente a:

$$\left( \frac{\partial f}{\partial x_1} \frac{\partial g}{\partial x_2} - \frac{\partial f}{\partial x_2} \frac{\partial g}{\partial x_1} \right) \Big|_{(x_1^*, x_2^*)} = 0 \quad (2.27)$$

La ecuación (2.27) da las condiciones necesarias para tener un punto extremo ( $x_1^*, x_2^*$ ) (mínimo o máximo).

### Solución por el método de Multiplicadores de Lagrange

En el método de multiplicadores de Lagrange, se introduce una variable adicional por cada restricción. Así, si el problema tiene  $n$  variables y  $m$  restricciones de igualdad, se adicionan  $m$  variables al problema, así que el número final cambia de  $n$  incógnitas a  $n + m$ . De hecho, hay algunas simplificaciones disponibles por las adición de las nuevas variables [20].

Considerar el problema de:

$$\text{minimizar } f(x_1, x_2) \text{ sujeto a } g(x_1, x_2) = 0 \quad (2.28)$$

Este problema se consideró para el caso anterior y se encontró que una condición necesaria para la existencia de un punto extremo en  $X = X^*$  es:

$$\left( \frac{\partial f}{\partial x_1} - \frac{(\partial f / \partial x_2)}{(\partial g / \partial x_2)} \frac{\partial g}{\partial x_1} \right) \Big|_{(x_1^*, x_2^*)} = 0 \quad (2.29)$$

Definiendo el multiplicador de Lagrange  $\lambda$  como:

$$\lambda = - \left( \frac{\partial f / \partial x_2}{\partial g / \partial x_2} \right) \Big|_{(x_1^*, x_2^*)} \quad (2.30)$$

La ecuación (2.29) se puede escribir como:

$$\left( \frac{\partial f}{\partial x_1} + \lambda \frac{\partial g}{\partial x_1} \right) \Big|_{(x_1^*, x_2^*)} = 0 \quad (2.31)$$

y la ecuación (2.30) se puede escribir como:

$$\left( \frac{\partial f}{\partial x_2} + \lambda \frac{\partial g}{\partial x_2} \right) \Big|_{(x_1^*, x_2^*)} = 0 \quad (2.32)$$

En adición, se satisface la ecuación de restricción en el punto extremo, i.e.:

$$g(x_1, x_2) \Big|_{(x_1^*, x_2^*)} = 0 \quad (2.33)$$

Así, las ecuaciones (2.31) a la (2.33) representan las condiciones necesarias para que el punto sea un punto extremo.

Las condiciones necesarias dadas por las ecuaciones (2.31) a la (2.33) también se pueden generar por la construcción de una función  $L$ , conocida como la función de Lagrange, como:

$$L(x_1, x_2, \lambda) = f(x_1, x_2) + \lambda g(x_1, x_2) \quad (2.34)$$

Si las derivadas parciales de la función de Lagrange con respecto a cada uno de los argumentos es igual a cero, se obtienen las condiciones necesarias dadas por las ecuaciones (2.31) a (2.33).

Así, se tienen las siguientes ecuaciones:

$$\left\{ \begin{array}{l} \frac{\partial L}{\partial x_1}(x_1, x_2, \lambda) = \frac{\partial f}{\partial x_1}(x_1, x_2) + \lambda \frac{\partial g}{\partial x_1}(x_1, x_2) \\ \frac{\partial L}{\partial x_2}(x_1, x_2, \lambda) = \frac{\partial f}{\partial x_2}(x_1, x_2) + \lambda \frac{\partial g}{\partial x_2}(x_1, x_2) \\ \frac{\partial L}{\partial \lambda}(x_1, x_2, \lambda) = g(x_1, x_2) = 0 \end{array} \right\} \quad (2.35)$$

Las condiciones necesarias deducidas anteriormente se pueden extender al problema general con  $n$  variables y  $m$  restricciones de igualdad. El resultado se encuentra en el siguiente teorema

**Teorema 2.3** *Una condición necesaria para que una función  $f(X)$  sujeta a las restricciones  $g_j(X) = 0$ ,  $j = 1, 2, \dots, m$  tengan un mínimo relativo en el punto  $X^*$  es que las primeras derivadas parciales de la función de Lagrange definida por  $L = L(x_1, x_2, \dots, x_n, l_1, l_2, \dots, l_m)$  con respecto a cada uno de sus argumentos sea cero.*

Construyendo la función de Lagrange como:

$$L(X, \lambda) = f(X) + \sum_{j=1}^n \lambda_j g_j(X) \quad (2.36)$$

Tomando las derivadas parciales de  $L$ , con respecto a cada uno de sus argumentos, iguales a cero se tienen las condiciones necesarias para el mínimo relativo de  $f(X)$  en  $X^*$ .

### 2.3.4. Optimización con restricciones de desigualdad

Esta sección es concerniente con la solución del siguiente problema:

Minimizar  $f(X)$  sujeto a:

$$g_j(X) \leq 0, \quad j = 1, 2, \dots, m \quad (2.37)$$

Las restricciones de desigualdad se pueden transformar a restricciones de igualdad añadiendo variables lentas no negativas,  $y_j^2$ , como sigue:

$$g_j(X) + y_j^2 = 0, \quad j = 1, 2, \dots, m \quad (2.38)$$

donde los valores de las variables lentas aún no se conocen. El problema ahora está en forma tal que se pueden aplicar los métodos discutidos en la sección anterior:

$$\text{Minimizar } f(X) \text{ sujeto a } G_j(X, Y) = g_j(X) + y_j^2 = 0, \quad j = 1, 2, \dots, m \quad (2.39)$$

donde  $Y = \begin{Bmatrix} y_1 \\ y_2 \\ \vdots \\ y_m \end{Bmatrix}$  es el vector de variables lentas.

Este problema se puede resolver convenientemente por el método de multiplicadores de Lagrange. Para este, se puede construir la función de Lagrange como:

$$L(X, Y, \lambda) = f(X) + \sum_{j=1}^m \lambda_j G_j(X, Y) \quad (2.40)$$

donde  $\lambda = \begin{Bmatrix} \lambda_1 \\ \lambda_2 \\ \vdots \\ \lambda_m \end{Bmatrix}$  es el vector de multiplicadores de Lagrange.

Los puntos estacionarios de la función de Lagrange se pueden encontrar resolviendo las siguientes ecuaciones (condiciones necesarias):

$$\frac{\partial L}{\partial x_i}(X, Y, \lambda) = \frac{\partial f}{\partial x_i}(X) + \sum_{j=1}^m \lambda_j \frac{\partial g_j}{\partial x_i}(X) = 0, \quad i = 1, 2, \dots, n \quad (2.41)$$

$$\frac{\partial L}{\partial \lambda}(X, Y, \lambda) = G_j(X, Y) = g_j(X) + y_j^2 = 0, \quad j = 1, 2, \dots, m \quad (2.42)$$

$$\frac{\partial L}{\partial y_j}(X, Y, \lambda) = 2\lambda_j y_j = 0, \quad j = 1, 2, \dots, m \quad (2.43)$$

La ecuación (2.42) asegura que se satisfacen las restricciones  $g_j(X) \leq 0$ ,  $j = 1, 2, \dots, m$ . Mientras (2.43) implica que cualquiera  $\lambda_j = 0$  o  $y_j = 0$ . Si  $\lambda_j = 0$  esto significa que las restricciones son inactivas y por lo tanto se pueden ignorar estas. Por otro lado, si  $y_j = 0$  esto significa que las restricciones son activas ( $g_j = 0$ ) en el punto óptimo. Considerar la división de las restricciones en dos subespacios  $J_1$  y  $J_2$  donde  $J_1 + J_2$  representan las restricciones del espacio total. Sea  $J_1$  indica los índices para las restricciones las cuales son activas en el punto óptimo y sea  $J_2$  incluye los índices para todas las restricciones inactivas.

Así, para  $j \in J_1$ ,  $y_j = 0$  (restricciones activas), y para  $j \in J_2$ ,  $\lambda_j = 0$  (restricciones inactivas), y la ecuación (2.41) se puede simplificar como:

$$\frac{\partial f}{\partial x_i} + \sum_{j \in J_1} \lambda_j \frac{\partial g_j}{\partial x_i} = 0, \quad i = 1, 2, \dots, n \quad (2.44)$$

Similarmente, las ecuaciones (2.42) se pueden escribir como:

$$g_j(X) = 0, \quad j \in J_1 \quad (2.45)$$

$$g_j(X) + y_j^2 = 0, \quad j \in J_2 \quad (2.46)$$

Las ecuaciones (2.44) a (2.46) representan  $n + p + (m - p) = n + m$  ecuaciones con  $n + m$  incógnitas  $x_i$  ( $i = 1, 2, \dots, n$ ),  $l_j$  ( $j \in J_1$ ) y  $y_j$  ( $j \in J_2$ ) donde  $p$  denota el número de restricciones activas.

Asumiendo que las primeras  $p$  restricciones son activas, la ecuación (2.44) se puede expresar como:

$$-\frac{\partial f}{\partial x_i} = \lambda_1 \frac{\partial g_1}{\partial x_i} + \lambda_2 \frac{\partial g_2}{\partial x_i} + \dots + \lambda_p \frac{\partial g_p}{\partial x_i}, \quad i = 1, 2, \dots, n \quad (2.47)$$

estas ecuaciones se pueden escribir colectivamente como:

$$-\nabla f = \lambda_1 \nabla g_1 + \lambda_2 \nabla g_2 + \dots + \lambda_p \nabla g_p \quad (2.48)$$

donde  $\nabla f$  y  $\nabla g_j$  son el gradiente de la función objetivo y la  $j$ -ésima restricción dados, respectivamente, por:

$$\nabla f = \begin{Bmatrix} \partial f / \partial x_1 \\ \partial f / \partial x_2 \\ \vdots \\ \partial f / \partial x_n \end{Bmatrix} \quad \text{y} \quad \nabla g_j = \begin{Bmatrix} \partial g_j / \partial x_1 \\ \partial g_j / \partial x_2 \\ \vdots \\ \partial g_j / \partial x_n \end{Bmatrix}$$

Así el negativo del gradiente de la función objetivo se puede expresar como una combinación lineal de los gradientes de las restricciones activas en el punto óptimo.

Más allá, se puede considerar el caso de un problema de minimización  $\lambda_j$ 's ( $j \in J_1$ ) tienen que ser positivos.

Suponga que son activas solamente dos restricciones ( $p = 2$ ) en el punto óptimo, entonces la ecuación (2.48) se reduce a:

$$-\nabla f = \lambda_1 \nabla g_1 + \lambda_2 \nabla g_2 \quad (2.49)$$

Sea  $S$  una dirección factible en el punto óptimo, premultiplicando a ambos lados de la ecuación (2.49) por  $S^T$  se tiene:

$$-S^T \nabla f = \lambda_1 S^T \nabla g_1 + \lambda_2 S^T \nabla g_2 \quad (2.50)$$

Desde que  $S$  es la dirección factible, esta debe satisfacer las relaciones:

$$\begin{aligned} S^T \nabla g_1 &< 0 \\ S^T \nabla g_2 &< 0 \end{aligned} \quad (2.51)$$

Así, si  $\lambda_1 > 0$  y  $\lambda_2 > 0$ , la cantidad  $S^T \nabla f$  indica la dirección del gradiente, a lo largo de la cual el valor de la función se incrementa en un valor máximo,  $S^T \nabla f$  representa el componente del incremento de  $f$  a lo largo de la dirección  $S$ . Debido al análisis previo, si  $\lambda_1$  y  $\lambda_2$  son positivos se considera que la ecuación (2.50) proporciona el valor (mínimo) óptimo de la función  $f$ . Por un procedimiento similar, se puede mostrar que los  $\lambda_j$ 's tienen que ser negativos para el problema de maximización.

### Condiciones Kuhn-Tucker

Así, las restricciones que se tienen que satisfacer para un punto mínimo,  $X^*$ , del problema tratado en ecuación (2.37), se pueden expresar como:

$$\frac{\partial f}{\partial x_i} + \sum_{j \in J_1} \lambda_j \frac{\partial g_j}{\partial x_i} = 0, \quad i = 1, 2, \dots, n \quad (2.52)$$

$$\lambda_j > 0, \quad j \in J_1 \quad (2.53)$$

Estas condiciones no son en general suficientes para asegurar un mínimo relativo. De cualquier manera, hay una clase de problemas, para los cuales, son necesarias y suficientes para un mínimo global [20].

Si no se conoce el espacio de las restricciones activas ( $y_j = 0$  y  $\lambda_j = 0$ ), las condiciones Kuhn-Tucker se denotan como sigue:

$$\left\{ \begin{array}{l} \frac{\partial f}{\partial x_i} + \sum_{j \in J_1} \lambda_j \frac{\partial g_j}{\partial x_i}(X) = 0, \quad i = 1, 2, \dots, n \\ \lambda_j g_j = 0, \quad j = 1, 2, \dots, m \\ g_j \leq 0, \quad j = 1, 2, \dots, m \\ \lambda_j \geq 0, \quad j = 1, 2, \dots, m \end{array} \right. \quad (2.54)$$

Note que si el problema es de maximización o si las restricciones son del tipo  $g_j \geq 0$ , entonces  $\lambda_j$  tiene que ser no positivo en la ecuación (2.54).

### Problema de programación convexa

Cualquier problema de optimización descrito por la ecuación (2.37) se le llama el problema de programación convexa donde se proporcionan la función objetivo  $f(X)$  y las funciones de restricción  $g_j(X)$  la cuales son convexas.

Suponga que  $f(X)$  y  $g_j(X)$ ,  $j = 1, 2, \dots, m$ , son funciones convexas, la función de Lagrange de la ecuación (2.40) se puede escribir como

$$L(X, Y, \lambda) = f(X) + \sum_{j=1}^m \lambda_j [g_j(X) + y_j^2] \quad (2.55)$$

Si  $\lambda_j \geq 0$ , entonces  $\lambda_j g_j(X)$  es convexa (dado que  $g_j(X)$  es convexa) y ya que  $\lambda_j y_j = 0$  desde la ecuación (2.43),  $L(X, Y, \lambda)$  también es una función convexa. Se tiene que obtener la derivada de las condiciones necesarias para que  $f(X)$  sea un mínimo relativo en  $X^*$ . De cualquier manera, si  $L(X, Y, \lambda)$  es una función convexa, su derivada es cero solamente en un punto y por tanto este punto debe ser un mínimo absoluto para la función  $f(X)$ . Así, las condiciones de Kuhn-Tucker son necesarias y suficientes para asegurar un mínimo absoluto de  $f(X)$  en  $X^*$ .



# Capítulo 3

## Modelos matemáticos del proceso de mezclado de petróleo crudo

### 3.1. Introducción

En este capítulo se presentan los modelos de las propiedades estáticas del mezclado de petróleo crudo [18]. Mas adelante se presentan dos modelos para el tanque, el modelo dependiente de la altura y el modelo dependiente de la densidad y del volumen en el tanque [1][25]. Posteriormente se analizan los procesos de mezclado que contienen más de un nodo de mezclado, para los cuales se utilizan el modelo distribuido y el modelo agregado [1]. En el modelo distribuido, el modelo de mezclado se representa por puntos de mezclado independientes con un controlador para cada uno, teniendo solo información local del proceso. En el modelo agregado, el proceso de mezclado se representa como un solo punto de mezclado utilizando solo un controlador el cual tiene información global del proceso. Al final se presenta como se modelan cada uno de estos modelos los cuales representan en el proceso de mezclado.

## 3.2. Modelos de las propiedades estáticas de mezclado

La termodinámica de las mezclas o soluciones homogéneas es un tema complejo. Las representaciones elementales de los sujetos complejos frecuentemente restringen el tratamiento a casos especiales que no requieren el desarrollo completo de la teoría general, la cual está acompañada de terminologías y notaciones especiales. Por otro lado, la solución ideal no representa adecuadamente el comportamiento de muchos sistemas de interés práctico para ingeniería química [18].

### 3.2.1. Relación de propiedad para sistemas de composición variable

La propiedad fundamental para los sistemas homogéneos (fase-individual) es:

$$d(nU) = Td(nS) - Pd(nV) + \sum(\mu_i dn_i) \quad (3.1)$$

Como resultado de las definiciones de entalpía, energía libre de Helmholtz, y la energía libre de Gibbs, se pueden escribir:

$$\begin{aligned} dH &= nU + P(nV) \\ nA &= nU - T(nS) \\ dG &= nU + P(nV) - T(nS) \end{aligned}$$

La diferenciación de estas ecuaciones y la substitución por  $d(nU)$  como se da en la ecuación (3.1) da como resultado las siguientes expresiones generales para  $d(nH)$ ,  $d(nA)$  y  $d(nG)$ :

$$d(nH) = Td(nS) - (nV)dP + \sum(\mu_i dn_i) \quad (3.2)$$

$$d(nA) = -(nS)dT - Pd(nV) + \sum(\mu_i dn_i) \quad (3.3)$$

$$d(nG) = -(nS)dT + (nV)dP + \sum(\mu_i dn_i) \quad (3.4)$$

Estas ecuaciones se aplican para los cambios en los estados de equilibrio en cualquier sistema de fluido homogéneo, sin importar si es abierto o cerrado. Cuando las  $n_i$  son todas constantes

( $dn_i = 0$ ), se reducen las ecuaciones a las aplicables a sistemas de masa constante y de composición constante. Además, considerando que las  $n_i$  son todas constantes ( $dn_i = 0$ ), el último término de ecuaciones (3.1) y (3.2) a (3.4) se reescribe de la siguiente manera:

$$\mu_i = \left[ \frac{\partial(nU)}{\partial n_i} \right]_{nS, nV, n_j} = \left[ \frac{\partial(nH)}{\partial n_i} \right]_{nS, P, n_j} = \left[ \frac{\partial(nA)}{\partial n_i} \right]_{nV, T, n_j} = \left[ \frac{\partial(nG)}{\partial n_i} \right]_{T, P, n_j} \quad (3.5)$$

donde el índice  $n_j$  indica la constancia del numero de moles total sin considerar  $n_i$ .

Aplicando el criterio de la exactitud a las expresiones diferenciales al lado derecho de las ecuaciones (3.1) y (3.2) a (3.4) se tienen 16 ecuaciones generales, de las cuales, 4 son ecuaciones de Maxwell. Dos ecuaciones seguidas de la ecuación (3.4) son de considerable utilidad:

$$\left( \frac{\partial \mu_i}{\partial T} \right)_{P, n} = - \left[ \frac{\partial(nS)}{\partial n_i} \right]_{T, P, n_j} \quad (3.6)$$

y

$$\left( \frac{\partial \mu_i}{\partial P} \right)_{T, n} = \left[ \frac{\partial(nV)}{\partial n_i} \right]_{T, P, n_j} \quad (3.7)$$

donde el índice  $n$  indica la constancia del número total de moles.

### 3.2.2. Propiedades molares parciales

Cada una de las derivadas de las ecuaciones de la (3.5) a la (3.7) es una derivada parcial de la forma  $\left[ \frac{\partial(nV)}{\partial n_i} \right]_{T, P, n_j}$ . Tales derivadas tienen un significado especial en las soluciones termodinámicas, y se les llama propiedades derivadas parciales del componente  $i$  en solución, designado por  $\overline{M}_i$ . Así por definición:

$$\overline{M}_i = - \left[ \frac{\partial(nM)}{\partial n_i} \right]_{T, P, n_j} \quad (3.8)$$

donde  $M$  representa cualquier propiedad termodinámica molar de una solución en la cual  $i$  es un componente. La variable con barra  $\overline{M}_i$  se identifica como una propiedad molar parcial. Los componentes de una solución están íntimamente interrelacionados y no pueden tener propiedades de componentes puros. Las propiedades molares parciales se les puede tratar

exactamente como si ellas representaran las propiedades molares de los componentes en solución.

Las propiedades para las cuales se aplica la ecuación (3.8) se conocen desde la experimentación como extensivas; esto es, si una solución se construye de  $n_1$  moles del componente 1,  $n_2$  moles del componente 2,  $n_i$  moles del componente  $i$ , etc, tiene una propiedad inicial total  $nM$ , donde  $n = \sum n_i$ , entonces incrementando cada  $n_i$  por el factor constante  $\alpha$  en T y P produce una solución con una propiedad total de  $\alpha(nM)$ . Matemáticamente, las propiedades  $nM$  se dice que son funciones homogéneas de primer grado en el número de moles de los constituyentes, y para tales propiedades el teorema de Euler proporciona una relación general:

$$nM = \sum (n_i \bar{M}_i) \quad (3.9)$$

Dividiendo por  $n$  produce la expresión alternativa:

$$M = \sum (x_i \bar{M}_i) \quad (3.10)$$

donde  $x_i$  es la fracción mol del componente  $i$  en la solución.

Las ecuaciones (3.9) y (3.10) resultan aplicando balances de materia a la ecuación (3.8), la cual se describe como la propiedad de una solución la cual se distribuye entre sus componentes. Esta expresión trata de ser lo más simple posible para la propiedad de una solución en relación a las propiedades molares parciales, la ecuación (3.10) muestra que la propiedad de la solución es el promedio fracción-peso de moles de las propiedades parciales.

Las propiedades de la solución las cuales se representan por el símbolo  $M$  pueden estar sobre una base en unidades-masa así como sobre una base de moles.

### 3.2.3. Soluciones ideales

Para algunos sistemas, el volumen molar parcial de un componente en una solución es linealmente el mismo que el volumen lineal de un componente puro. Se toman como referencia los sistemas llamados soluciones ideales, para los cuales el término significa que para una solución el volumen molar parcial de cada componente en solución es igual al volumen del componente puro a la misma temperatura y presión [18].

El lado derecho de la ecuación (3.8) representa el cambio en el volumen de  $n$  moles de una mezcla cuando se adicionan  $dn_i$  moles del componente  $i$  a T y P constantes, y  $n_j$ , para una solución ideal, es simplemente el volumen molar puro  $i$ . Por lo tanto, la ecuación (3.8) se convierte en:

$$\bar{V}_i = \left[ \frac{\partial(nV)}{\partial n_i} \right]_{T,P,n_j} = V_i \quad (3.11)$$

desde que el volumen molar parcial es igual al volumen molar del componente puro para una solución ideal, la contraparte de ecuaciones (3.9) y (3.10) son:

$$nV = \sum (n_i V_i) \quad (3.12)$$

y

$$V = \sum (x_i V_i) \quad (3.13)$$

Las ecuaciones (3.11) a la (3.13) se pueden emplear para las propiedades extensivas como la entalpía y la energía interna, pero no para la entropía y la energía de Gibbs. Hay una diferencia entre  $\bar{S}_i$  y  $S_i$  aún para soluciones ideales, la diferencia es debida a la irreversibilidad inherente del proceso de mezclado considerado a temperatura y presión constantes.

### 3.2.4. Cambio de las propiedades de una mezcla (actividad)

La ecuación (3.13) muestra que el volumen molar de una solución ideal es simplemente el promedio de los volúmenes molares de los componentes puros a la misma temperatura y presión, cada peso acorde a su fracción mol. Si esto fuera cierto en general para todas las propiedades extensivas termodinámicas, se podría escribir:

$$M = \sum x_i M_i$$

Desafortunadamente, esta ecuación es cierta solamente para casos especiales. Para que tenga una validez universal se adiciona un término  $\Delta M$ . Así la propiedad de la mezcla general se expresa mediante la siguiente ecuación:

$$M = \sum x_i M_i + \Delta M \quad (3.14)$$

Donde  $\Delta M$  es la diferencia es la diferencia entre la propiedad de la solución  $M$  y la suma de las propiedades de los componentes puros los cuales construyen esta, todos a la misma temperatura y presión como en la solución. Por lo tanto el término  $\Delta M$  es llamado cambio de propiedad o no linealidad de la mezcla.

Una definición más general de esta ecuación está dada por:

$$\Delta M = M - \sum x_i M_i^o \quad (3.15)$$

donde  $M_i^o$  es la propiedad molar de el componente  $i$  en un estado estándar especificado. Si, por ejemplo,  $M$  es el volumen molar de una solución líquida y si todos los componentes existen en un estado líquido a la misma presión y temperatura como en la solución, entonces  $M_i^o$  se convierte en  $V_i^o = V_i$ . en este caso la ecuación (3.15) se reescribe de la siguiente manera:

$$\Delta V = V - \sum x_i V_i$$

donde  $\Delta V$  es el cambio de volumen de la mezcla observado cuando 1 mol de mezcla se forma a P y T constantes desde los componentes líquidos puros. Los cambios de propiedad de la mezcla son funciones de la temperatura, la presión, la composición, la afinidad molecular, así como cualquier otra propiedad termodinámica.

La ecuación (3.10) relaciona las propiedades termodinámicas molares  $M$  de una solución a las propiedades molares parciales de los componentes:

$$M = \sum (x_i \bar{M}_i)$$

La sustitución de esta ecuación para  $M$  dentro de la ecuación (3.15) da como resultado:

$$\Delta M = \sum x_i \bar{M}_i - \sum x_i M_i^o = \sum x_i (\bar{M}_i - M_i^o)$$

Si se define el cambio de la propiedad molar parcial como:

$$\overline{\Delta M}_i = (\bar{M}_i - M_i^o) \quad (3.16)$$

entonces la ecuación (3.15) se puede reescribir como:

$$\Delta M = \sum x_i \overline{\Delta M}_i \quad (3.17)$$

donde la cantidad  $\overline{\Delta M}_i$  representa el cambio de propiedad de  $i$  cuando 1 mol de  $i$  puro en su estado estándar se convierte en un componente de una solución de una composición dada a la misma  $T$  y  $P$ . Esta es también una propiedad molar parcial con respecto a  $\Delta M$ , y está en función de  $T$ ,  $P$  y  $x$ .

La relación general dada por la ecuación (3.17) se puede aplicar a diferentes tipos de cambios de la mezcla. Así, se pueden escribir las siguientes funciones:

$$\begin{aligned}\frac{\Delta G}{RT} &= \frac{1}{RT} \sum [x_i (\overline{G}_i - G_i^o)] \\ \frac{P\Delta V}{RT} &= \frac{P}{RT} \sum [x_i (\overline{V}_i - V_i^o)] \\ \frac{\Delta H}{RT} &= \frac{1}{RT} \sum [x_i (\overline{H}_i - H_i^o)] \\ \frac{\Delta S}{R} &= \frac{1}{R} \sum [x_i (\overline{S}_i - S_i^o)]\end{aligned}$$

y haciendo manipulaciones matemáticas se obtienen la siguiente ecuaciones:

$$\frac{\Delta G}{RT} = \sum [x_i \ln \hat{a}_i] \quad (3.18)$$

$$\frac{P\Delta V}{RT} = \sum \left[ x_i \left( \frac{\partial \ln \hat{a}_i}{\partial \ln P} \right)_{T,x} \right] \quad (3.19)$$

$$\frac{\Delta H}{RT} = - \sum \left[ x_i \left( \frac{\partial \ln \hat{a}_i}{\partial \ln T} \right)_{P,x} \right] \quad (3.20)$$

$$\frac{\Delta S}{R} = - \sum [x_i \ln \hat{a}_i] - \sum \left[ x_i \left( \frac{\partial \ln \hat{a}_i}{\partial \ln T} \right)_{P,x} \right] \quad (3.21)$$

donde la siguiente proporción se le llama la actividad  $\hat{a}_i$  del componente  $i$  en solución:

$$\hat{a}_i = \frac{\hat{f}_i}{f_i^o} \quad (3.22)$$

donde  $f_i$  es la fugacidad del componente puro  $i$ , y  $\hat{f}_i$  es la fugacidad del componente en solución.

Estas ecuaciones muestran la importancia de la proporción  $\hat{a}_i = \frac{\hat{f}_i}{f_i^o}$  que juega en las soluciones termodinámicas. Los valores de los cambios de las propiedades de la mezcla dependen de la elección de los estados estándar.

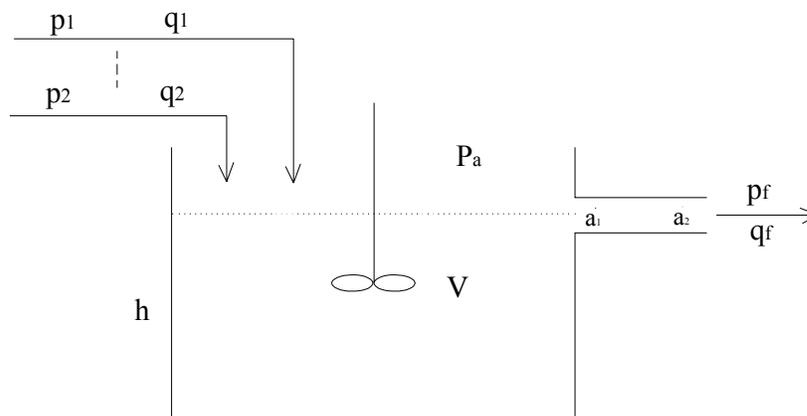


Figura 3.1: Propiedades dinámicas del flujo en el tanque

### 3.3. Modelos de las propiedades dinámicas de mezclado

#### 3.3.1. Modelo 1. En función de la altura en el tanque

Se emplean las siguientes consideraciones para el tanque: 1). mezclado perfecto; 2). operación isotérmica; 3). reacción química que toma lugar en el líquido del tanque [25].

La Figura 3.1 muestra las propiedades dinámicas en el tanque. A partir de un balance de masa de este sistema se tiene:

$$\frac{d}{dt}m = \sum_{i=1}^n q_i - q_f$$

donde  $m$  es la masa total en el tanque,  $m = \rho_f V = \rho_f A h$ ,  $\rho_f$  es la densidad del producto de salida,  $V$ ,  $A$  y  $h$  son volumen, área de la superficie y la altura del líquido del tanque (en el caso de mezcla perfecta se tiene  $\frac{d}{dt}\rho_f = 0$ ), de la ecuación anterior se tiene:

$$\frac{d}{dt}m = \rho_f A \frac{d}{dt}h = \sum_{i=1}^n q_i - q_f \quad (3.23)$$

La velocidad de flujo másico de salida se puede escribir en términos de la velocidad de salida  $v_f$ , por tanto se tiene:

$$q_f = \rho_f A_f v_f \quad (3.24)$$

donde  $A_f$  es el área de cruce-seccional en la salida. Desde que la energía almacenada es una sumatoria de la energía interna, la energía potencial y el trabajo del flujo, un balance de energía en el puerto de salida (puntos  $a_1$  y  $a_2$ ) es:

$$\frac{d}{dt}E = 0 = q_f (u_1 - u_2) + \frac{q_f}{2} (v_1^2 - v_2^2) + q_f g (z_1 - z_2) + \frac{q_f}{\rho_f} (P_1 - P_2)$$

Si se asumen también cambios insignificantes en la energía interna ( $u_1 = u_2$ ) y en la elevación ( $z_1 = z_2$ ), que  $v_1$  es pequeño comparado a  $v_2$ , la conservación de la energía en la ecuación de estados da:

$$v_f = v_2 = \sqrt{\frac{2}{\rho_f} (P_1 - P_2)} = \sqrt{\frac{2}{\rho_f} (P_a + \rho_f g h - P_a)} = \sqrt{2gh} \quad (3.25)$$

donde  $P_a$  es la presión atmosférica. Desde (3.23), (3.24) y (3.25) se tiene:

$$\rho_f A \frac{d}{dt}h = -\rho_f A_f \sqrt{2gh} + \sum_{i=1}^n q_i \quad (3.26)$$

Así, la ecuación para este sistema de primer orden es no lineal. Esta expresión dice que la velocidad del cambio de masa dentro es igual a la velocidad de flujo másico de entrada menos la velocidad de flujo másico en la salida, donde la masa en el tanque y la velocidad de flujo en la salida son ambos escritos en términos de la altura ( $h$ ) del fluido en el tanque.

Si se define:

$$R = \frac{h}{\rho_f A_f \sqrt{2gh}}, \quad C = \frac{\rho_f A h}{h} = \rho_f A$$

La ecuación (3.26) se convierte:

$$\dot{x} = -\frac{1}{R(x)C}x + \frac{1}{C}u \quad (3.27)$$

donde  $u = \sum_{i=1}^n q_i$ ,  $x = h$ .

Las propiedades químicas son exotérmicas (liberan energía) o endotérmicas (requieren la entrada de energía) y por lo tanto requieren ya sea eliminar o adicionar energía al reactor para que se mantenga en este una temperatura constante. Las reacciones exotérmicas son los sistemas más importantes para el caso de estudio debido a los problemas de seguridad (un rápido incremento en la temperatura, algunas veces llamado comportamiento "encendido") y la posibilidad de un comportamiento exótico como un estado múltiple (para el mismo valor en la variable de entrada podría haber varios posibles valores de la variable de salida). Se considera que  $m$  elementos de los  $n$  flujos de entrada tienen reacción química, la reacción es exotérmica irreversible de primer orden,  $A_1 + A_2 \cdots + A_m \rightarrow B$ .

En la Figura 3.1 se observa que se alimentan continuamente  $m$  torrentes de fluido al reactor y un torrente de fluido se elimina en forma continua desde el reactor. Debido a que el reactor mezcla perfectamente, el torrente de salida tiene la misma concentración y temperatura que la del fluido en el reactor. El balance de material total correspondiente al volumen es:

$$\frac{d}{dt}(\rho_f V) = -\rho_f V_f + \sum_{i=1}^n (\rho_i V_i) \quad (3.28)$$

donde  $V$  es el volumen del tanque,  $V_i$  es el volumen de entrada y  $V_f$  es el volumen de salida. El balance sobre el componente  $A_i$  ( $i = 1 \cdots m$ ) es:

$$\frac{d}{dt}(c_{ri} V_i) = -c_{ri} V_i + c_i V_i - r_i V \quad (3.29)$$

donde  $c_{ri}$  es la concentración en el reactor correspondiente al componente  $A_i$ ,  $c_i$  es la concentración en el flujo de alimentación,  $r_i$  es la velocidad de reacción en  $A_i$ . El balance sobre el componente  $B$  es:

$$\frac{d}{dt}(c_f V_f) = -c_{rf} V_f + c_f V_f + r_f V$$

donde  $c_{rf}$  es la concentración en el reactor del componente  $B$ ,  $c_f$  es la concentración a la salida del tanque,  $r_f$  es la velocidad de reacción en  $B$ .

Las dos propiedades para el mezclado del tanque discutidas en este documento son: el nivel del tanque  $h$  el cual está descrito en la ecuación (3.27); la concentración de cada elemento está descrita en la ecuación (3.29).

Ejemplo:

El modelo del tanque en función de la altura se describe mediante la ecuación (3.27) la cual se enuncia de nuevo a continuación:

$$\dot{x} = -\frac{1}{R(x)C}x + \frac{1}{C}u$$

$$R = \frac{h}{\rho_f A_f \sqrt{2gh}}$$

$$C = \rho_f A$$

$$u = \sum_{i=1}^n q_i$$

$$x = h$$

donde:

$A$  área de la superficie.

$A_f$  área de cruce seccional en la salida.

$\rho_f$  densidad a la salida del tanque equivalente a la densidad en el tanque.

$g$  aceleración de la gravedad.

$q_i$  flujo en la  $i$ -ésima entrada del tanque.

Realizando la sustitución de los elementos en la anterior ecuación se tiene:

$$\begin{aligned} \dot{x} &= -\frac{1}{R(x)C}x + \frac{1}{C}u \\ \dot{h} &= -\frac{\rho_f A_f \sqrt{2gh}}{h \rho_f A} h + \frac{1}{\rho_f A} \sum_{i=1}^n q_i \end{aligned}$$

$$\dot{h} = -\frac{A_f}{A} \sqrt{2gh} + \frac{1}{\rho_f A} \sum_{i=1}^n q_i \quad (3.30)$$

La ecuación (3.30) describe la propiedad dinámica de la altura en el tanque. Como un

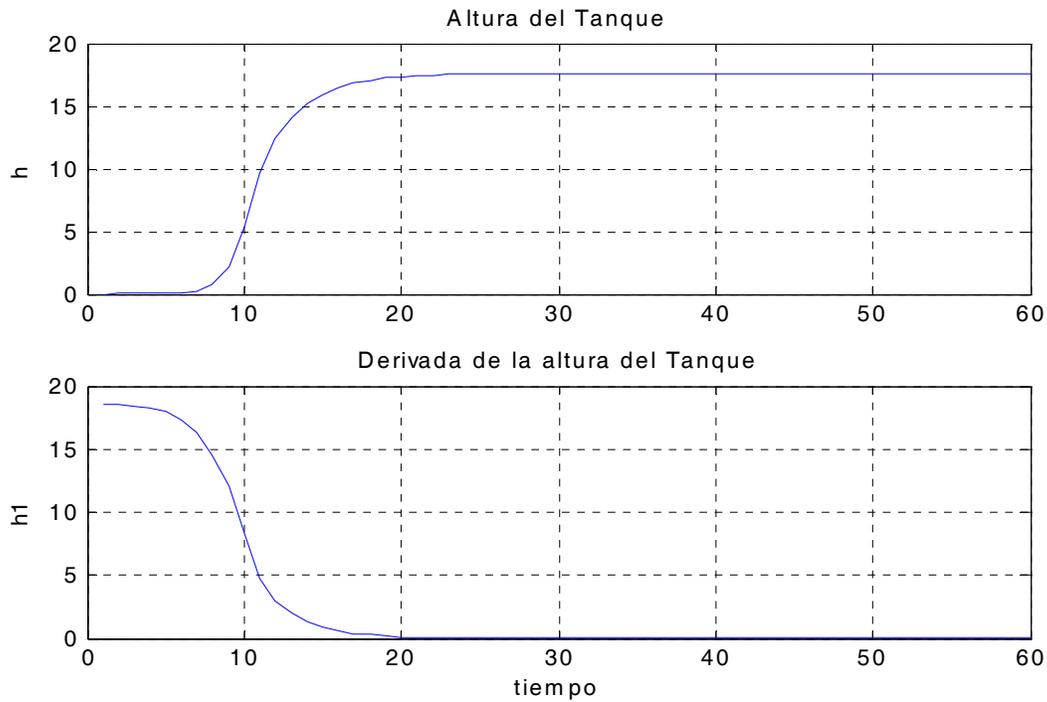


Figura 3.2: Altura y su derivada en el tanque

ejemplo, para los siguientes datos:

$$\begin{aligned}
 A &= 100 \text{ m}^2 \\
 A_f &= 100 \text{ m}^2 \\
 \rho_f &= 32,23 \text{ }^\circ\text{API} \\
 g &= 9,81 \text{ Kg/m}^2 \\
 \sum_{i=1}^2 q_i &= 59903 \text{ m}^3/\text{h}
 \end{aligned}$$

El resultado esta expresado en la gráfica de la Figura 3.2.

### 3.3.2. Modelo 2. En función de la densidad y del volumen en el tanque

Para simular el comportamiento dinámico de la densidad y el volumen de la sección de almacenamiento y carga a buque tanques se utiliza la siguiente ecuación:

$$\frac{dm}{dt} = f_{in} - f_o \quad (3.31)$$

donde  $m$  representa la masa de petróleo crudo contenido en el tanque.  $f_{in}$  y  $f_o$  son los flujos de entrada y de salida respectivamente en el tanque, con la diferencia de que en el buque tanque  $f_o = 0$  [1]. Las variables de interés son la densidad y el volumen en el tanque y el buque tanque, desarrollando (3.31) se tiene:

$$\begin{aligned} \frac{d(\rho V)}{dt} &= f_{in} - f_o \\ \rho \frac{dV}{dt} + V \frac{d\rho}{dt} &= f_{in} - f_o \\ \frac{dV}{dt} &= \left( f_{in} - f_o - V \frac{d\rho}{dt} \right) \frac{1}{\rho} \end{aligned} \quad (3.32)$$

La ecuación (3.32) describiría el volumen, pero el problema es que contiene dos variables y solo una ecuación lo cual hace imposible encontrar una solución. Para solucionar este problema, en la práctica se observa que el comportamiento dinámico de la densidad en un tanque puede ser descrita por:

$$\frac{d\rho}{dt} = \frac{(\rho_{in} - k_2\rho)}{k_1} \quad (3.33)$$

donde  $k_{1,2}$  son los parámetros de ajuste y  $\rho_{in}$  es la densidad de la corriente que ingresa al tanque o al buque tanque.

Sustituyendo la ecuación (3.33) en la ecuación (3.32) se tiene:

$$\begin{aligned} \frac{dV}{dt} &= \left( f_{in} - f_o - V \frac{d\rho}{dt} \right) \frac{1}{\rho} \\ \frac{dV}{dt} &= \left[ f_{in} - f_o - V \left( \frac{\rho_{in} - k_2\rho}{k_1} \right) \right] \frac{1}{\rho} \\ \frac{dV}{dt} &= \left[ f_{in} - f_o - V \left( \frac{\rho_{in} - k_2\rho}{k_1} \right) \right] \frac{1}{\rho} \end{aligned} \quad (3.34)$$

La ecuación (3.33) describe el comportamiento dinámico de la densidad y la ecuación (3.34) describe el comportamiento dinámico del volumen en la sección de almacenamiento, por tanto, las ecuaciones (3.33) y (3.34) describen el comportamiento dinámico en el tanque y en el buque tanque.

Ejemplo:

El modelo del tanque en función de la densidad y del volumen se describe mediante las ecuaciones (3.33) y (3.34) respectivamente, las cuales se enuncian de nuevo a continuación:

$$\frac{d\rho}{dt} = \frac{(\rho_{in} - k_2\rho)}{k_1}$$

$$\frac{dV}{dt} = \left[ f_{in} - f_o - V \left( \frac{\rho_{in} - k_2\rho}{k_1} \right) \right] \frac{1}{\rho}$$

donde:

$f_{in}$  flujo de entrada al tanque o al buque tanque.

$f_o$  flujo de salida al tanque o al buque tanque. Para el caso del buque tanque se tiene  $f_o = 0$ .

$\rho_{in}$  densidad de la corriente que ingresa al tanque o al buque tanque.

$V$  volumen del tanque o del buque tanque.

$\rho$  densidad del tanque o del buque tanque.

$k_{1,2}$  parámetros de ajuste.

Para el caso del buque tanque donde se tienen los siguientes datos:

$$\rho_{in} = 32,23 \text{ } ^\circ API$$

$$f_{in} = 176979 \text{ m}^3/\text{h}$$

$$f_o = 0$$

El cual se ajusta para los siguientes parámetros de ajuste:

$$k_1 = 6$$

$$k_2 = 1$$

El resultado de la densidad y el volumen en el buque tanque están expresados en la gráfica de la Figura 3.3.

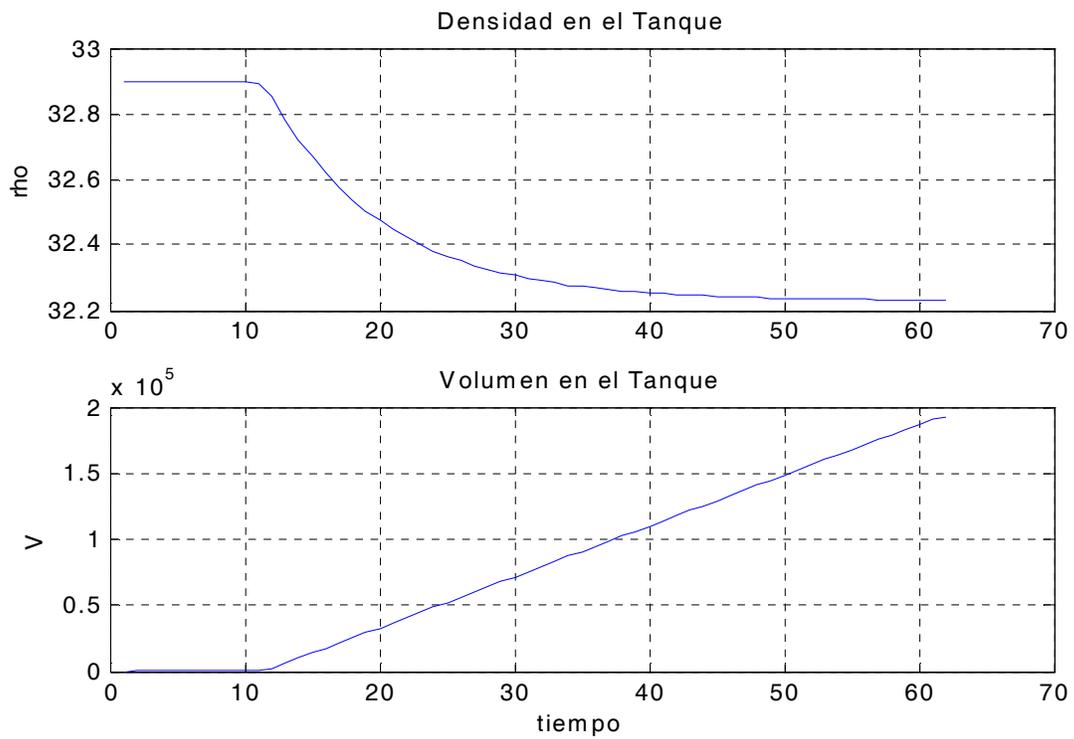


Figura 3.3: Densidad y Volumen en el buque tanque

### 3.4. Modelos del mezclado de petróleo crudo

En la Figura (3.5) se muestra el arreglo físico de un proceso de mezclado de petróleo crudo, considerado representativo de los que se utilizan actualmente en la industria petrolera nacional. El proceso consiste de dos etapas: una etapa de mezclado y una etapa de almacenamiento y carga a buque tanques [1].

La etapa de mezclado está compuesta por tres tipos de nodos: nodos de mezclado (Mezcladores), un nodo de deshidratación (Deshidratador) y un nodo divisor de corriente (Círculo). Los mezcladores tienen dos corrientes de entrada y una de salida cada uno, el nodo divisor de corriente es utilizado para el envío de una mezcla considerada para consumo nacional y el nodo deshidratador se utiliza para la separación del agua contenida en el petróleo crudo.

Se utilizará un modelo agregado y uno distribuido para representar el proceso de mezclado. En el modelo distribuido, el proceso se representa por nodos de mezcla individuales con un controlador para cada uno, donde solo se dispone de información local del proceso. En el modelo agregado, el proceso se representa como un solo nodo de mezclado utilizando solo un controlador, el cual tiene información global del proceso.

La etapa de almacenamiento consta de tanques cerrados para almacenar el petróleo crudo que se produce en la etapa de mezclado y cargar los buque tanques.

Actualmente, las operaciones de mezclado en la industria petrolera se realizan de manera tal, que las propiedades de la mezcla que entregan están por encima de lo requerido ocasionando una fuga económica por producir una mezcla fuera de especificación. Esta situación crea una oportunidad para la aplicación de técnicas de control avanzadas en las operaciones de mezclado para entregar una mezcla que cumpla con las propiedades requeridas y al mismo tiempo obtener un beneficio económico extra.

#### 3.4.1. Regla de mezclado para petróleo crudo

Considere el mezclador de la Figura (3.4) con una corriente de salida  $S_M$  como la mezcla homogénea de las  $m$  corrientes de entrada  $S$  [1]. Cada corriente de entrada es representada

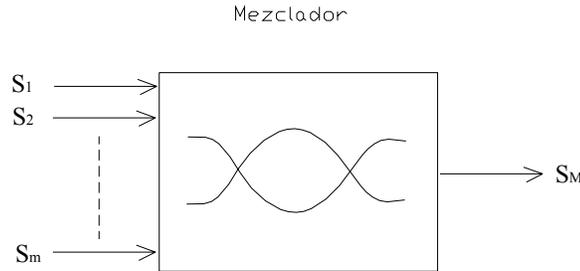


Figura 3.4: Mezcla homogénea de m componentes

por un vector de propiedades como

$$S_i = \{f_{i \text{ mín}}, f_{i \text{ máx}}, \rho_i, c_i\}, \quad i = 1, 2, \dots, m$$

donde:

- $f_{i(\text{mín},\text{máx})}$  es el flujo mínimo y máximo disponible de la i-ésima corriente de entrada, (kg/h).

$\rho_i$  es la densidad de la corriente de salida, ( $^{\circ}API$ , donde  $^{\circ}API$  es una medida indirecta de la densidad empleada en el Instituto Mexicano del Petróleo).

$c_i$  es un costo asociado a la corriente por su utilización en la mezcla, (USD/kg).

Mientras que la corriente de salida se define por el vector:

$$S_M = \{f_{M \text{ mín}}, f_{M \text{ máx}}, \rho_{M \text{ mín}}, \rho_{M \text{ máx}}, c_M\}$$

donde:

$f_{M(\text{mín},\text{máx})}$  es el flujo mínimo y máximo requerido, (kg/h).

$\rho_{M(\text{mín,máx})}$  es la densidad mínima y máxima permitida en la mezcla , ( $^{\circ}API$ ).

$c_M$  es el costo de producción de la mezcla, (USD/kg).

Si la mezcla fuera ideal su densidad, para todo  $t > 0$  (con  $t$  denotando la variable tiempo), se puede obtener como:

$$\rho_M(t) = \sum_{i=1}^m \frac{f_i(t)\rho_i}{f_M} \quad (3.35)$$

pero en general la ecuación (3.35) no es representativa del proceso, ya que la regla de mezclado está dada por una función que depende de las densidades y una no linealidad en los flujos de las corrientes de entrada,  $\rho_M(t) = g(f(t), \rho)$ . Una alternativa es agregar un término de corrección  $\Delta\rho_M(t)$  a (3.35), llamado "propiedad en exceso." "desviación de lo ideal". Siguiendo la idea de las ecuaciones de las propiedades en exceso en soluciones no ideales, el término  $\Delta\rho_M(t)$  se propone como:

$$\Delta\rho_M(t) = \sum_{j=1}^{m-1} \prod_{i=j+1}^m \frac{\pi_{j,i} f_j(t) f_i(t)}{f_M^2} \quad (3.36)$$

donde  $\pi_{j,i}$  es el coeficiente de iteración entre los componentes  $j$  e  $i$  que se mezclarán y su valor es obtenido empíricamente como:

$$\pi_{j,i} = -\delta (\rho_j + \rho_i)$$

donde  $\delta$  es un parámetro de ajuste. Se sabe que para el petróleo crudo se cumple la siguiente relación:

$$\Delta\rho_M(t) \leq 0,1 \left[ \sum_{i=1}^m \frac{f_i(t)\rho_i}{f_M} \right] \quad (3.37)$$

Entonces la densidad de la mezcla se obtiene de una regla de mezclado definida como la suma de una función lineal y un término de corrección que es una función no lineal:

$$\begin{aligned} \rho_M(t) &= \rho_M(t)_{Lineal} + \rho_M(t)_{NoLineal} \\ \rho_M(t) &= \sum_{i=1}^m \frac{f_i(t)\rho_i}{f_M} + \sum_{j=1}^{m-1} \prod_{i=j+1}^m \frac{\pi_{j,i} f_j(t) f_i(t)}{f_M^2} \end{aligned} \quad (3.38)$$

la ecuación (3.38) es la que describe el proceso de mezclado general.

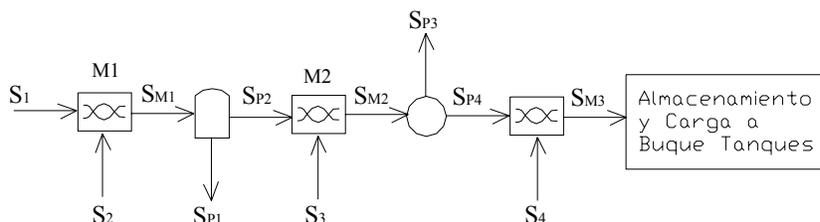


Figura 3.5: Etapa de mezclado de la industria petrolera nacional

### 3.4.2. Proceso de mezclado que incluye diferentes tipos de nodos

Para mostrar como se incluyen los modelos de los distintos tipos de nodos en el modelo de un proceso de mezclado, considere el mezclado de la Figura 3.5, el cual tiene un nodo deshidratador y un nodo divisor de corriente [1].

Para obtener el modelo del deshidratador se sabe que las corrientes  $S_{M1}$  y  $S_{P2}$  están compuestas por una cantidad de petróleo crudo  $O$  y una cantidad de agua  $W$ .

La corriente  $S_{P1}$  está compuesta únicamente por la diferencia de agua entre  $S_{M1}$  y  $S_{P2}$ .

$$f_{M1} = O_{f_{M1}} + W_{f_{M1}} = af_{M1} + bf_{M1} \quad (3.39)$$

$$f_{P1} = W_{f_{M1}} - W_{f_{P2}} \quad (3.40)$$

$$f_{P2} = O_{f_{P2}} + W_{f_{P2}} = cf_{P2} + df_{P2} \quad (3.41)$$

donde  $a$ ,  $b$ ,  $c$  y  $d$  representan las fracciones másicas de los componentes. Asumiendo que la masa total de petróleo crudo que ingresa al deshidratador se conserva a la salida (sólo se separa el agua en el proceso), entonces se tiene:

$$O_{f_{P2}} = O_{f_{M1}}$$

$$\begin{aligned}
 cf_{P2} &= af_{M1} \\
 f_{P2} &= \beta f_{M1}
 \end{aligned}
 \tag{3.42}$$

donde  $\beta = (a/b)$  y la cantidad de agua en  $S_{P2}$  se obtiene como:

$$W_{f_{P2}} = df_{P2} = \frac{ad}{c} f_{M1}$$

entonces (3.40) queda como:

$$\begin{aligned}
 f_{P1} &= bf_{M1} - \frac{ad}{c} f_{M1} \\
 f_{P1} &= \alpha f_{M1}
 \end{aligned}
 \tag{3.43}$$

donde  $\alpha = (bc - ad)/c$ . La nueva densidad de la corriente que sale del deshidratador  $\rho_{P2}$  se obtiene como:

$$\rho_{P2} = \frac{f_{M1}\rho_{P1} - f_{P1}\rho_{P1}}{f_{P2}}$$

utilizando (3.42) y (3.43) se tiene lo siguiente:

$$\rho_{P2} = \frac{\rho_{M1} - \alpha\rho_{P1}}{\beta} \tag{3.44}$$

donde  $\alpha, \beta$  son las constantes de deshidratación y  $\rho_{P1}$  es la densidad del agua dada en °API, la ecuación (3.44) es la que describe la relación que existe entre las densidades de los crudos a la entrada y a la salida del deshidratador.

Ahora, realizando una resta de la ecuaciones (3.39) menos la ecuación (3.41) se tiene lo siguiente:

$$f_{M1} - f_{P2} = W_{f_{M1}} - W_{f_{P2}} + O_{f_{M1}} - O_{f_{P2}} = (bf_{M1} - df_{P2}) + (af_{M1} - cf_{P2}) \tag{3.45}$$

pero como se asumió que la masa de petróleo crudo que ingresa al deshidratador se conserva a la salida descrito por la siguiente ecuación:

$$\begin{aligned}
 O_{f_{P2}} &= O_{f_{M1}} \\
 cf_{P2} &= af_{M1}
 \end{aligned}$$

entonces sustituyendo este resultado en la ecuación (3.45) se tiene el siguiente resultado:

$$f_{M1} - f_{P2} = bf_{M1} - df_{P2}$$

$$f_{M1} = \frac{1-d}{1-b} f_{P2} \quad (3.46)$$

donde  $b$  y  $d$  son las fracciones másicas de agua a la entrada y a la salida del deshidratador, respectivamente, con la ecuación (3.46) se obtiene el flujo que se tiene que introducir a la entrada del deshidratador debido a la restricción impuesta por el flujo que se requiere a la salida del deshidratador.

Para el modelo de un divisor de corriente se considera que la densidad de la corriente de entrada es la misma en las corrientes de salida, esto es  $\rho_{M2} = \rho_{P3} = \rho_{P4}$ , el balance de materia es:

$$f_{M2} = f_{P3} + f_{P4} \quad (3.47)$$

### 3.4.3. Modelo distribuido

Desde este enfoque, cada punto de mezclado es visto de una forma independiente, como se muestra en la Figura 3.6 [1].

Cada nodo mezclador solo tiene información de las propiedades de las corrientes que ingresan en él y las características requeridas en las corrientes de salida o la mezcla deseada. Esto hace que la información que obtiene el controlador asignado a este punto de mezcla solo conozca información local del proceso y por lo tanto desconoce las capacidades físicas de los demás puntos de mezclado.

De la Figura 4.2, el flujo  $f_{Mi}$  y la densidad  $\rho_{Mi}$  de la corriente  $S_{Mi}$  se obtienen como:

$$f_{Mi} = f_{(Mi-1)} + f_i$$

$$\rho_{Mi} = \left( f_{(Mi-1)} \bar{\rho}_{(Mi-1)} + f_j \bar{\rho}_j + \frac{\pi_{(Mi-1,j)} f_{(Mi-1)} f_j}{f_{Mi}} \right) \frac{1}{f_{Mi}}$$

$$i = 1, 2, \dots, n, \quad j = i + 1$$

La representación distribuida de la etapa de mezclado de la Figura 3.5 solo considera los nodos de mezclado en el control, la información de los otros nodos como el deshidratador

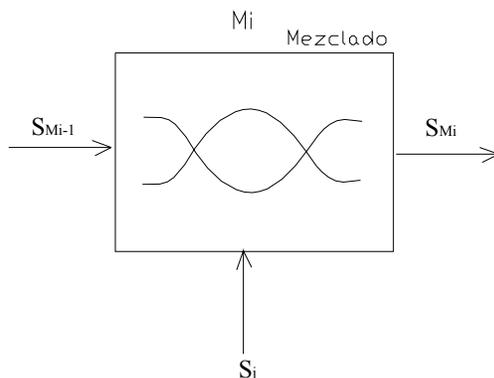


Figura 3.6: Representación distribuida del proceso de mezclado

y el divisor de corriente se incluyen de manera indirecta y no forman parte del modelo de control, por tanto se utiliza la misma regla de mezclado.

#### 3.4.4. Modelo agregado

El modelo consiste en representar el modelo en línea de petróleo crudo de manera agregada, como se muestra en la Figura 3.7 [1].

Bajo esta representación la relación existente entre las corrientes de entrada  $S$  y la de salida  $S_{Mn}$  es obtenida a partir de las ecuaciones de balance de materia y regla de mezclado existentes en cada mezclador.

De la Figura 3.7, el flujo  $f_{M1}$  y la densidad  $\rho_{M1}$  en la corriente  $S_{M1}$  están dados por:

$$f_{M1} = f_1 + f_2 \quad (3.48)$$

$$\rho_{M1} = \left( \bar{\rho}_1 f_1 + \bar{\rho}_2 f_2 + \frac{\pi_1 f_1 f_2}{f_{M1}} \right) \frac{1}{f_{M1}} \quad (3.49)$$

y para el flujo  $f_{M2}$  y la densidad  $\rho_{M2}$  en la corriente  $S_{M2}$  se tiene:

$$f_{M2} = f_{M1} + f_2$$

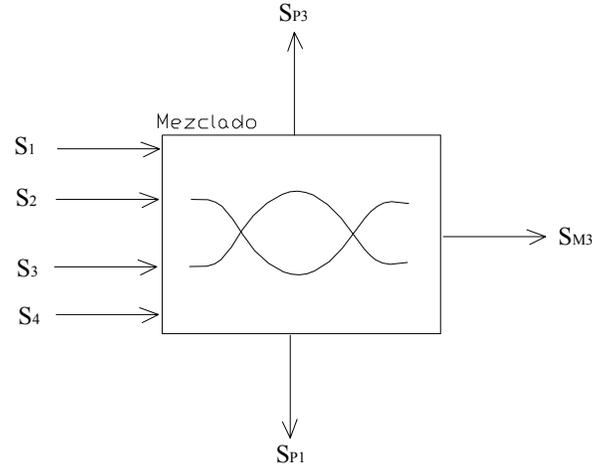


Figura 3.7: Representación agregada de la etapa de mezclado

$$\rho_{M2} = \left( \bar{\rho}_{M1} f_{M1} + \bar{\rho}_3 f_3 + \frac{\pi_2 f_{M1} f_3}{f_{M2}} \right) \frac{1}{f_{M2}}$$

utilizando (3.48) y (3.49) se tiene lo siguiente:

$$f_{M2} = f_1 + f_2 + f_3 \quad (3.50)$$

$$\rho_{M2} = \left[ \left( \bar{\rho}_1 f_1 + \bar{\rho}_2 f_2 + \frac{\pi_1 f_1 f_2}{f_{M1}} \right) \left( \frac{f_1 + f_2}{f_1 + f_2} \right) + \bar{\rho}_3 f_3 + \frac{\pi_2 (f_1 + f_2) f_3}{f_{M2}} \right] \frac{1}{f_{M2}} \quad (3.51)$$

finalmente, separando (3.51) en parte lineal y no lineal se tiene:

$$\rho_{M2} = \left[ \bar{\rho}_1 f_1 + \bar{\rho}_2 f_2 + \bar{\rho}_3 f_3 + \frac{\pi_1 f_1 f_2}{f_1 + f_2} + \frac{\pi_2 (f_1 + f_2) f_3}{f_1 + f_2 + f_3} \right] \frac{1}{f_1 + f_2 + f_3} \quad (3.52)$$

El modelo agregado de la etapa de mezclado de la Figura 3.5, se muestra en la Figura 3.7. Bajo esta representación, la relación entre las corrientes de entrada y la de salida se obtienen de las ecuaciones de balance de materia y de la regla de mezclado existentes entre cada mezclador.

Siguiendo el mismo procedimiento donde se obtuvieron las ecuaciones (3.50) y (3.52) para calcular el flujo y la densidad final en una representación agregada, se obtienen las siguientes

relaciones:

$$f_{M3} = \beta (f_1 + f_2) + f_3 + f_4 - f_{P3}$$

$$\rho_{M3} = \frac{(\bar{\rho}_1 - \alpha\rho_{P1}) f_1 + (\bar{\rho}_2 - \alpha\rho_{P1}) f_2 + \bar{\rho}_3 f_3 + \bar{\rho}_4 f_4 + \Theta}{f_{M3}}$$

donde:

$$\Theta = \left( \frac{\pi_1 f_1 f_2}{f_{M1}} + \frac{\pi_2 f_{P2} f_3}{f_{M2}} \right) \left( 1 - \frac{f_{P3}}{f_{M2}} \right) - ((\bar{\rho}_1 - \alpha\rho_{P1}) f_1 + (\bar{\rho}_2 - \alpha\rho_{P1}) f_2 + \bar{\rho}_3 f_3) \frac{f_{P3}}{f_{M2}} + \left( \frac{\pi_3 f_{P4} f_4}{f_{M3}} \right)$$

$\Theta$  representa la parte no lineal de la mezcla. Aquí la no linealidad de la mezcla es distinta que la del modelo distribuido.

# Capítulo 4

## Modelación del mezclado de petróleo crudo vía redes neuronales y datos reales

### 4.1. Introducción

En este capítulo se presenta como se obtienen los datos reales que se tienen una base de datos reales en un programa de Excel, se establece la comunicación para extraer los datos de Excel con una sentencia de Matlab [22], posteriormente estos datos se emplean para desarrollar la estimación del comportamiento del proceso de mezclado para el modelo agregado y para el modelo distribuido, vía Least Square, vía Redes neuronales y se presentan los resultados para cada uno de los casos, por último se presenta una tabla comparativa que muestra los resultados para cada caso.

Para la modelación del proceso de mezclado de petróleo crudo se consideran 3 tipos de modelos de identificación del mezclado de petróleo crudo: la identificación de parámetros vía least square descrita mediante el inciso (a) de la Figura 4.1, la identificación de parámetros vía redes neuronales para el modelo completo descrita mediante el inciso (b) de la Figura 4.1 y la identificación de parámetros vía redes neuronales para la no linealidad descrita mediante

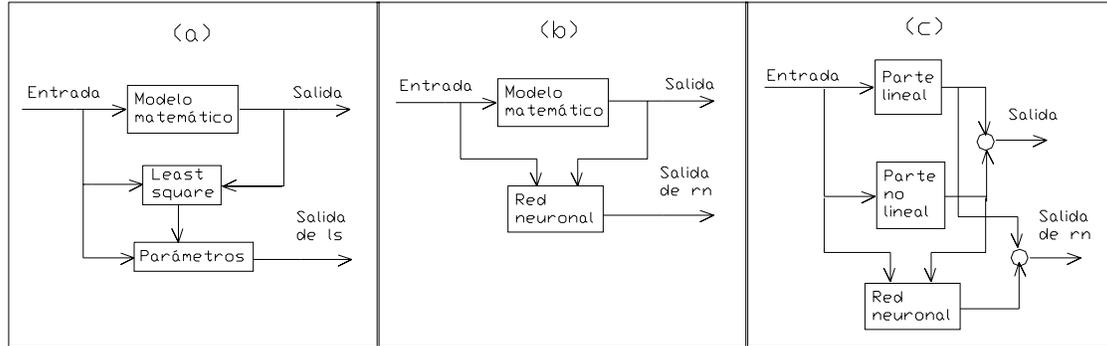


Figura 4.1: Modelos de identificación del mezclado de petróleo crudo

el inciso (c) de la Figura 4.1.

## 4.2. Least square (LS) para identificar los parámetros del mezclado de petróleo crudo

La propiedad estática del mezclado de petróleo crudo se puede expresar mediante la siguiente relación:

$$y = \theta_1 x_1 + \theta_2 x_2 + \dots + \theta_n x_n \quad (4.1)$$

donde  $(x_1, x_2, \dots, x_n, y)$  son datos conocidos y  $(\theta_1, \theta_2, \dots, \theta_n)$  son los parámetros [24][10].

Experimentalmente estos datos no se pueden considerar como una relación exacta, por

tanto se expresan como sigue:

$$\begin{aligned} y_1 &= \theta_1 x_{11} + \theta_2 x_{12} + \dots + \theta_n x_{1n} + e_1 \\ y_2 &= \theta_1 x_{21} + \theta_2 x_{22} + \dots + \theta_n x_{2n} + e_2 \\ &\vdots \\ y_m &= \theta_1 x_{m1} + \theta_2 x_{m2} + \dots + \theta_n x_{mn} + e_m \end{aligned}$$

En forma matricial:

$$Y = X\Theta + E \quad (4.2)$$

donde:

$$\begin{aligned} Y &= [y_1, \dots, y_m]^T \\ X &= \begin{bmatrix} x_{11} & \dots & x_{1n} \\ \vdots & \ddots & \vdots \\ x_{m1} & \dots & x_{mn} \end{bmatrix} \\ \Theta &= [\theta_1, \dots, \theta_n]^T \\ E &= [e_1, \dots, e_m]^T \end{aligned}$$

Aplicando mínimos cuadrados al modelo de la ecuación (4.2) se tiene:

$$J = E^T E = \sum_{i=1}^m e_i^2, \quad \min_{\theta} J \rightarrow \text{least square} \left( \frac{\partial J}{\partial \Theta} = 0 \right)$$

donde  $J$  es la función de costo. Se aplica la derivada a la función de costo con respecto al vector de los parámetros a identificar para obtener el mínimo de la función de costo como sigue:

$$\begin{aligned} \frac{\partial J}{\partial \Theta} &= \frac{\partial J}{\partial E} \frac{\partial E}{\partial \Theta} = 2E \frac{\partial}{\partial \Theta} (Y - X\Theta) \\ &= 2(Y - X\Theta)^T (-X) = \\ &= -2Y^T X + 2\Theta^T X^T X = 0 \end{aligned}$$

Así la estimación de los parámetros se obtiene mediante:

$$\Theta = (X^T X)^{-1} Y^T X \quad (4.3)$$

La no singularidad de  $X^T X$  depende de los datos de entrada  $(x_1, \dots, x_n)$ .

Para el modelo ARX (Auto-Regresivo  $Ay$  con entrada eXtra  $Bu$ ) se tiene

$$Ay(k) = Bu(k) \quad (4.4)$$

donde:

$$\begin{aligned} A &= 1 + a_1q^{-1} + \dots + a_nq^{-n} \\ B &= b_1q^{-1} + \dots + b_mq^{-m} \\ q^{-1} &= \text{unidad de retraso} \end{aligned}$$

Este se puede expresar en forma regresiva lineal de la siguiente manera:

$$y(k) = -a_1y(k-1) - \dots - a_ny(k-n) + b_1u(k-1) + \dots + b_mu(k-m)$$

Cuando se da una entrada  $u(k)$  a la planta, hay una correspondiente salida  $y(k)$ . El vector de parámetros es:

$$\theta = [a_1 \dots a_n, b_1 \dots b_m]^T$$

El vector de datos es:

$$\varphi(k) = [-y(k-1) \dots -y(k-n), u(k-1) \dots, u(k-m)]^T$$

Desde que estos datos no pueden describir exactamente la planta, ésta se puede escribir de la siguiente manera:

$$y(k) = \varphi^T(k)\theta + e(k) \quad (4.5)$$

donde:

$$\begin{aligned} \varphi(k) &= [-y(k-1) \dots -y(k-n), u(k-1) \dots, u(k-m)]^T \\ \theta &= [a_1 \dots a_n, b_1 \dots b_m]^T \end{aligned}$$

$e(k)$  = vector que contiene los errores de modelado, el cual es independiente de la entrada  $u(k)$  y de la salida  $y(k)$

Aplicando mínimos cuadrados al modelo descrito por la ecuación (4.5) se tiene:

$$J = \sum_{i=1}^m e_i^2(k), \quad \underset{\theta}{\text{mín}} J \rightarrow \text{least square} \left( \frac{\partial J}{\partial \theta} = 0 \right)$$

donde  $l > \max(m, n)$ . Se aplica la derivada a la función de costo con respecto al vector de los parámetros a identificar para obtener el mínimo de la función de costo, aplicando la regla de la cadena y sustituyendo el error de modelado de la ecuación (4.5) se tiene lo siguiente:

$$\begin{aligned} \frac{\partial J}{\partial \theta} &= \sum_{k=1}^l \frac{\partial J}{\partial e(k)} \frac{\partial e(k)}{\partial \theta} = 2 \sum_{k=1}^l e(k) \frac{\partial}{\partial \theta} [y(k) - \varphi^T(k)\theta] \\ &= 2 \sum_{k=1}^l [y(k) - \varphi^T(k)\theta] [-\varphi^T(k)] = \\ &= 2 \sum_{k=1}^l y(k)\varphi^T(k) - 2 \left[ \sum_{k=1}^l \varphi^T(k)\varphi(k) \right] \theta = 0 \end{aligned}$$

Así la estimación de los parámetros se obtiene mediante:

$$\theta = \left[ \sum_{k=1}^l \varphi^T(k)\varphi(k) \right]^{-1} \sum_{k=1}^l y(k)\varphi^T(k) \quad (4.6)$$

donde la no-singularidad de  $\sum_{k=1}^l \varphi^T(k)\varphi(k)$  depende de los datos de entrada  $[u(k-1) \dots, u(k-m)]$ .

### 4.3. Redes neuronales para identificar los parámetros del mezclado de petróleo crudo

Las propiedad a la salida del modelo de mezclado de petróleo crudo se puede escribir en la siguiente forma general:

$$y(k) = \Phi [u_1, u_2, u_3, \dots] = \Phi [X(k)] \quad (4.7)$$

donde:

$$X(k) = [u_1, u_2, u_3, \dots]^T \quad (4.8)$$

$\Phi(\cdot)$  es una función lineal conocida que representa la operación de mezclado,  $u_i$  es una entrada escalar medible, esta puede ser la fracción volumen del componente,  $y(k)$  es el valor de la propiedad a la salida del modelo de mezclado al tiempo  $k$ . Este es el modelo NARMA [24].

Se considera una red neuronal multicapa (o perceptrón multicapa, MLP) al modelo estático del proceso de mezclado de la ecuación (4.7), la ecuación que describe la red neuronal multicapa es la siguiente:

$$\hat{y}(k) = V_k \phi [W_k X(k)] \quad (4.9)$$

donde la salida escalar es  $\hat{y}(k) \in R^{n \times 1}$ , el vector de entrada es  $X(k) \in R^{n \times 1}$  definido en ecuación (4.8), los pesos sinópticos en la capa de salida son  $V_k \in R^{1 \times n}$ , los pesos en la capa oculta son  $W_k \in R^{m \times n}$ ,  $\phi$  es una función vectorial m-dimensional, la representación típica de la función  $\phi_i(\cdot)$  es una función sigmoideal debido a que se necesita una función continua para poder obtener su derivada ya que la derivada también se emplea en el algoritmo [25][26]. El error de identificación en la capa de salida se define mediante:

$$e(k) = \hat{y}(k) - y(k)$$

El sistema de mezclado de la ecuación (4.7) se puede reescribir como:

$$y(k) = V^* \phi [W^* X(k)] - \mu(k)$$

donde  $V^*$  y  $W^*$  son los pesos sinópticos desconocidos de la capa de salida y de la capa oculta respectivamente, los cuales pueden minimizar el error de modelado  $\mu(k)$ . La planta no-lineal de ecuación (4.7) se puede también expresar como:

$$y(k) = V^o \phi [W^* X(k)] - \delta(k) \quad (4.10)$$

donde  $V^o$  es un vector seleccionado por el usuario. En general,  $\|\delta(k)\| \geq \|\mu(k)\|$ . Usando la Serie de Taylor alrededor del punto  $W_k X(k)$ , el error de identificación se puede expresar de la siguiente manera:

$$\begin{aligned} e(k) &= V_k \phi [W_k X(k)] - V^o \phi [W^* X(k)] + \delta(k) \\ &= V_k \phi [W_k X(k)] - V^o \phi [W_k X(k)] + V^o \phi [W_k X(k)] - V^o \phi [W^* X(k)] + \delta(k) \\ &= \tilde{V}_k \phi [W_k X(k)] + V^o \phi \tilde{W}_k X(k) + \zeta(k) \end{aligned} \quad (4.11)$$

donde  $\phi'$  es la derivada de la función de activación no-lineal  $\phi(\cdot)$  en el punto  $W_k X(k)$ ,  $\tilde{W}_k = W_k - W^*$ ,  $\tilde{V}_k = V_k - V^o$ ,  $\zeta(k) = V^o \varepsilon(k) + \delta(k)$ , aquí  $\varepsilon(k)$  es la aproximación de segundo orden del error de la Serie de Taylor.

En este documento solamente se está interesado en la identificación en lazo abierto, se puede asumir que la planta de la ecuación (4.7) es acotada en la entrada, acotada en la salida, por lo tanto, es estable, i.e.,  $u(k)$  y  $y(k)$  en ecuación (4.7) son acotadas. Desde que  $X(k) = [u_1, u_2, u_3, \dots]^T$ ,  $X(k)$  es acotada. Debido a que la función  $\phi$  es acotada se asume que  $\delta(k)$  de la ecuación (4.10) es acotada, también  $\varepsilon(k)$  es acotado. Así  $\zeta(k)$  en ecuación (4.11) es acotada. El siguiente teorema prueba la estabilidad del algoritmo backpropagation como un algoritmo para una red neuronal multicapa en tiempo discreto [25][26].

**Teorema 4.1** *Si se usa la red neuronal multicapa de la ecuación (4.9) para identificar el proceso no-lineal de la ecuación (4.7) el siguiente algoritmo parecido al backpropagation puede hacer el error de identificación  $e(k)$  acotado.*

$$\begin{aligned} W_{k+1} &= W_k - \eta_k e(k) \phi V^{oT} X^T(k) \\ V_{k+1} &= V_k - \eta_k e(k) \phi^T \end{aligned} \quad (4.12)$$

donde  $\eta_k = \frac{\eta}{1 + \|\phi V^{oT} X^T(k)\|^2 + \|\phi\|^2}$ ,  $0 < \eta \leq 1$ . El promedio del error de identificación satisface:

$$J = \limsup_{T \rightarrow \infty} \frac{1}{T} \sum_{k=1}^T e^2(k) \leq \frac{\eta \bar{\zeta}}{\pi} \quad (4.13)$$

donde:

$$\begin{aligned} \pi &= \frac{K_a \eta}{1+k} \left[ 2 - \frac{\eta k}{1+k} \right] > 0 \\ k &= \max_k \left( \|\phi V^{oT} X^T(k)\|^2 + \|\phi\|^2 \right) \\ \bar{\zeta} &= \frac{2K_a}{1+k} \max_k [e(k) \zeta(k)] \end{aligned}$$

**Demostración.** Se selecciona la matriz positiva definida  $L_k$  como:

$$L_k = \left\| \widetilde{W}_k K_w \widetilde{W}_k^T \right\| + \left\| \widetilde{V}_k K_v \widetilde{V}_k^T \right\| \quad (4.14)$$

donde  $\widetilde{W}_k = W_k - W^*$ ,  $\widetilde{V}_k = V_k - V^o$ . Desde la ley de actualización de los pesos sinópticos de la ecuación (4.12) se tiene:

$$\begin{aligned} \widetilde{W}_{k+1} &= \widetilde{W}_k - \eta_k e(k) \phi V^{oT} X^T(k) \\ \widetilde{V}_{k+1} &= \widetilde{V}_k - \eta_k e(k) \phi^T \end{aligned}$$

desde que la matriz  $\phi'$  es una matriz diagonal, y usando la ecuación (4.11) se tiene:

$$\begin{aligned}
\Delta L_k &= \left\| \widetilde{W}_{k+1} K_w \widetilde{W}_{k+1}^T \right\| + \left\| \widetilde{V}_{k+1} K_v \widetilde{V}_{k+1}^T \right\| - \left\| \widetilde{W}_k K_w \widetilde{W}_k^T \right\| - \left\| \widetilde{V}_k K_v \widetilde{V}_k^T \right\| \\
\Delta L_k &= \left\| \left( \widetilde{W}_k - \eta_k e(k) \phi V^{oT} X^T(k) \right) K_w \left( \widetilde{W}_k - \eta_k e(k) \phi V^{oT} X^T(k) \right)^T \right\| + \\
&+ \left\| \left( \widetilde{V}_k - \eta_k e(k) \phi^T \right) K_v \left( \widetilde{V}_k - \eta_k e(k) \phi^T \right)^T \right\| - \left\| \widetilde{W}_k K_w \widetilde{W}_k^T \right\| - \left\| \widetilde{V}_k K_v \widetilde{V}_k^T \right\| \\
&= \left\| \widetilde{W}_k K_w \widetilde{W}_k^T - \widetilde{W}_k K_w \left( \eta_k e(k) \phi V^{oT} X^T(k) \right)^T - \eta_k e(k) \phi V^{oT} X^T(k) K_w \widetilde{W}_k^T + \right. \\
&\quad \left. \eta_k e(k) \phi V^{oT} X^T(k) K_w \left( \eta_k e(k) \phi V^{oT} X^T(k) \right)^T \right\| + \\
&+ \left\| \widetilde{V}_k K_v \widetilde{V}_k^T - \widetilde{V}_k K_v \left( \eta_k e(k) \phi^T \right)^T - \eta_k e(k) \phi^T K_v \widetilde{V}_k^T + \eta_k e(k) \phi^T K_v \left( \eta_k e(k) \phi^T \right)^T \right\| - \\
&\quad - \left\| \widetilde{W}_k K_w \widetilde{W}_k^T \right\| - \left\| \widetilde{V}_k K_v \widetilde{V}_k^T \right\| \\
&\leq \left\| \widetilde{W}_k K_w \widetilde{W}_k^T \right\| + \left\| -\widetilde{W}_k K_w \left( \eta_k e(k) \phi V^{oT} X^T(k) \right)^T - \eta_k e(k) \phi V^{oT} X^T(k) K_w \widetilde{W}_k^T + \right. \\
&\quad \left. \eta_k e(k) \phi V^{oT} X^T(k) K_w \left( \eta_k e(k) \phi V^{oT} X^T(k) \right)^T \right\| + \\
&\left\| \widetilde{V}_k K_v \widetilde{V}_k^T \right\| + \left\| -\widetilde{V}_k K_v \left( \eta_k e(k) \phi^T \right)^T - \eta_k e(k) \phi^T K_v \widetilde{V}_k^T + \eta_k e(k) \phi^T K_v \left( \eta_k e(k) \phi^T \right)^T \right\| - \\
&\quad - \left\| \widetilde{W}_k K_w \widetilde{W}_k^T \right\| - \left\| \widetilde{V}_k K_v \widetilde{V}_k^T \right\| \\
&\leq \left\| -\widetilde{W}_k K_w \left( \eta_k e(k) \phi V^{oT} X^T(k) \right)^T - \eta_k e(k) \phi V^{oT} X^T(k) K_w \widetilde{W}_k^T + \right. \\
&\quad \left. \eta_k e(k) \phi V^{oT} X^T(k) K_w \left( \eta_k e(k) \phi V^{oT} X^T(k) \right)^T \right\| + \\
&+ \left\| -\widetilde{V}_k K_v \left( \eta_k e(k) \phi^T \right)^T - \eta_k e(k) \phi^T K_v \widetilde{V}_k^T + \eta_k e(k) \phi^T K_v \left( \eta_k e(k) \phi^T \right)^T \right\| \\
&\leq \left\| -2\eta_k e(k) \phi V^{oT} X^T(k) K_w \widetilde{W}_k^T \right\| + \left\| \eta_k e(k) \phi V^{oT} X^T(k) K_w \left( \eta_k e(k) \phi V^{oT} X^T(k) \right)^T \right\| + \\
&\quad + \left\| -2\eta_k e(k) \phi^T K_v \widetilde{V}_k^T \right\| + \left\| \eta_k e(k) \phi^T K_v \left( \eta_k e(k) \phi^T \right)^T \right\| \\
&\quad \leq -2\eta_k \|e(k)\| \left\| \phi V^{oT} X^T(k) K_w \widetilde{W}_k^T + \phi^T K_v \widetilde{V}_k^T \right\| + \\
&\quad \left\| \eta_k e(k) \phi V^{oT} X^T(k) K_w \left( \eta_k e(k) \phi V^{oT} X^T(k) \right)^T \right\| + \left\| \eta_k e(k) \phi^T K_v \left( \eta_k e(k) \phi^T \right)^T \right\| \\
&\quad \text{Si } K_w = K_v = K_a \\
&\leq -2K_a \eta_k \|e(k)\| \left\| \left( \widetilde{W}_k X(k) V^{oT} \phi + \widetilde{V}_k \phi^T \right)^T \right\| + K_a \left\| \eta_k e(k) \phi V^{oT} X^T(k) \right\|^2 + K_a \left\| \eta_k e(k) \phi^T \right\|^2 \\
&\leq -2K_a \eta_k \left\| e^2(k) - e(k) \zeta^T(k) \right\| + K_a \eta_k^2 e^2(k) \left\| \phi V^{oT} X^T(k) \right\|^2 + K_a \eta_k^2 e^2(k) \left\| \phi^T \right\|^2 \\
&\leq -K_a \eta_k e^2(k) \left[ 2 - \eta_k \left( \left\| \phi V^{oT} X^T(k) \right\|^2 + \left\| \phi \right\|^2 \right) \right] + 2K_a \eta_k e(k) \zeta^T(k) \\
&\leq -\frac{K_a \eta}{1+k} \left[ 2 - \frac{\eta k}{1+k} \right] e^2(k) + \eta \frac{2K_a}{1+k} e(k) \zeta^T(k) \\
\Delta L_k &\leq -\pi e^2(k) + \eta \bar{\zeta} \tag{4.15}
\end{aligned}$$

donde  $\pi$  se define en la ecuación (4.13). Desde (4.11) y (4.14) se tiene que  $V_k$  esta en función de  $e(k)$  y  $\zeta(k)$ , así se admite que  $L_k$  es una función de Lyapunov suave, la dinámica del error es entrada-a-estado estable. Debido a que la "ENTRADA" $\zeta(k)$  es acotada, el "ESTADO" $e(k)$  es acotado, lo anterior se comprueba en la siguiente desarrollo, la ecuación (4.15) se puede reescribir como:

$$\Delta L_k \leq -\pi e^2(k) + \eta \zeta^2(k) \leq -\pi e^2(k) + \eta \bar{\zeta} \quad (4.16)$$

Considerando el intervalo de  $L_k$  de la ecuación (4.16) desde 1 a  $T$ , y considerando que  $L_T > 0$  y que  $L_1$  es una constante se tiene:

$$\begin{aligned} L_T - L_1 &\leq -\pi \sum_{k=1}^T e^2(k) + T\eta\bar{\zeta} \\ \pi \sum_{k=1}^T e^2(k) &\leq L_1 - L_T + T\eta\bar{\zeta} \leq L_1 + T\eta\bar{\zeta} \leq T\eta\bar{\zeta} \\ \frac{1}{T} \sum_{k=1}^T e^2(k) &\leq \frac{\eta\bar{\zeta}}{\pi} \end{aligned}$$

de donde el se tiene el promedio del error de identificación de la ecuación (4.13) el cual es estable, con el desarrollo anterior se comprueba el teorema. ■

**Comentario 4.1**  $V^\circ$  no tiene efecto sobre la propiedad de estabilidad de la neuroidentificación, pero esta tiene influencia sobre la exactitud de la identificación, ver ecuación (4.13). Se ha diseñado un método off-line (fuera de línea) para encontrar el mejor valor de  $V^\circ$ . Si se fija  $V^\circ = V_0$ , el algoritmo de la ecuación (4.12) puede hacer convergente el error de identificación, i.e.,  $V_k$  hará el error de identificación más pequeño que el de  $V_0$ ,  $V^\circ$  se puede seleccionar empleando los siguientes pasos:

1. Inicia desde cualquier valor inicial para  $V^\circ = V_0$ .
2. Hacer la identificación con este  $V^\circ$  hasta  $T_0$ .
3. Si  $\|e(T_0)\| < \|e(0)\|$ , entonces  $V_T$  es el nuevo  $V^\circ$ , i.e.,  $V^\circ = V_{T_0}$ , ir al paso 2 y repetir el proceso de identificación.

4. Si  $\|e(T_0)\| \geq \|e(0)\|$ , termina la identificación off-line (fuera de línea), ahora  $V_{T_0}$  es el valor final para  $V^\circ$ .

Conociendo con anterioridad  $V^\circ$ , se puede iniciar la identificación con el algoritmo de la ecuación (4.12).

#### 4.4. Obtención de los datos desde una base de datos reales

Para los datos reales se tiene el reporte de medición tomado diariamente del mes de noviembre del 2002 para el proceso de mezclado de la Figura 3.5 tomando en cuenta las siguientes equivalencias en los símbolos:

$$\begin{aligned}
 S_1 &= L_3, \quad S_2 = \text{Puerto Ceiba (PC)}, \quad S_3 = \text{Maya (M)}, \quad S_4 = \text{El Golpe (EG)} \\
 S_{M1} &= Q_a, \quad S_{M2} = Q_c, \quad S_{M3} = \text{Recibo en Tanques (RT)} \\
 S_{P1} &= S_{P1}, \quad S_{P2} = Q_b, \quad S_{P3} = \text{Envío a Nueva Etapa}, \quad S_{P4} = Q_d \\
 b &= b \text{ (Fracción másica de agua en la entrada)} \\
 d &= d \text{ (Fracción másica de agua en la salida)}
 \end{aligned} \tag{4.17}$$

Los segundos símbolos después del signo de igualdad corresponden a los símbolos que se emplean a partir de este momento, ya que estos se emplean en el programa de Excel y son los datos que vienen en el reporte diario del mes de Noviembre del 2002; Puerto Ceiba, Maya y El Golpe corresponden a tipos especiales de crudo que se caracterizan por tener un valor diferente de  $^\circ API$  (donde  $^\circ API$  es una medida indirecta de la densidad empleada en el Instituto Mexicano del Petróleo) [15].

Los datos de densidad en  $^\circ API$  para cada etapa del proceso de la Figura 3.5 vienen reportados en el reporte diario del mes de Noviembre del 2002.

La cantidad de flujo para cada etapa del proceso de la Figura 3.5 se obtiene de la siguiente

manera:

Dispositivo	Flujo ( $m^3/h$ )	
Mezclador $M3$	$S_{P4} =$ (Recibo en Tanques) $-$ (El Golpe)	
Divisor de corriente	$Q_c = Q_d +$ (Envío a Nueva Etapa)	(4.18)
Mezclador $M2$	$Q_b = Q_c -$ (Maya)	
Deshidratador	$Q_a = \frac{1-d}{1-b} Q_b$	
Mezclador $M1$	$L_3 = Q_a -$ (Puerto Ceiba)	

Note que la ecuación para el deshidratador es no lineal, esta ecuación es la que se describió como la ecuación (3.46) en donde viene la obtención detallada de esta ecuación.

Se obtiene el dato de la no linealidad para la densidad de salida de cada mezclador del modelo distribuido para posteriormente enviar los datos a un programa de Matlab. Los datos se obtienen de la siguiente manera:

Dispositivo	No linealidad de la densidad en $^{\circ}API$			
Mezclador $M1$	$NLD1 = Q_{a^{\circ}API} -$	$\left[ \left( \frac{L_{3f}}{Q_{af}} \right) * L_{3^{\circ}API} \right] -$	$\left[ \left( \frac{PC_f}{Q_{af}} \right) * PC_{^{\circ}API} \right]$	(4.19)
Mezclador $M2$	$NLD2 = Q_{c^{\circ}API} -$	$\left[ \left( \frac{Q_{bf}}{Q_{cf}} \right) * Q_{b^{\circ}API} \right] -$	$\left[ \left( \frac{M_f}{Q_{cf}} \right) * M_{^{\circ}API} \right]$	
Mezclador $M3$	$NLD3 = RT_{^{\circ}API} -$	$\left[ \left( \frac{Q_{df}}{RT_f} \right) * Q_{d^{\circ}API} \right] -$	$\left[ \left( \frac{EG_f}{RT_f} \right) * EG_{^{\circ}API} \right]$	

Se obtienen las fracciones de flujo a la entrada de cada mezclador del modelo distribuido para posteriormente enviar los datos a un programa de Matlab. Los datos se obtienen de la siguiente manera:

Dispositivo	Fracción de flujo en entrada 1 (s/u)	Fracción de flujo en entrada 2 (s/u)	
Mezclador $M1$	$u(1, k) = \frac{L_{3f}}{Q_{af}}$	$u(2, k) = \frac{PC_f}{Q_{af}}$	(4.20)
Mezclador $M2$	$u(1, k) = \frac{Q_{bf}}{Q_{cf}}$	$u(2, k) = \frac{M_f}{Q_{cf}}$	
Mezclador $M3$	$u(1, k) = \frac{Q_{df}}{RT_f}$	$u(2, k) = \frac{EG_f}{RT_f}$	

Se obtiene el dato de la no linealidad para la densidad de salida de todos los mezcladores del modelo agregado para posteriormente enviar el dato a un programa de Matlab. El dato

se obtiene de la siguiente manera:

$$\begin{aligned}
 & \text{No linealidad de la densidad en } ^{\circ}API \\
 NLA &= RT_{\circ API} - \left[ \left( \frac{L_{3f}}{prom} \right) * L_{3^{\circ}API} \right] - \left[ \left( \frac{PC_f}{prom} \right) * PC_{\circ API} \right] \\
 & - \left[ \left( \frac{M_f}{prom} \right) * M_{\circ API} \right] - \left[ \left( \frac{EG_f}{prom} \right) * EG_{\circ API} \right] \\
 prom &= L_{3f} + PC_f + M_f + EG_f
 \end{aligned} \tag{4.21}$$

Se obtienen las fracciones de flujo en cada entrada para todos los mezcladores en el modelo agregado para posteriormente enviar los datos a un programa de Matlab. Los datos se obtienen de la siguiente manera:

$$\begin{array}{cccc}
 \text{Flujo, entrada 1 (s/u)} & \text{Flujo, entrada 2 (s/u)} & \text{Flujo, entrada 3 (s/u)} & \text{Flujo, entrada 4 (s/u)} \\
 u(1, k) = \frac{L_{3f}}{prom} & u(2, k) = \frac{PC_f}{prom} & u(3, k) = \frac{M_f}{prom} & u(4, k) = \frac{EG_f}{prom}
 \end{array}$$

Se extraen los datos del programa de Excel hacia programas de Matlab con la instrucción:

$$A = xlsread(data)$$

donde  $A$  es la matriz donde se alojan en Matlab todos los datos numéricos,  $data$  es el nombre del programa en Excel (Ver Figura 4.2) [22].

Las propiedades estáticas en una mezcla de petróleo crudo se describen mediante la ecuación (3.14) [18], la cual se enuncia de nuevo a continuación:

$$M = \sum x_i M_i + \Delta M$$

donde:

$\Delta M$  cambio de propiedad de la mezcla.

$x_i$  fracción mol del componente  $i$  en la solución.

$\Delta M$  se obtiene mediante la ecuación (3.15) la cual se enuncia a continuación:

$$\Delta M = M - \sum x_i M_i^o$$

donde:



Figura 4.2: Comunicación Excel-Matlab

$M_i^o$  propiedad molar estándar del componente  $i$  en un estado estándar especificado.

Para este documento interesa la propiedad de la densidad.

Para el caso de los modelos distribuido y agregado y si todos los componentes existen en un estado líquido a la misma presión y temperatura como en solución, entonces  $M_i^o$  se convierte en:

$$M_i^o = \rho_i$$

donde:

$\rho_i$  densidad para cada componente a la entrada del mezclador, dada en  $^oAPI$ .

y  $x_i$  se convierte a:

$$x_i = \frac{f_{in}}{\sum f_{in}} = u_i$$

donde:

$f_{in}$  flujo de cada entrada del mezclador, dado en  $m^3/h$ .

$u_i$  flujo parcial de cada componente, sin unidades.

Entonces la ecuación (3.15) se convierte en la siguiente ecuación:

$$\Delta\varrho = \varrho - \sum u_i\varrho_i \quad (4.22)$$

donde

$\varrho$  densidad a la salida del mezclador, en la caso agregado es la densidad a la salida del ultimo mezclador, dada en  $^{\circ}API$ .

$\Delta\varrho$  no linealidad de la densidad en la mezcla, dada en  $^{\circ}API$ .

y la ecuación (3.14) se convierte en la siguiente ecuación:

$$\varrho = \sum u_i\varrho_i + \Delta\varrho \quad (4.23)$$

La ecuación (4.23) describe el comportamiento de la densidad de la mezcla por cada día, y la ecuación (4.22) describe el comportamiento de la no linealidad en la mezcla por cada día, para este documento se tiene el dato de 26 días.

Por ejemplo, para el caso del modelo distribuido del mezclador 1 se tienen las siguientes ecuaciones:

$$x_i = \frac{f_{in}}{2}$$

$$\sum_{i=1}^2 f_{in}$$

$$\Delta\varrho = \varrho - \sum_{i=1}^2 u_i\varrho_i$$

$$\varrho = \sum_{i=1}^2 u_i\varrho_i + \Delta\varrho$$

y se tiene la repuesta dada por la Figura 4.3.

Para el caso del modelo agregado se tienen las siguientes ecuaciones:

$$x_i = \frac{f_{in}}{4}$$

$$\sum_{i=1}^4 f_{in}$$

$$\Delta\varrho = \varrho - \sum_{i=1}^4 u_i\varrho_i$$

$$\varrho = \sum_{i=1}^4 u_i\varrho_i + \Delta\varrho$$

y se tiene la repuesta dada por la Figura 4.4.

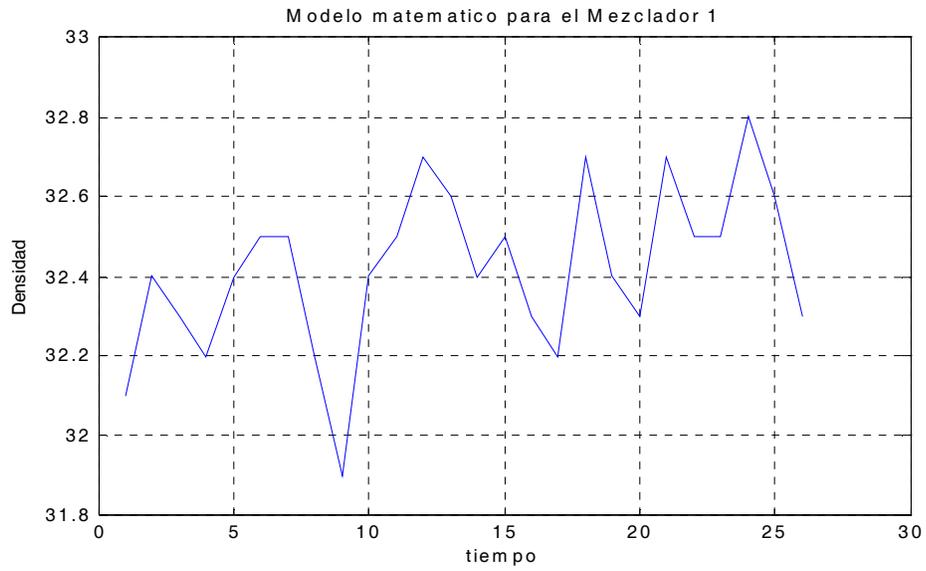


Figura 4.3: Densidad a la salida del mezclador 1

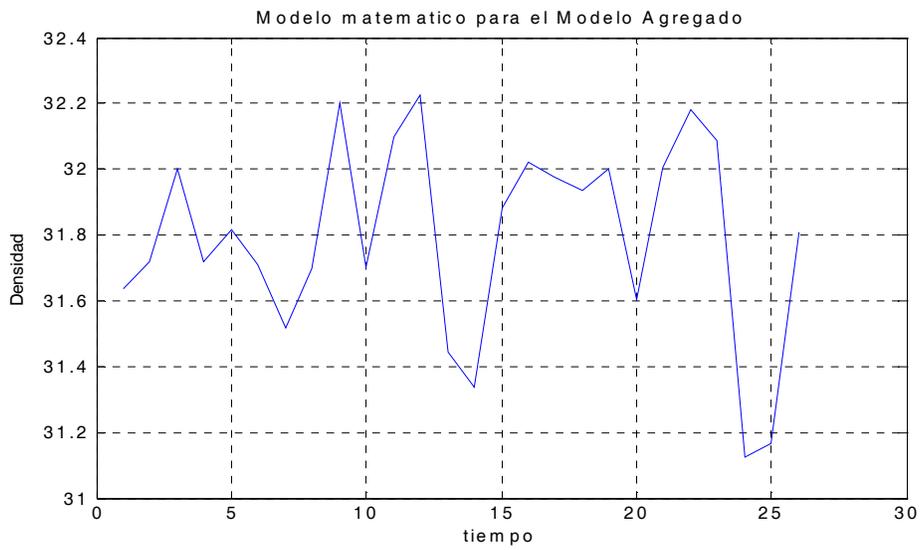


Figura 4.4: Densidad de salida del modelo agregado

## 4.5. Modelación del modelo distribuido

Para el modelo distribuido se consideran cada uno de los 3 mezcladores por separado (ver Figuras 3.5 y 3.6) teniéndose un programa para cada mezclador. Se obtienen 26 datos de una base de datos reales los cuales se emplean para realizar cada programa. Se consideran 3 programas para identificación de parámetros vía least square (uno por cada mezclador), se tienen 6 programas para identificación de parámetros vía redes neuronales debido a que hay dos modalidades que son para el modelo completo y para la no linealidad las cuales se explicarán más adelante.

### 4.5.1. Identificación de parámetros vía least square

Basta con explicar solamente un programa dado que los tres programas de identificación de parámetros vía Least square son idénticos, con la diferencia que cada programa contiene los datos correspondientes a cada mezclador.

Para este programa se emplea la formula descrita por la ecuación (4.3) para el Least square monovariante la cual se vuelve a citar a continuación:

$$\begin{aligned}
 \alpha &= (X^T X)^{-1} Y^T X \\
 X &= [x(1), x(2), \dots, x(26)] \\
 Y &= [y(1), y(2), \dots, y(26)] \\
 x(k) &= u(1, k) * u(2, k), \quad k = 1, 2, \dots, 26 \\
 y1(k) &= \alpha * x(k), \quad k = 1, 2, \dots, 26
 \end{aligned} \tag{4.24}$$

donde  $u(1, k)$  es la fracción de flujo de la entrada 1 para cada mezclador,  $u(2, k)$  es la fracción de flujo de la entrada 2 para cada mezclador,  $\alpha$  es el parámetro a identificar vía least square,  $y(k)$  es la salida proporcionada por los datos reales y  $y1(k)$  es la salida identificada vía least square.

Posteriormente se grafican los resultados, que son tres gráficas en una Figura y una Figura para cada mezclador (ver Figuras 4.5, 4.6 y 4.7):

- La primer gráfica compara la señal deseada de los datos proporcionados por el programa de Excel contra señal de salida que se tiene con la identificación vía Least square.
- La segunda gráfica los puntos de la entrada contra la salida para el programa de Excel, también gráfica en una línea recta la pendiente que se obtiene vía Least square la cual es el valor del parámetro  $\alpha$ , donde  $\alpha$  es el parámetro de iteración del modelo de iteración.
- La tercer gráfica proporciona el error promedio cuadrático de la señal deseada y la señal de salida para el algoritmo de Least square.

A continuación se analizan las gráficas de los programas de Matlab de Least square para cada mezclador:

De la gráfica del inciso (a) de la Figura 4.5 donde se encuentran la señal deseada con la estimada del mezclador 1 se observa que la señal estimada es una línea recta que pasa por en medio de todos los valores de la señal deseada. De la gráfica del inciso (b) de la Figura 4.5 donde se encuentra la pendiente para el mezclador 1 proporciona una pendiente muy cercana a cero. De la gráfica del inciso (c) de la Figura 4.5 donde se encuentra el error promedio cuadrático del mezclador 1 se observa que el valor del error promedio cuadrático es de 0,2046.

De la gráfica del inciso (a) de la Figura 4.6 donde se encuentran la señal deseada con la estimada del mezclador 2 se observa que la señal estimada es una línea recta que pasa por en medio de todos los valores de la señal deseada. De la gráfica del inciso (b) de la Figura 4.6 donde se encuentra la pendiente para el mezclador 2 proporciona una pendiente muy cercana a cero. De la gráfica del inciso (c) de la Figura 4.6 donde se encuentra el error promedio cuadrático del mezclador 2 se observa que el valor del error promedio cuadrático es de 0,0869.

De la gráfica del inciso (a) de la Figura 4.7 donde se encuentran la señal deseada con la estimada del mezclador 3 se observa que las señales son muy aproximadas debido a que la señal deseada el altamente lineal. De la gráfica del inciso (b) de la Figura 4.7 donde se encuentra la pendiente para el mezclador 3 proporciona una pendiente muy cercana a cero.

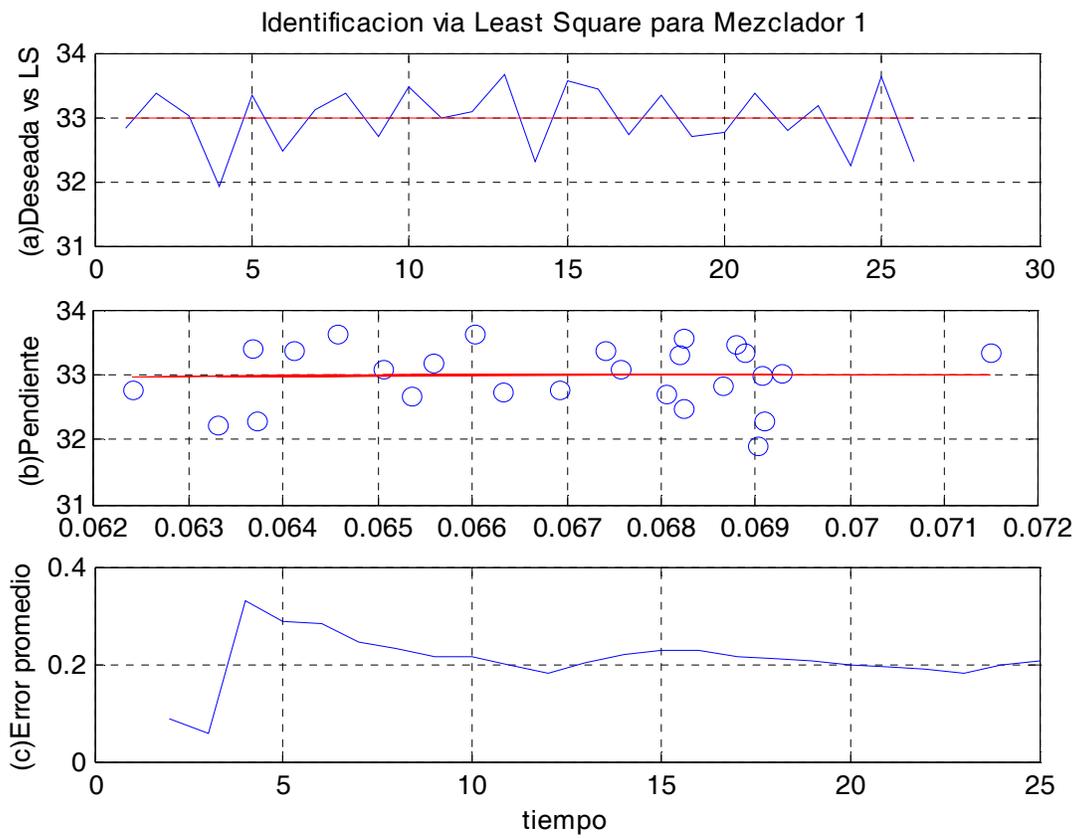


Figura 4.5: Least square para el mezclador 1

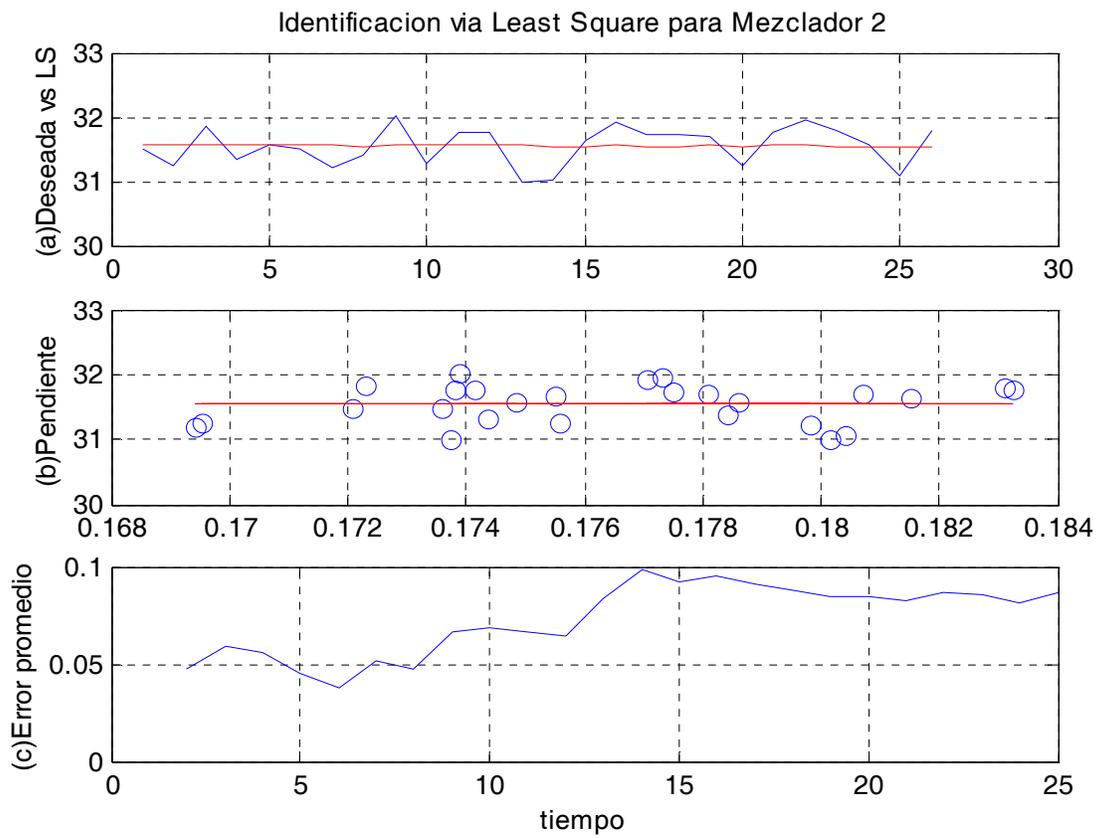


Figura 4.6: Least square para el mezclador 2

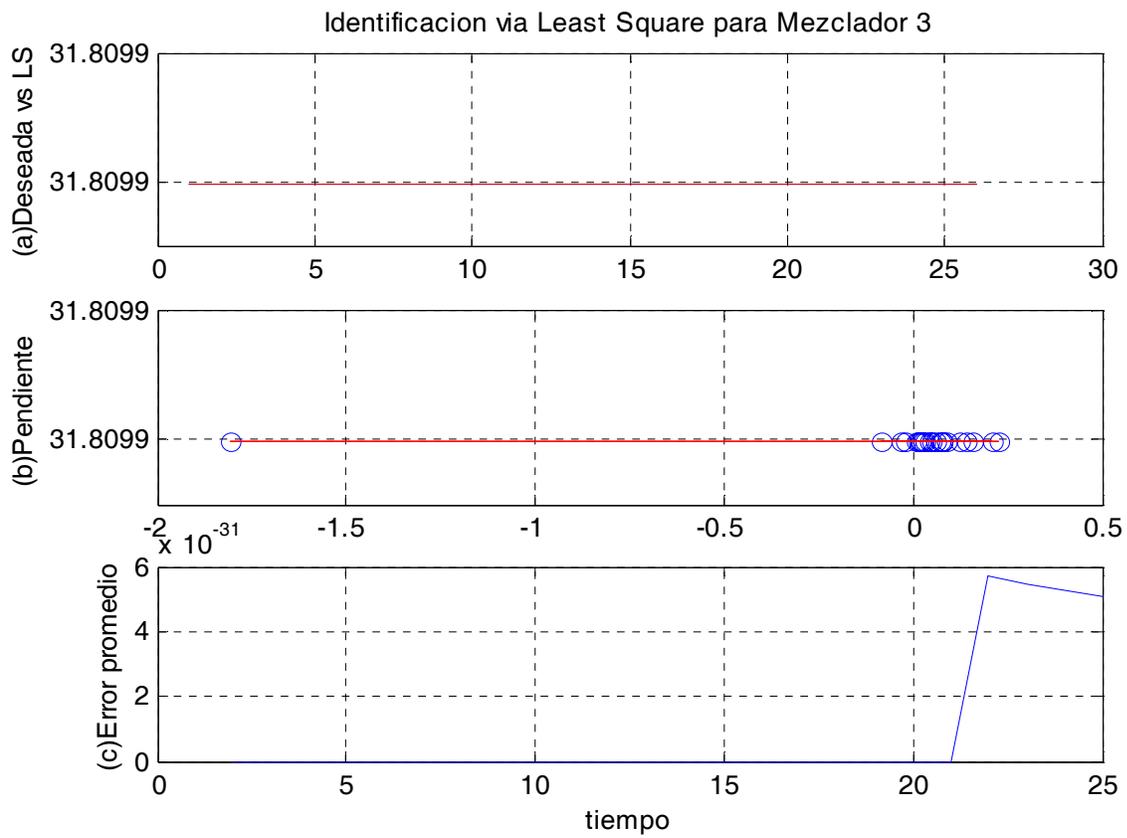


Figura 4.7: Least square para el mezclador 3

De la gráfica del inciso (c) de la Figura 4.7 donde se encuentra el error promedio cuadrático del mezclador 3 se observa que el valor del error promedio cuadrático es de  $5,0487e - 031$ .

### 4.5.2. Identificación vía redes neuronales para el modelo completo

Basta con explicar solamente un programa dado que los tres programas de identificación de parámetros vía redes neuronales son idénticos, con la diferencia que cada programa contiene los datos correspondientes a cada mezclador.

Para este programa se emplea el algoritmo denominado de aprendizaje con parámetros de aprendizaje variables descrito al principio de éste capítulo, este algoritmo es parecido al algoritmo backpropagation, de hecho en el camino hacia adelante son idénticos, el camino hacia adelante es obtener la salida de cada neurona a partir de sus entradas, para esto se emplea la formula (2.1) la cual se vuelve a citar a continuación para la  $i$ -ésima neurona:

$$u_i = \sum_{j=1}^n \omega_{ij} x_j$$

$$y_i = \phi(u_i - \rho_i)$$

donde:

$x_j$      $j$ -ésimo componente de la entrada.

$\omega_{ij}$     peso de la conexión entre la  $j$ -ésima componente de la entrada y la  $i$ -ésima neurona.

$u_i$     salida del sumador.

$\rho_i$     umbral.

$\phi(\cdot)$     función de activación no lineal.

$y_i$     salida de la  $i$ -ésima neurona.

En este caso no se emplea el umbral quedando la salida de cada neurona como sigue:

$$y_i = \phi(u_i)$$

Para el modelo distribuido se tienen dos entradas que son  $u(1, k)$  y  $u(2, k)$ .

Para el caso del camino hacia atrás se tiene como se propaga el error de la salida a través de cada neurona en la red neuronal, la diferencia entre el algoritmo de aprendizaje

con parámetros de aprendizaje variables y el backpropagation estriba en que en el caso del algoritmo de aprendizaje con parámetros de aprendizaje variables emplea el parámetro de aprendizaje que varía con cada iteración y además, en éste se puede comprobar la estabilidad.

La actualización de los pesos sinópticos se describe mediante la ecuación (4.12) la cual se vuelve a citar a continuación:

$$\begin{aligned} W_{k+1} &= W_k - \eta_k e(k) \phi V^o T X^T(k) \\ V_{k+1} &= V_k - \eta_k e(k) \phi^T \end{aligned}$$

donde la salida escalar de la red neuronal es  $\hat{y}(k) \in R^{n \times 1}$ , el vector de entrada a la red es  $X(k) \in R^{n \times 1}$  definida en ecuación (4.8) donde se emplean las dos entradas que son  $u(1, k)$  y  $u(2, k)$ , los pesos sinópticos en la capa de salida son  $V_k \in R^{1 \times m}$ , los pesos en la capa oculta son  $W_k \in R^{m \times n}$ ,  $\phi$  es una función vectorial m-dimensional, la representación típica de la función  $\phi_i(\cdot)$  es una función sigmoideal, pero para propósitos del programa se selecciona la función tangente hiperbólica ya que esta además de ser derivable (se emplea la derivada en el algoritmo) también es asimétrica por lo que considera valores menores y mayores a cero proporcionando mejores resultados en la simulación. El error de identificación en la capa de salida se define como:

$$e(k) = \hat{y}(k) - y(k)$$

donde  $\eta$  es el parámetro de aprendizaje inicial que tiene el valor de 1.  $V^o$  es un vector seleccionado por el usuario para calibrar la red que para propósitos del programa en Matlab se pone como  $V^o = \text{rand}$  lo cual quiere decir que puede tomar cualquier valor entre 0 y 1.

Para el entrenamiento y examinación se emplean las siguientes fórmulas:

$$\begin{aligned} U &= [u(1, k), u(2, k)], & k &= 1, 2, \dots, 26 \\ DE &= [DE(1, k), DE(2, k)], & k &= 1, 2, \dots, 26 \\ y(k) &= (DE * U^T) + NLD1(1, k), & k &= 1, 2, \dots, 26 \\ yn(k) &= (DE * U^T) + yn(k), & k &= 1, 2, \dots, 26 \end{aligned} \tag{4.25}$$

donde  $DE(1, k)$  es la densidad de la primer entrada para cada mezclador,  $DE(2, k)$  es la densidad de la segunda entrada para cada mezclador,  $NLD1(1, k)$  es la no linealidad del

modelo distribuido para el mezclador1,  $u(1, k)$  es la fracción de flujo de la entrada 1 para cada mezclador,  $u(2, k)$  es la fracción de flujo de la entrada 2 para cada mezclador,  $y(k)$  es la salida deseada para cada mezclador y  $yn(k)$  es la salida de la red neuronal para cada mezclador.

Posteriormente se grafican los resultados, que son cuatro gráficas en una Figura y una Figura para cada mezclador (ver Figuras 4.8, 4.9 y 4.10):

- La primer gráfica compara la señal deseada de los datos proporcionados por el programa de Excel contra señal de salida que se tiene con la identificación vía Redes Neuronales para el caso del entrenamiento.
- La segunda gráfica compara la señal deseada de los datos proporcionados por el programa de Excel contra señal de salida que se tiene con la identificación vía Redes Neuronales para el caso de la generalización.
- La tercer gráfica proporciona el error promedio cuadrático de la señal deseada y la señal de salida para el entrenamiento del algoritmo de Redes Neuronales.
- La cuarta gráfica proporciona el error promedio cuadrático de la señal deseada y la señal de salida para la generalización del algoritmo de Redes Neuronales.

A continuación se analizan las gráficas de los programas de Matlab de Redes neuronales para el modelo completo de cada mezclador:

En la Figura 4.8 se simula una red neuronal con parámetro de aprendizaje variable, 2 entradas, 1 salida, 5 neuronas en la capa oculta, una ganancia para la función tangente hiperbólica de 4, 20 patrones con 100 épocas para el entrenamiento y 6 patrones con 50 épocas para la generalización. De la gráfica del inciso (a) de la Figura 4.8 donde se encuentran la señal deseada con la estimada para el entrenamiento del mezclador 1 se observa que ambas señales son muy aproximadas teniéndose por consiguiente un buen entrenamiento. De la gráfica del inciso (b) de la Figura 4.8 donde se encuentran la señal deseada con la estimada para la generalización del mezclador 1 se observa que ambas señales son aproximadas teniéndose por

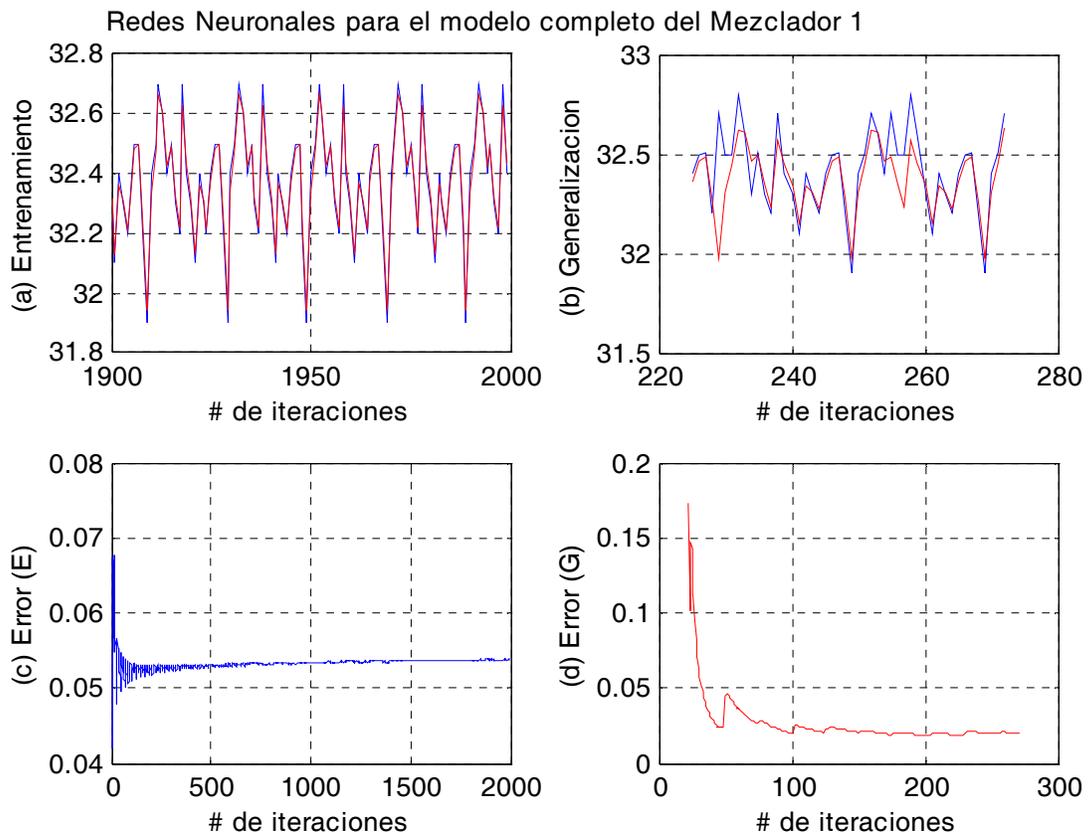


Figura 4.8: Red neuronal para el modelo completo del mezclador 1

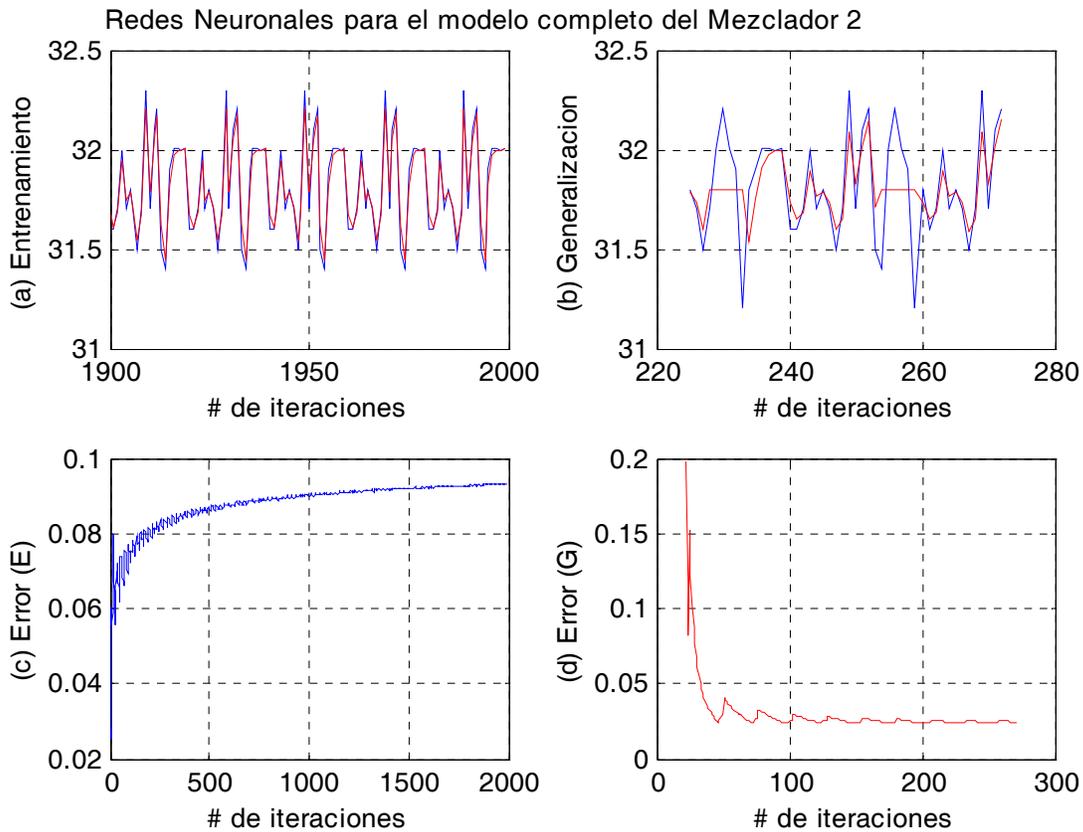


Figura 4.9: Red neuronal para el modelo completo del mezclador 2

consiguiente una buena generalización. De la gráfica del inciso (c) de la Figura 4.8 donde se encuentra el error promedio cuadrático para el entrenamiento del mezclador 1 se observa que el valor del error promedio cuadrático de entrenamiento es de 0,0538. De la gráfica del inciso (d) de la Figura 4.8 donde se encuentra el error promedio cuadrático para la generalización del mezclador 1 se observa que el valor del error promedio cuadrático de generalización es de 0,0197.

En la Figura 4.9 se simula una red neuronal con parámetro de aprendizaje variable, 2 entradas, 1 salida, 5 neuronas en la capa oculta, una ganancia para la función tangente

hiperbólica de 4, 20 patrones con 100 épocas para el entrenamiento y 6 patrones con 50 épocas para la generalización. De la gráfica del inciso (a) de la Figura 4.9 donde se encuentran la señal deseada con la estimada para el entrenamiento del mezclador 2 se observa que ambas señales son muy aproximadas teniéndose por consiguiente un buen entrenamiento. De la gráfica del inciso (b) de la Figura 4.9 donde se encuentran la señal deseada con la estimada para la generalización del mezclador 2 se observa que ambas señales son aproximadas teniéndose por consiguiente una buena generalización. De la gráfica del inciso (c) de la Figura 4.9 donde se encuentra el error promedio cuadrático para el entrenamiento del mezclador 2 se observa que el valor del error promedio cuadrático de entrenamiento es de 0,0931. De la gráfica del inciso (d) de la Figura 4.9 donde se encuentra el error promedio cuadrático para la generalización del mezclador 2 se observa que el valor del error promedio cuadrático de generalización es de 0,0243.

En la Figura 4.10 se simula una red neuronal con parámetro de aprendizaje variable, 2 entradas, 1 salida, 5 neuronas en la capa oculta, una ganancia para la función tangente hiperbólica de 4, 20 patrones con 100 épocas para el entrenamiento y 6 patrones con 50 épocas para la generalización. De la gráfica del inciso (a) de la Figura 4.10 donde se encuentran la señal deseada con la estimada para el entrenamiento del mezclador 3 se observa que ambas señales son muy aproximadas teniéndose por consiguiente un buen entrenamiento. De la gráfica del inciso (b) de la Figura 4.10 donde se encuentran la señal deseada con la estimada para la generalización del mezclador 3 se observa que ambas señales son aproximadas teniéndose por consiguiente una buena generalización. De la gráfica del inciso (c) de la Figura 4.10 donde se encuentra el error promedio cuadrático para el entrenamiento del mezclador 3 se observa que el valor del error promedio cuadrático de entrenamiento es de 0,0867. De la gráfica del inciso (d) de la Figura 4.10 donde se encuentra el error promedio cuadrático para la generalización del mezclador 3 se observa que el valor del error promedio cuadrático de generalización es de 0,0433.

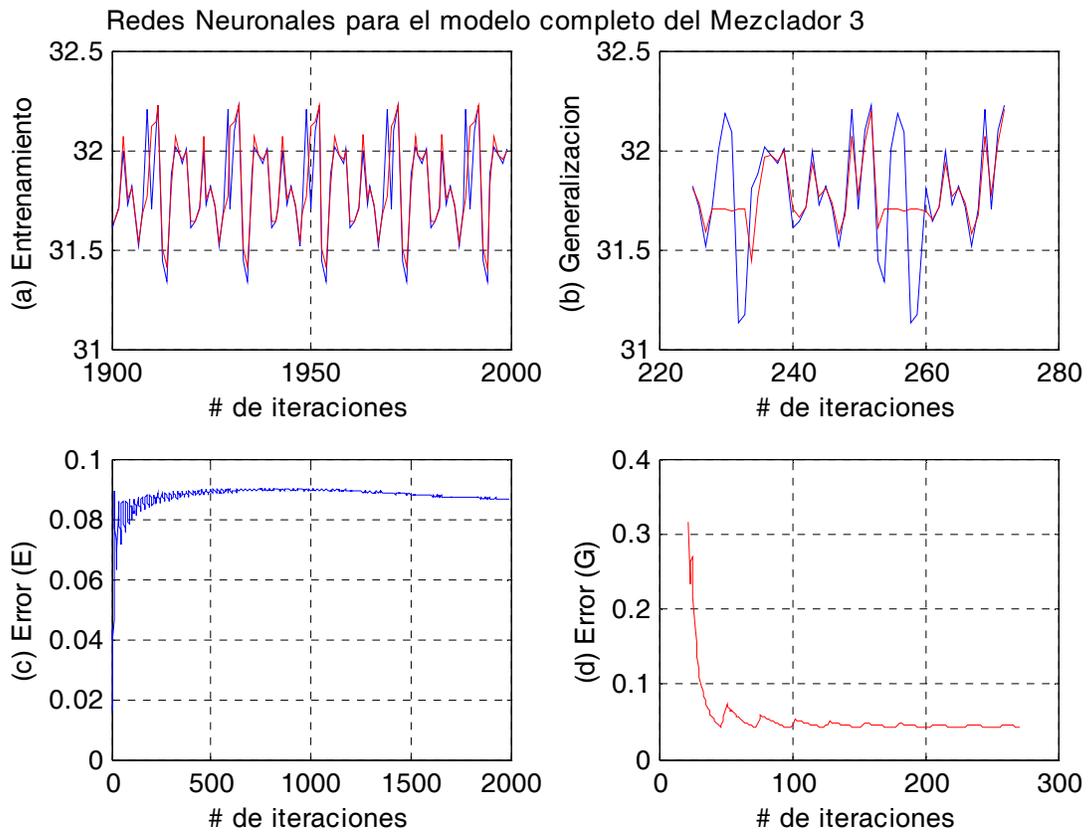


Figura 4.10: Red neuronal para el modelo completo del mezclador 3

### 4.5.3. Identificación vía redes neuronales para la parte no lineal

Basta con explicar solamente un programa dado que los tres programas de identificación de parámetros vía Redes Neuronales son idénticos, con la diferencia que cada programa contiene los datos correspondientes a cada mezclador.

Este programa es idéntico al programa de redes neuronales para el modelo completo con la siguiente diferencia:

Para el entrenamiento se emplean las siguientes fórmulas:

$$\begin{aligned}
 U &= [u(1, k), u(2, k)], & k &= 1, 2, \dots, 26 \\
 DE &= [DE(1, k), DE(2, k)], & k &= 1, 2, \dots, 26 \\
 y(k) &= (NLD1(1, k) * P), & k &= 1, 2, \dots, 26 \\
 yn(k) &= yn(k), & k &= 1, 2, \dots, 26
 \end{aligned}
 \tag{4.26}$$

donde  $DE(1, k)$  es la densidad de la primer entrada para cada mezclador,  $DE(2, k)$  es la densidad de la segunda entrada para cada mezclador,  $NLD1(1, k)$  es la no linealidad del modelo distribuido para el mezclador 1,  $u(1, k)$  es la fracción de flujo de la entrada 1 para cada mezclador,  $u(2, k)$  es la fracción de flujo de la entrada 2 para cada mezclador,  $y(k)$  es la salida deseada del entrenamiento para cada mezclador,  $yn(k)$  es la salida de la red neuronal del entrenamiento para cada mezclador,  $P$  es una ganancia para la salida deseada, este parámetro se emplea debido a que  $NLD1(1, k)$  es muy pequeño y se necesita amplificar para que la función tangente hiperbólica pueda operar adecuadamente.

Para la examinación se emplean las siguientes fórmulas:

$$\begin{aligned}
 ya(k) &= (DE * U^T) + NLD1(1, k), & k &= 1, 2, \dots, 26 \\
 yna(k) &= (DE * U^T) + (yn(k)/P), & k &= 1, 2, \dots, 26
 \end{aligned}
 \tag{4.27}$$

Note que en esta etapa se le divide por el parámetro  $P$  debido a que este parámetro se multiplica a la salida deseada para tener un mejor aprendizaje,  $ya(k)$  es la salida deseada de la examinación para el modelo,  $yna(k)$  es la salida de la red neuronal de la examinación para el modelo.

Posteriormente se grafican los resultados, que son cuatro gráficas en una Figura y una Figura para cada mezclador (ver Figuras 4.11, 4.12 y 4.13):

- La primer gráfica compara la señal deseada de los datos proporcionados por el programa de Excel contra señal de salida que se tiene con la identificación vía Redes Neuronales para el caso del entrenamiento.
- La segunda gráfica compara la señal deseada de los datos proporcionados por el programa de Excel contra señal de salida que se tiene con la identificación vía Redes Neuronales para el caso de la generalización.
- La tercer gráfica proporciona el error promedio cuadrático de la señal deseada y la señal de salida para el entrenamiento del algoritmo de Redes Neuronales.
- La cuarta gráfica proporciona el error promedio cuadrático de la señal deseada y la señal de salida para la generalización del algoritmo de Redes Neuronales.

A continuación se analizan las gráficas de los programas de Matlab de Redes neuronales para la no linealidad de cada mezclador:

En la Figura 4.11 se simula una red neuronal con parámetro de aprendizaje variable, 2 entradas, 1 salida, 5 neuronas en la capa oculta, una ganancia para la función tangente hiperbólica de 0.001, una ganancia de la salida deseada de 0.01, 20 patrones con 100 épocas para el entrenamiento y 6 patrones con 3 épocas para la generalización. De la gráfica del inciso (a) de la Figura 4.11 donde se encuentran la señal deseada con la estimada para el entrenamiento del mezclador 1 se observa que ambas señales son muy aproximadas teniéndose por consiguiente un buen entrenamiento. De la gráfica del inciso (b) de la Figura 4.11 donde se encuentran la señal deseada con la estimada para la generalización del mezclador 1 se observa que ambas señales son muy aproximadas teniéndose por consiguiente una buena generalización. De la gráfica del inciso (c) de la Figura 4.11 donde se encuentra el error promedio cuadrático para el entrenamiento del mezclador 1 se observa que el valor del error promedio cuadrático de entrenamiento es de 0,2314. De la gráfica del inciso (d) de la Figura 4.11 donde se encuentra el error promedio cuadrático para la generalización del mezclador 1 se observa que el valor del error promedio cuadrático de generalización es de 0,1053.

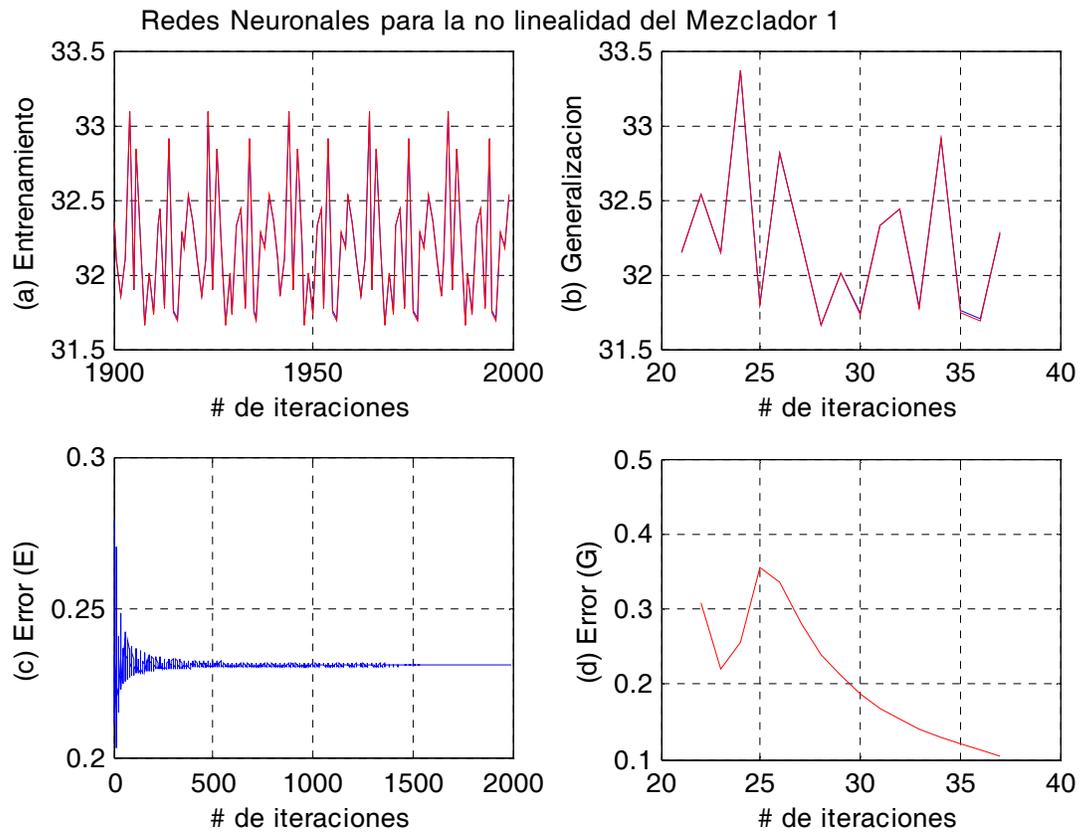


Figura 4.11: Red neuronal para la no linealidad del mezclador 1

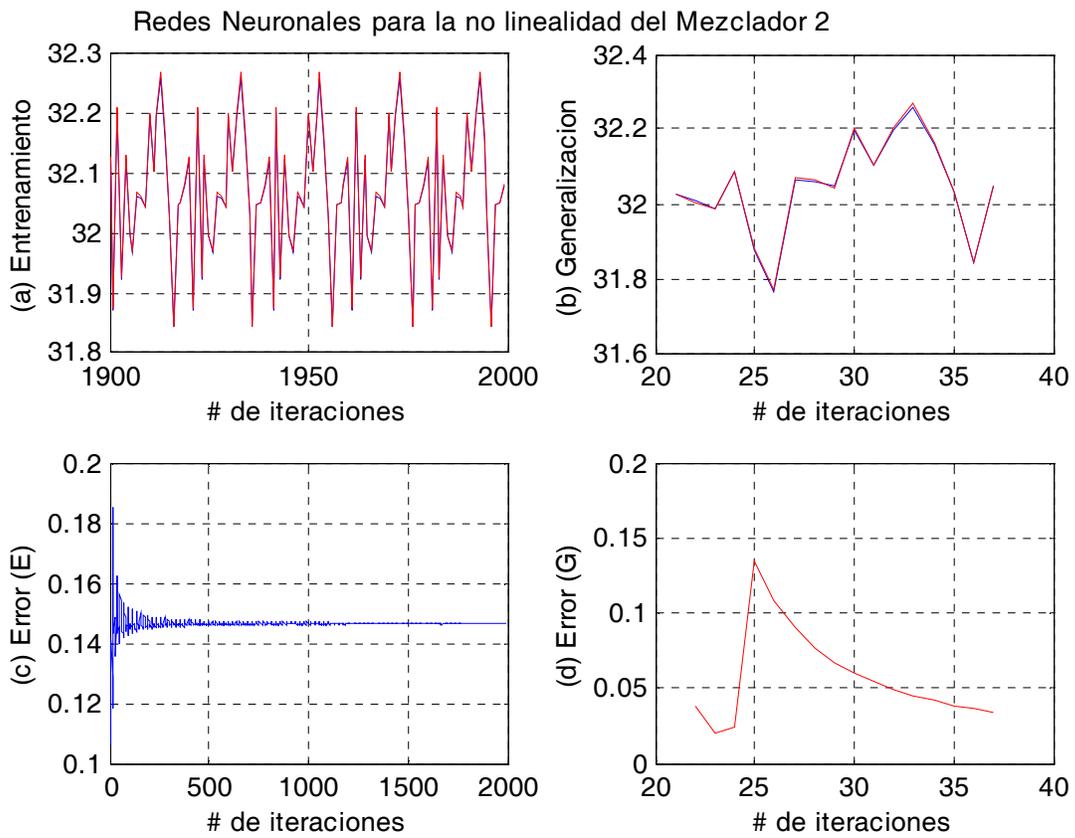


Figura 4.12: Red neuronal para la no linealidad del mezclador 2

En la Figura 4.12 se simula una red neuronal con parámetro de aprendizaje variable, 2 entradas, 1 salida, 5 neuronas en la capa oculta, una ganancia para la función tangente hiperbólica de 0.001, una ganancia de la salida deseada de 0.01, 20 patrones con 100 épocas para el entrenamiento y 6 patrones con 3 épocas para la generalización. De la gráfica del inciso (a) de la Figura 4.12 donde se encuentran la señal deseada con la estimada para el entrenamiento del mezclador 2 se observa que ambas señales son muy aproximadas teniéndose por consiguiente un buen entrenamiento. De la gráfica del inciso (b) de la Figura 4.12 donde se encuentran la señal deseada con la estimada para la generalización del mezclador 2 se observa que ambas señales son muy aproximadas teniéndose por consiguiente una buena generalización. De la gráfica del inciso (c) de la Figura 4.12 donde se encuentra el error promedio cuadrático para el entrenamiento del mezclador 2 se observa que el valor del error promedio cuadrático de entrenamiento es de 0,1463. De la gráfica del inciso (d) de la Figura 4.12 donde se encuentra el error promedio cuadrático para la generalización del mezclador 2 se observa que el valor del error promedio cuadrático de generalización es de 0,0337.

En la Figura 4.13 se simula una red neuronal con parámetro de aprendizaje variable, 2 entradas, 1 salida, 5 neuronas en la capa oculta, una ganancia para la función tangente hiperbólica de 0.001, una ganancia de la salida deseada de 0.01, 20 patrones con 100 épocas para el entrenamiento y 6 patrones con 3 épocas para la generalización. De la gráfica del inciso (a) de la Figura 4.13 donde se encuentran la señal deseada con la estimada para el entrenamiento del mezclador 3 se observa que ambas señales son muy aproximadas teniéndose por consiguiente un buen entrenamiento. De la gráfica del inciso (b) de la Figura 4.13 donde se encuentran la señal deseada con la estimada para la generalización del mezclador 3 se observa que ambas señales son muy aproximadas teniéndose por consiguiente una buena generalización. De la gráfica del inciso (c) de la Figura 4.13 donde se encuentra el error promedio cuadrático para el entrenamiento del mezclador 3 se observa que el valor del error promedio cuadrático de entrenamiento es de  $6,7055e - 006$ . De la gráfica del inciso (d) de la Figura 4.13 donde se encuentra el error promedio cuadrático para la generalización del mezclador 3 se observa que el valor del error promedio cuadrático de generalización es de  $1,9722e - 010$ .

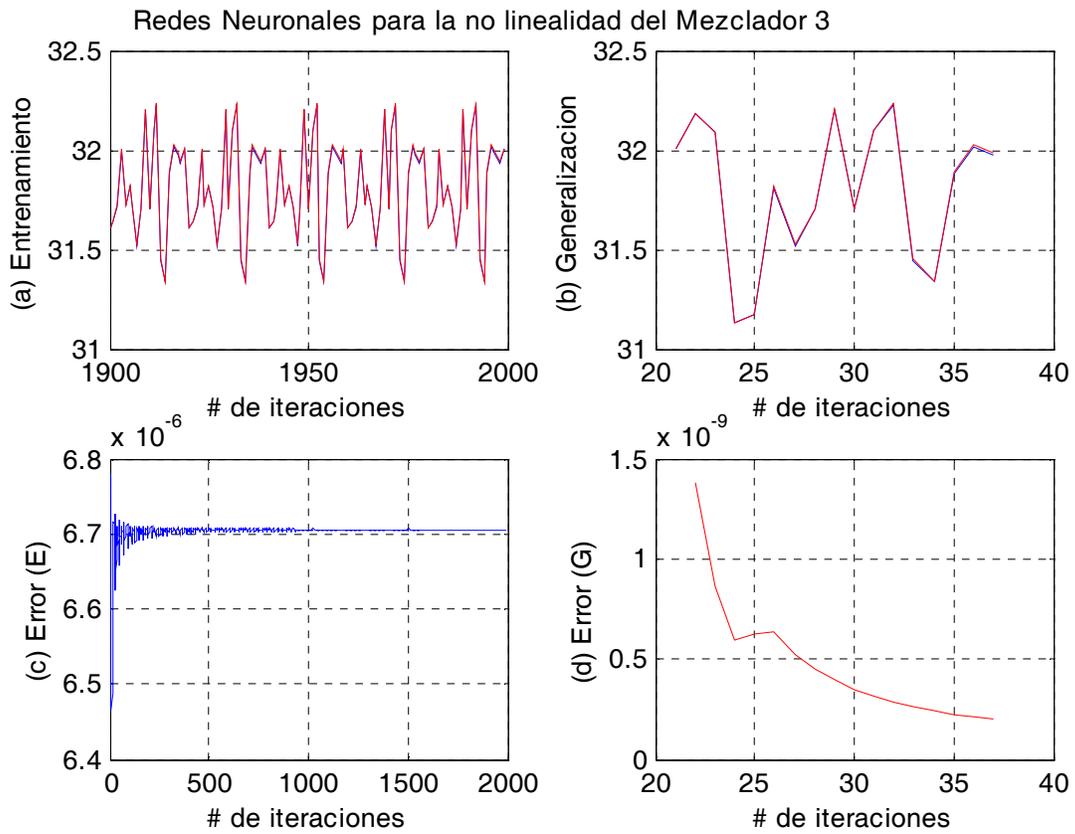


Figura 4.13: Red neuronal para la no linealidad del mezclador 3

## 4.6. Modelación del modelo agregado

Para el modelo agregado se consideran los 3 mezcladores en un modelo (ver Figuras 3.5 y 3.7) teniéndose un programa para todos los mezcladores. Se obtienen 26 datos de una base de datos reales los cuales se emplean para realizar el programa. Se considera 1 programa para identificación de parámetros vía least square, se tienen 2 programas para identificación de parámetros vía redes neuronales debido a que hay dos modalidades.

### 4.6.1. Identificación de parámetros vía least square

Para este programa se emplea la formula descrita por la ecuación (4.3) para el Least square monovariante la cual se vuelve a citar a continuación:

$$\begin{aligned}
 \alpha &= (X^T X)^{-1} Y^T X \\
 X &= [x(1), x(2), \dots, x(26)] \\
 Y &= [y(1), y(2), \dots, y(26)] \\
 x(k) &= u(1, k) * u(2, k) * u(3, k) * u(4, k), \quad k = 1, 2, \dots, 26 \\
 y1(k) &= \alpha * x(k), \quad k = 1, 2, \dots, 26
 \end{aligned}
 \tag{4.28}$$

donde  $u(1, k)$  es la fracción de flujo de la entrada 1 para el modelo,  $u(2, k)$  es la fracción de flujo de la entrada 2 para el modelo,  $u(3, k)$  es la fracción de flujo de la entrada 3 para el modelo,  $u(4, k)$  es la fracción de flujo de la entrada 4 para el modelo,  $\alpha$  es el parámetro a identificar vía least square,  $y(k)$  es la salida proporcionada por los datos reales y  $y1(k)$  es la salida identificada vía least square.

Posteriormente se grafican los resultados, que son tres gráficas en una Figura (ver Figura 4.14):

- La primer gráfica compara la señal deseada de los datos proporcionados por el programa de Excel contra señal de salida que se tiene con la identificación vía Least square.
- La segunda gráfica los puntos de la entrada contra la salida para el programa de Excel, también gráfica en una línea recta la pendiente que se obtiene vía Least square la

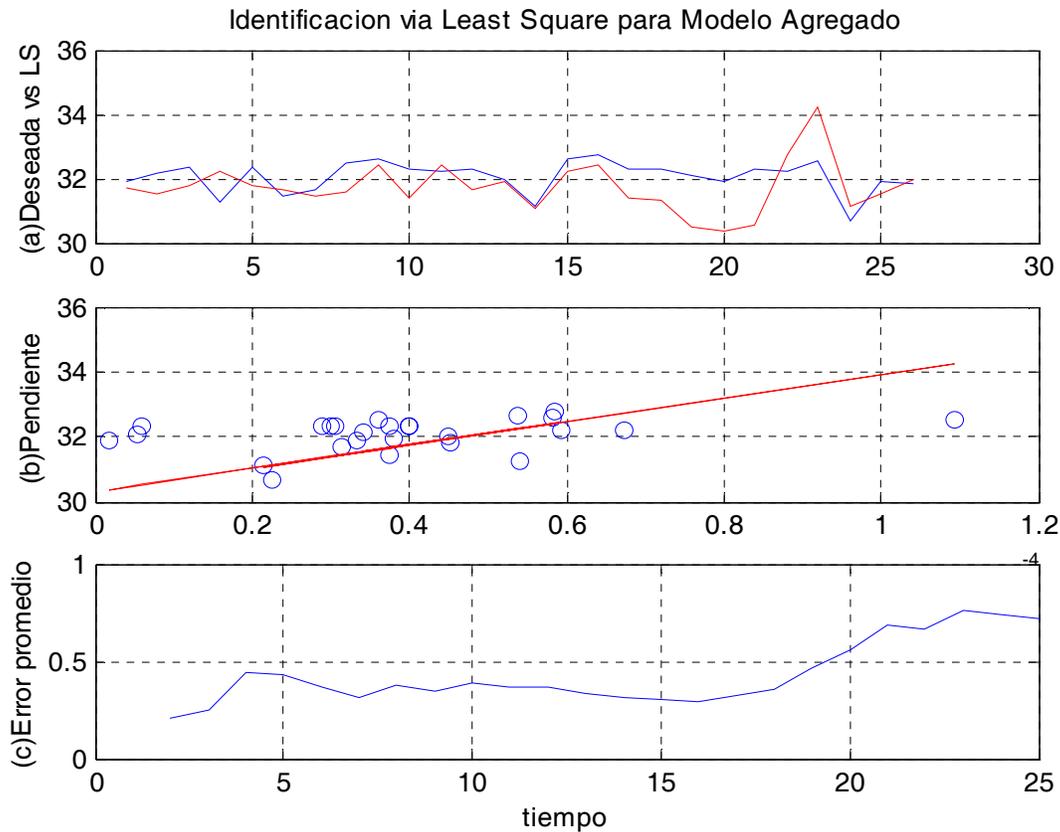


Figura 4.14: Least square para el modelo agregado

cual es el valor del parámetro  $\alpha$ , donde  $\alpha$  es el parámetro de iteración del modelo de iteración.

- La tercer gráfica proporciona el error promedio cuadrático de la señal deseada y la señal de salida para el algoritmo de Least square.

A continuación se analiza la gráfica del programa de Matlab de Least square para el Modelo agregado:

De la gráfica del inciso (a) de la Figura 4.14 donde se encuentran la señal deseada con

la estimada del modelo agregado se observa que la señal estimada se aproxima a la señal deseada. De la gráfica del inciso (b) de la Figura 4.14 donde se encuentra la pendiente para el modelo agregado proporciona una pendiente con valor aproximado a 4. De la gráfica del inciso (c) de la Figura 4.14 donde se encuentra el error promedio cuadrático del modelo agregado se observa que el valor del error promedio cuadrático es de 0,7155.

#### 4.6.2. Identificación vía redes neuronales para el modelo completo

Para este programa se emplea el algoritmo denominado de aprendizaje con parámetros de aprendizaje variables descrito al principio de éste capítulo, este algoritmo es parecido al algoritmo backpropagation, de hecho en el camino hacia adelante son idénticos, el camino hacia adelante es obtener la salida de cada neurona a partir de sus entradas, para esto se emplea la formula (2.1) la cual se vuelve a citar a continuación para la  $i$ -ésima neurona:

$$u_i = \sum_{j=1}^n \omega_{ij} x_j$$

$$y_i = \phi(u_i - \rho_i)$$

donde:

$x_j$  j-ésimo componente de la entrada.

$\omega_{ij}$  peso de la conexión entre la  $j$ -ésima componente de la entrada y la  $i$ -ésima neurona.

$u_i$  salida del sumador.

$\rho_i$  umbral.

$\phi(\cdot)$  función de activación no lineal.

$y_i$  salida de la  $i$ -ésima neurona.

En este caso no se emplea el umbral quedando la salida de cada neurona como sigue:

$$y_i = \phi(u_i)$$

Para el modelo agregado se tienen cuatro entradas que son  $u(1, k)$ ,  $u(2, k)$ ,  $u(3, k)$  y  $u(4, k)$ .

Para el caso del camino hacia atrás se tiene como se propaga el error de la salida a través de cada neurona en la red neuronal, la diferencia entre el algoritmo de aprendizaje

con parámetros de aprendizaje variables y el backpropagation estriba en que en el caso del algoritmo de aprendizaje con parámetros de aprendizaje variables emplea el parámetro de aprendizaje que varía con cada iteración y además, en éste se puede comprobar la estabilidad.

La actualización de los pesos sinópticos se describe mediante la ecuación (4.12) la cual se vuelve a citar a continuación:

$$\begin{aligned} W_{k+1} &= W_k - \eta_k e(k) \phi V^o T X^T(k) \\ V_{k+1} &= V_k - \eta_k e(k) \phi^T \end{aligned}$$

donde la salida escalar de la red neuronal es  $\hat{y}(k) \in R^{n \times 1}$ , el vector de entrada a la red es  $X(k) \in R^{n \times 1}$  definida en ecuación (4.8) donde se emplean la cuatro entradas que son  $u(1, k)$ ,  $u(2, k)$ ,  $u(3, k)$  y  $u(4, k)$ , los pesos sinópticos en la capa de salida son  $V_k \in R^{1 \times m}$ , los pesos en la capa oculta son  $W_k \in R^{m \times n}$ ,  $\phi$  es una función vectorial  $m$ -dimensional, la representación típica de la función  $\phi_i(\cdot)$  es una función sigmoideal, pero para propósitos del programa se selecciona la función tangente hiperbólica ya que esta además de ser derivable (se emplea la derivada en el algoritmo) también es asimétrica por lo que considera valores menores y mayores a cero proporcionando mejores resultados en la simulación. El error de identificación en la capa de salida se define como:

$$e(k) = \hat{y}(k) - y(k)$$

donde  $\eta$  es el parámetro de aprendizaje inicial que tiene el valor de 1.  $V^o$  es un vector seleccionado por el usuario para calibrar la red que para propósitos del programa en Matlab se pone como  $V^o = \text{rand}$  lo cual quiere decir que puede tomar cualquier valor entre 0 y 1.

Para el entrenamiento y examinación se emplean las siguientes fórmulas:

$$\begin{aligned} U &= [u(1, k), u(2, k), u(3, k), u(4, k)], & k &= 1, 2, \dots, 26 \\ DE &= [DE(1, k), DE(2, k), DE(3, k), DE(4, k)], & k &= 1, 2, \dots, 26 \\ y(k) &= (DE * U^T) + NLA(1, k), & k &= 1, 2, \dots, 26 \\ yn(k) &= (DE * U^T) + yn(k), & k &= 1, 2, \dots, 26 \end{aligned} \tag{4.29}$$

donde  $DE(1, k)$  es la densidad de la primer entrada para el modelo,  $DE(2, k)$  es la densidad de la segunda entrada para el modelo,  $DE(3, k)$  es la densidad de la tercer entrada para el

modelo,  $DE(4, k)$  es la densidad de la cuarta entrada para el modelo,  $NLA(1, k)$  es la no linealidad del modelo agregado,  $u(1, k)$  es la fracción de flujo de la entrada 1 para el modelo,  $u(2, k)$  es la fracción de flujo de la entrada 2 para el modelo,  $u(3, k)$  es la fracción de flujo de la entrada 3 para el modelo,  $u(4, k)$  es la fracción de flujo de la entrada 4 para el modelo,  $y(k)$  es la salida deseada para el modelo y  $yn(k)$  es la salida de la red neuronal del entrenamiento para el modelo.

Posteriormente se grafican los resultados, que son cuatro gráficas en una Figura (ver Figura 4.15):

- La primer gráfica compara la señal deseada de los datos proporcionados por el programa de Excel contra señal de salida que se tiene con la identificación vía Redes Neuronales para el caso del entrenamiento.
- La segunda gráfica compara la señal deseada de los datos proporcionados por el programa de Excel contra señal de salida que se tiene con la identificación vía Redes Neuronales para el caso de la generalización.
- La tercer gráfica proporciona el error promedio cuadrático de la señal deseada y la señal de salida para el entrenamiento del algoritmo de Redes Neuronales.
- La cuarta gráfica proporciona el error promedio cuadrático de la señal deseada y la señal de salida para la generalización del algoritmo de Redes Neuronales.

A continuación se analiza la gráfica del programa de Matlab de Redes neuronales para el modelo completo del modelo agregado:

En la Figura 4.15 se simula una red neuronal con parámetro de aprendizaje variable, 2 entradas, 1 salida, 5 neuronas en la capa oculta, una ganancia para la función tangente hiperbólica de 4, 20 patrones con 100 épocas para el entrenamiento y 6 patrones con 50 épocas para la generalización. De la gráfica del inciso (a) de la Figura 4.15 donde se encuentran la señal deseada con la estimada para el entrenamiento del modelo agregado se observa que ambas señales son muy aproximadas teniéndose por consiguiente un buen entrenamiento.

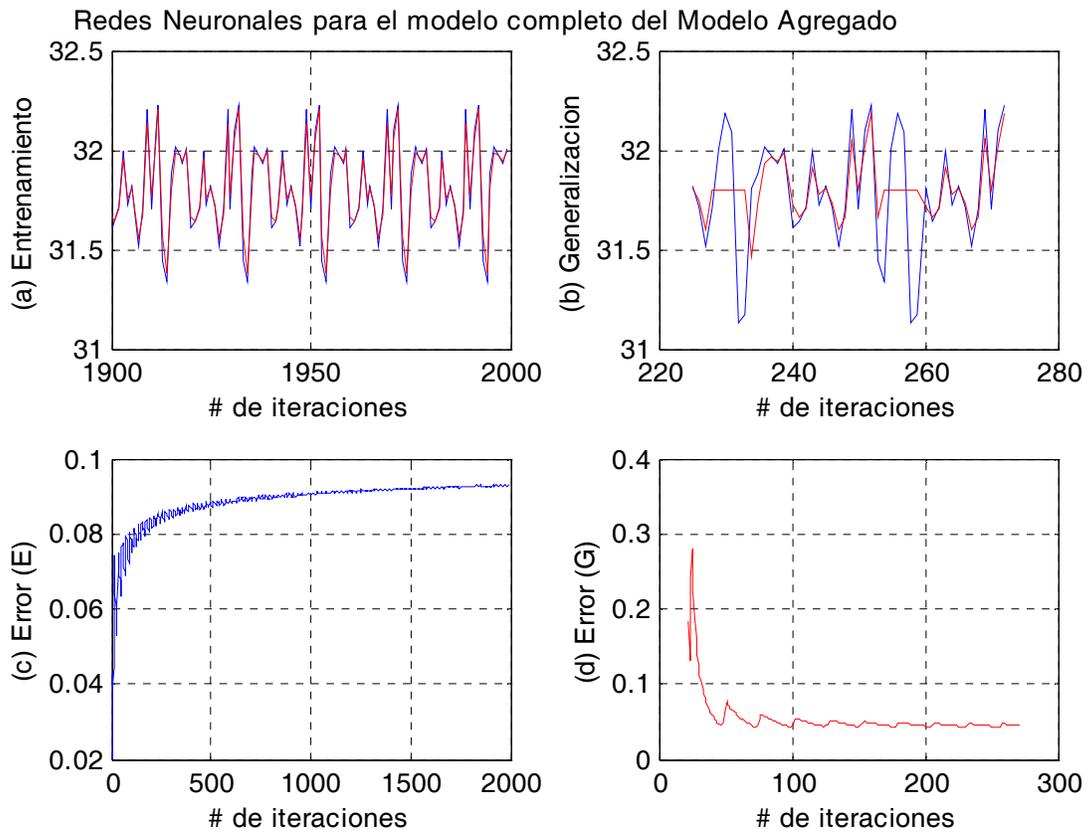


Figura 4.15: Red neuronal para el modelo completo del modelo agregado

De la gráfica del inciso (b) de la Figura 4.15 donde se encuentran la señal deseada con la estimada para la generalización del modelo agregado se observa que ambas señales son aproximadas teniéndose por consiguiente una buena generalización. De la gráfica del inciso (c) de la Figura 4.15 donde se encuentra el error promedio cuadrático para el entrenamiento del modelo agregado se observa que el valor del error promedio cuadrático de entrenamiento es de 0,0929. De la gráfica del inciso (d) de la Figura 4.15 donde se encuentra el error promedio cuadrático para la generalización del modelo agregado se observa que el valor del error promedio cuadrático de generalización es de 0,0447.

### 4.6.3. Identificación vía redes neuronales para la parte no lineal

Este programa es idéntico al programa de redes neuronales para el modelo completo con la siguiente diferencia:

Para el entrenamiento se emplean las siguientes fórmulas:

$$\begin{aligned}
 U &= [u(1, k), u(2, k)], & k &= 1, 2, \dots, 26 \\
 DE &= [DE(1, k), DE(2, k)], & k &= 1, 2, \dots, 26 \\
 y(k) &= (NLD1(1, k) * P), & k &= 1, 2, \dots, 26 \\
 yn(k) &= yn(k), & k &= 1, 2, \dots, 26
 \end{aligned} \tag{4.30}$$

donde  $DE(1, k)$  es la densidad de la primer entrada para el modelo,  $DE(2, k)$  es la densidad de la segunda entrada para el modelo,  $DE(3, k)$  es la densidad de la tercer entrada para el modelo,  $DE(4, k)$  es la densidad de la cuarta entrada para el modelo,  $NLA(1, k)$  es la no linealidad del modelo agregado,  $u(1, k)$  es la fracción de flujo de la entrada 1 para el modelo,  $u(2, k)$  es la fracción de flujo de la entrada 2 para el modelo,  $u(3, k)$  es la fracción de flujo de la entrada 3 para el modelo,  $u(4, k)$  es la fracción de flujo de la entrada 4 para el modelo,  $y(k)$  es la salida deseada del entrenamiento para el modelo,  $yn(k)$  es la salida de la red neuronal del entrenamiento para el modelo,  $P$  es una ganancia para la salida deseada, este parámetro se emplea debido a que  $NLA(1, k)$  es muy pequeño y se necesita amplificar para que la función tangente hiperbólica pueda operar adecuadamente.

Para la examinación se emplean las siguientes fórmulas:

$$\begin{aligned} ya(k) &= (DE * U^T) + NLD1(1, k), & k = 1, 2, \dots, 26 \\ yna(k) &= (DE * U^T) + (yn(k)/P), & k = 1, 2, \dots, 26 \end{aligned} \quad (4.31)$$

Note que en esta etapa se le divide por el parámetro  $P$  debido a que este parámetro se multiplica a la salida deseada para tener un mejor aprendizaje,  $ya(k)$  es la salida deseada de la examinación para el modelo,  $yna(k)$  es la salida de la red neuronal de la examinación para el modelo.

Posteriormente se grafican los resultados, que son cuatro gráficas en una Figura (ver Figura 4.16):

- La primer gráfica compara la señal deseada de los datos proporcionados por el programa de Excel contra señal de salida que se tiene con la identificación vía Redes Neuronales para el caso del entrenamiento.
- La segunda gráfica compara la señal deseada de los datos proporcionados por el programa de Excel contra señal de salida que se tiene con la identificación vía Redes Neuronales para el caso de la generalización.
- La tercer gráfica proporciona el error promedio cuadrático de la señal deseada y la señal de salida para el entrenamiento del algoritmo de Redes Neuronales.
- La cuarta gráfica proporciona el error promedio cuadrático de la señal deseada y la señal de salida para la generalización del algoritmo de Redes Neuronales.

A continuación se analiza la gráfica del programa de Matlab de Redes neuronales para la no linealidad del modelo agregado:

En la Figura 4.16 se simula una red neuronal con parámetro de aprendizaje variable, 2 entradas, 1 salida, 5 neuronas en la capa oculta, una ganancia para la función tangente hiperbólica de 0.001, una ganancia de la salida deseada de 0.01, 20 patrones con 100 épocas para el entrenamiento y 6 patrones con 3 épocas para la generalización. De la gráfica del inciso (a) de la Figura 4.16 donde se encuentran la señal deseada con la estimada para

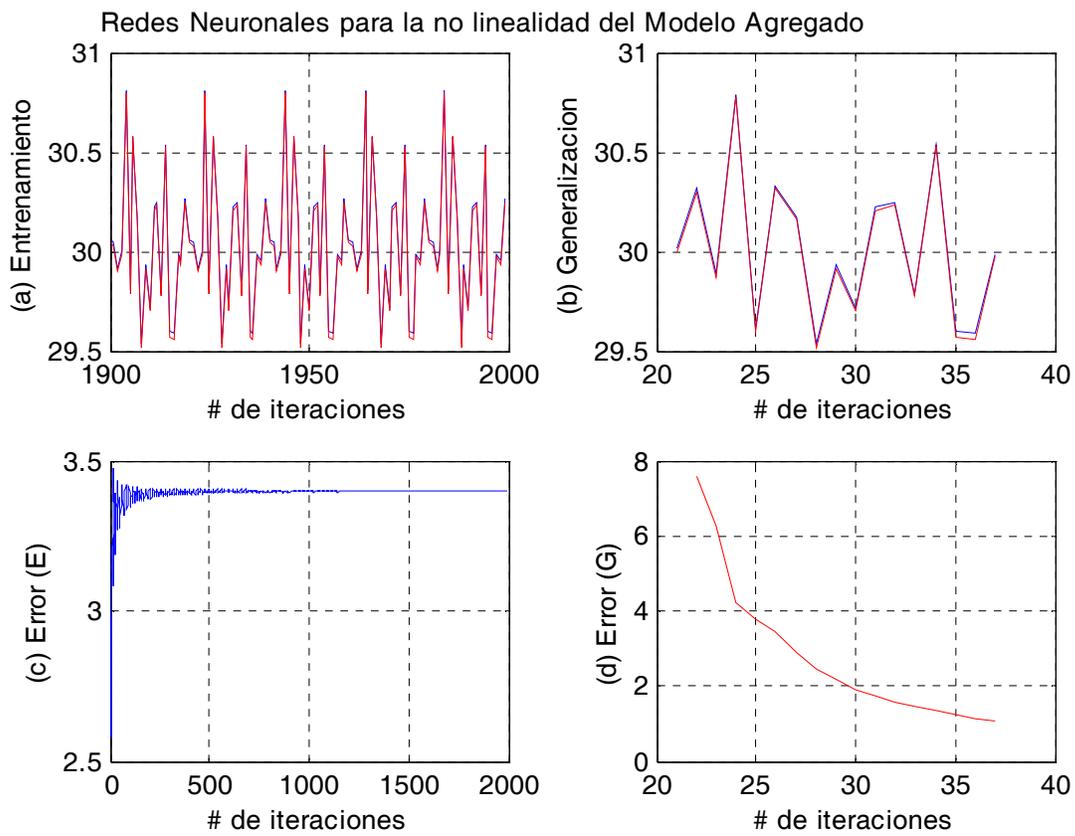


Figura 4.16: Red neuronal para la no linealidad del modelo agregado

el entrenamiento del modelo agregado se observa que ambas señales son muy aproximadas teniéndose por consiguiente un buen entrenamiento. De la gráfica del inciso (b) de la Figura 4.16 donde se encuentran la señal deseada con la estimada para la generalización del modelo agregado se observa que ambas señales son muy aproximadas teniéndose por consiguiente una buena generalización. De la gráfica del inciso (c) de la Figura 4.16 donde se encuentra el error promedio cuadrático para el entrenamiento del modelo agregado se observa que el valor del error promedio cuadrático de entrenamiento es de 3,4020. De la gráfica del inciso (d) de la Figura 4.16 donde se encuentra el error promedio cuadrático para la generalización del modelo agregado se observa que el valor del error promedio cuadrático de generalización es de 1,0822.

## 4.7. Conclusión

En el desarrollo de esta sección se realiza una comparación entre la identificación vía least square contra la identificación vía redes neuronales para el modelo completo y contra la identificación vía redes neuronales para la no linealidad teniéndose mejores resultados para la identificación vía redes neuronales para la no linealidad dado que en éste caso se tiene una generalización excelente ,i.e., las señales identificada y la deseada son muy aproximadas, y además, se ocupa un menor número de épocas para la generalización, comprobándose con esto que es recomendable el empleo de las redes neuronales para tener mejores resultados en la identificación para el proceso de mezclado de crudo. El resultado anterior se corrobora mediante la siguiente tabla.

	LS = Least square			M1 = Mezclador 1
	NNMC = Red neuronal para el modelo completo			M2 = Mezclador 2
	NNNL = Red neuronal para la no linealidad			M3 = Mezclador 3
				MA = M. Agregado
	error promedio	#	#	Generalización
	cuadrático	entradas	salidas	
LSM1	0,2046	2	1	Mala
LSM2	0,0869	2	1	Mala
LSM3	$5,0487e - 031$	2	1	Buena
LSMA	0,7155	4	1	Regular

	# entradas	# salidas	Ganancia la de función tangente hiperbólica	Ganancia de la salida deseada	
NNMCM1	2	1	4	—	
NNMCM2	2	1	4	—	
NNMCM3	2	1	4	—	
NNMCMA	4	1	4	—	
NNNLM1	2	1	0,001	0,01	
NNNLM2	2	1	0,001	0,01	
NNNLM3	2	1	0,001	0,01	
NNNLMA	4	1	0,001	0,01	
	error promedio de entrenamiento		error promedio de generalización	# neuronas en la capa oculta	Generalización
NNMCM1	0,0538		0,0197	5	Buena
NNMCM2	0,0931		0,0243	5	Buena
NNMCM3	0,0867		0,0433	5	Buena
NNMCMA	0,0929		0,0447	5	Buena
NNNLM1	0,2314		0,1053	5	Excelente
NNNLM2	0,1463		0,0337	5	Excelente
NNNLM3	$6,7055e - 006$		$1,9722e - 010$	5	Excelente
NNNLMA	3,4020		1,0822	5	Excelente
	# patrones de entrenamiento	# patrones de generalización	# épocas de entrenamiento	# épocas de generalización	
NNMCM1	20	6	100	50	
NNMCM2	20	6	100	50	
NNMCM3	20	6	100	50	
NNMCMA	20	6	100	50	
NNNLM1	20	6	100	3	
NNNLM2	20	6	100	3	
NNNLM3	20	6	100	3	
NNNLMA	20	6	100	3	



# Capítulo 5

## Optimización de petróleo crudo vía redes neuronales y actualización de datos (bias updating)

### 5.1. Introducción

En este capítulo se presenta como se aplica el controlador optimizante con redes neuronales al proceso de mezclado, para ello en primer lugar se demuestra la robustez del controlador optimizante [12], después se demuestra la robustez del controlador optimizante con redes neuronales [26], posteriormente se realiza una comparación entre el controlador optimizante que opera fuera de línea al cual se le introducen datos reales y el controlador optimizante con redes neuronales que opera fuera de línea, por último se presenta una tabla comparativa que muestra los resultados para cada caso.

Muchos controladores optimizantes de mezclado de petróleo crudo están basados sobre la técnica de programación lineal. En general, la regla de mezclado es no lineal, un método alternativo es adicionar un término de corrección (incertidumbre) a la regla de mezclado lineal. Para llevar a cabo la operación de mezclado con las incertidumbres, se ha propuesto la optimización en tiempo real. El esquema de actualización de datos es una herramienta

para resolver el problema de optimización en tiempo real [3][12]. La desventaja principal del esquema de actualización es que requiere de la medición en línea de las propiedades de mezclado. Algunas veces también se necesita la optimización fuera de línea, por ejemplo, si se espera encontrar la fracción de velocidad de flujo a la entrada óptimo sobre datos históricos, este tipo de control es útil para decisión, supervisión y control [26].

Un modelo real para el mezclado de petróleo crudo basado sobre los datos de operación es muy importante para los ingenieros. Resultados recientes muestran que la técnica de redes neuronales es muy efectiva para modelar una gran cantidad de sistemas no lineales complejos donde no se tienen información completa del modelo. Es muy importante asegurar la estabilidad del modelo neuro teóricamente antes de usarlo en algunas aplicaciones reales.

## 5.2. Optimización del mezclado de petróleo crudo con redes neuronales y un esquema de actualización (bias updating)

### 5.2.1. Optimización del mezclado petróleo crudo con un esquema de actualización

El proceso de mezclado de aceite crudo esta sujeto a un número de restricciones. Los objetivos del mezclado de petróleo crudo son maximizar el valor de todos los productos mientras se satisfacen sus restricciones, o minimizar el costo del mezclado [26]. El mezclado de aceite crudo se puede considerar como una mezcla de  $n$  componentes con:

$$\begin{aligned}
 q &= [q_1, \dots, q_n]^T = \text{velocidad de flujo en la entrada} \\
 p &= [p_1, \dots, p_n]^T = \text{propiedad en la entrada} \\
 q_f &= \text{velocidad de flujo en la salida} \\
 p_f &= \text{propiedad en la salida} \\
 u &= \left[ \frac{q_1}{q_f}, \dots, \frac{q_n}{q_f} \right]^T = \text{fracción de flujo en la entrada}
 \end{aligned}$$

Considerando los efectos dinámicos insignificantes, la regla de mezclado es una función no lineal:

$$p_f = f(u, p) \tag{5.1}$$

El problema control de mezclado es la optimización del costo sobre las variables manipulables o el vector de flujo en la entrada de control  $u$ , la función objetivo esta en forma de programación lineal:

$$\text{mín } c^T u \tag{5.2}$$

donde  $c$  es el vector de parámetros de costo del alimentación. Las restricciones son como sigue:

1. Disponibilidad mínima y máxima:

$$u_{i,\text{mín}} \leq u_i \leq u_{i,\text{máx}} \tag{5.3}$$

2. Balance de masa:

$$\sum_{i=1}^n u_i = u_f = 1 \tag{5.4}$$

3. Requerimientos de las propiedades cualitativas:

$$p_{f,\text{mín}} \leq p_f \leq p_{f,\text{máx}} \tag{5.5}$$

Una forma estándar la cual se puede resolver con el método simplex es:

$$\begin{aligned} &\text{mín } c^T u \\ &\text{sujeto a:} \\ &Au \leq b \\ &A_{eq}u = b_{eq} \\ &u_{i,\text{mín}} \leq u_i \leq u_{i,\text{máx}} \end{aligned} \tag{5.6}$$

Se puede ver que ecuaciones (5.3) y (5.4) están incluidas en la forma estándar de la ecuación (5.6), pero la ecuación (5.5) no se puede aplicar en la ecuación (5.6) directamente, se emplea

el esquema de actualización para resolver este problema. Una suposición básica del esquema de actualización es que la regla de mezclado de la ecuación (5.1) se puede expresar como una parte lineal conocida y una parte no lineal desconocida:

$$p_f = \frac{q^T p}{n} + \Delta \tag{5.7}$$

$$\sum_{i=1} q_i$$

donde  $p_f$  es medible,  $p$  es un vector constante, se decidirá el valor de  $u$  mediante la optimización, en aplicaciones,  $\Delta$  se puede estimar por el valor previo de  $u = \frac{q}{q_f} = q$ . Desde una condición inicial para  $\Delta$ , por ejemplo  $\Delta(0) = 0$ ,  $\Delta$  se puede estimar después de calcular  $u = q$  mediante de la optimización.

$$\Delta(k + 1) = p_f(k) - \frac{q^T(k)p}{n} = p_f(k) - p^T(k)u(k) \tag{5.8}$$

$$\sum_{i=1} q_i$$

De las ecuaciones (5.5) y (5.7) se tiene lo siguiente:

$$p_{f,\text{mín}} \leq \frac{q^T p}{q_f} + \Delta(k) \leq p_{f,\text{máx}} \tag{5.9}$$

$$p_{f,\text{mín}} - \Delta(k) \leq \frac{p}{q_f} q \leq p_{f,\text{máx}} - \Delta(k)$$

$$-(p_{f,\text{mín}} - \Delta(k)) \geq -\frac{p}{q_f} q$$

$$-\frac{p}{q_f} q \leq -p_{f,\text{mín}} + \Delta(k) \quad \frac{p}{q_f} q \leq p_{f,\text{máx}} - \Delta(k)$$

$$\begin{bmatrix} \frac{p}{q_f} \\ -\frac{p}{q_f} \end{bmatrix} q(k) \leq \begin{bmatrix} p_{f,\text{máx}} - \Delta(k) \\ -p_{f,\text{mín}} + \Delta(k) \end{bmatrix}$$

Ahora, las ecuaciones (5.2), (5.3), (5.4), (5.7) y (5.8) se transforman en la forma estándar de la programación lineal:

$$\begin{aligned} & \text{mín } c^T u(k) \\ & \text{sujeto a:} \\ & Au(k) \leq b \\ & A_{eq} u(k) = b_{eq} \\ & u_{\text{mín}} \leq u(k) \leq u_{\text{máx}} \end{aligned} \tag{5.10}$$

donde:

$$\begin{aligned}
 A &= \begin{bmatrix} \frac{p}{q_f} \\ -\frac{p}{q_f} \end{bmatrix} = \begin{bmatrix} p \\ -p \end{bmatrix} \\
 b &= \begin{bmatrix} p_{f,\text{máx}} - \Delta(k) \\ -p_{f,\text{mín}} + \Delta(k) \end{bmatrix} \\
 A_{eq} &= [1, \dots, 1] \\
 b_{eq} &= u_f = 1 \\
 \Delta(k) &= p_f(k-1) - p^T(k-1)u(k-1)
 \end{aligned}$$

En muchos problemas de ingeniería, se requiere que la convergencia rápida hacia la solución óptima. De cualquier manera, los algoritmos tradicionales para computadora, tales como los Toolbox de Optimización de Matlab, no son muy eficientes, debido a que el tiempo necesario para el cálculo depende de la dimensión y de la estructura del problema, una solución a éste problema es aplicar redes neuronales las cuales hacen que el problema de optimización converja más rápido.

### 5.2.2. Optimización fuera de línea del mezclado de petróleo crudo con redes neuronales

El alcance propuesto en la sección anterior, solamente se puede aplicar en línea, por que el esquema de actualización requiere de la propiedad de salida  $p_f$  la cual se obtiene mediante un analizador en línea. Algunas veces, se necesita la optimización fuera de línea, por ejemplo, cuando se desea encontrar la velocidad de flujo a la entrada óptima sobre datos históricos, este control es útil para la decisión, supervisión y control. En esta sección se da la respuesta a como diseñar un controlador optimizante con datos históricos [26].

El diagrama esquemático de la optimización fuera de línea del mezclado de petróleo crudo se muestra en la Figura 5.1.

Se puede evitar la suposición de analizar en tiempo real mediante el empleo de un esquema de actualización y de la red neuronal para estimar el comportamiento del mezclador con datos históricos, después del aprendizaje de la red neuronal se tiene el valor de salida de la red

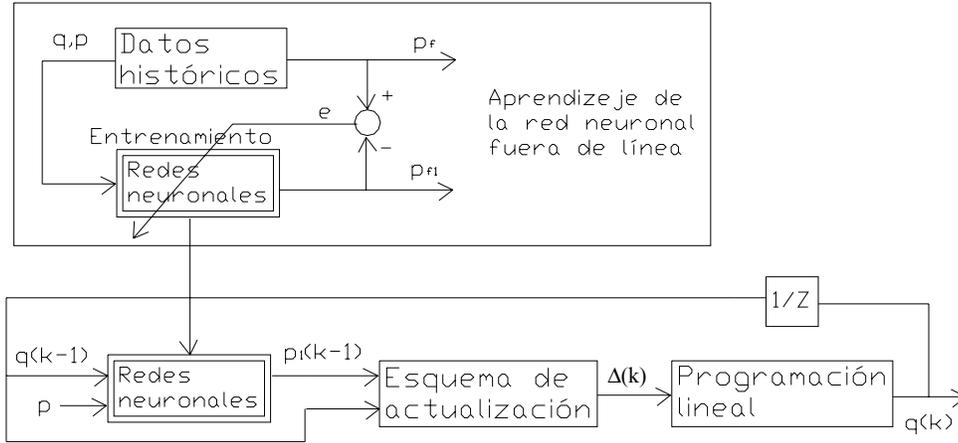


Figura 5.1: Diagrama de la optimización fuera de línea del mezclado de crudo con nn neuronal como las propiedades cualitativas del proceso de mezclado:

$$\hat{p}_f = V_k \phi [W_k X(k)] \quad (5.11)$$

Del requerimiento de las propiedades cualitativas de la ecuación (5.5) se tiene:

$$\begin{aligned} p_{f,\text{mín}} &\leq p_f \leq p_{f,\text{máx}} \\ p_{f,\text{mín}} &\leq V_k \phi [W_k X(k)] \leq p_{f,\text{máx}} \\ p_{f,\text{mín}} &\leq Aq(k) + V_k \phi [W_k X(k)] - Aq(k) \leq p_{f,\text{máx}} \end{aligned} \quad (5.12)$$

donde  $A$  es una matriz constante,  $\Delta$  se puede estimar con la ecuación (5.8) después de calcular  $u = q$  mediante de la optimización:

$$p_{f,\text{mín}} \leq \frac{q^T p}{q_f} + \Delta(k) \leq p_{f,\text{máx}}$$

Pero de la ecuación (5.7) se tiene:

$$\Delta(k) = \hat{p}_f - \frac{q^T p}{q_f}$$

$$\Delta(k) = \widehat{p}_f - Aq(k) \quad (5.13)$$

Sustituyendo ecuación (5.13) en ecuación (5.12) se tiene:

$$p_{f,\text{mín}} \leq Aq(k) + \Delta(k) \leq p_{f,\text{máx}} \quad (5.14)$$

Por tanto, como las ecuaciones (5.9) y (5.14) son iguales, se tienen las mismas restricciones que las de la sección anterior, la única diferencia radica en que para cuando se emplean redes neuronales se emplea la ecuación (5.11) evitando el empleo de un modelo matemático.

Ahora, las ecuaciones (5.2), (5.3), (5.4), (5.7) y (5.8) se transforman en la forma estándar de la programación lineal:

$$\begin{aligned} & \text{mín } c^T u(k) \\ & \text{sujeto a:} \\ & Au(k) \leq b \\ & A_{eq}u(k) = b_{eq} \\ & u_{\text{mín}} \leq u(k) \leq u_{\text{máx}} \end{aligned} \quad (5.15)$$

donde:

$$\begin{aligned} A &= \begin{bmatrix} \frac{p}{q_f} \\ -\frac{p}{q_f} \end{bmatrix} = \begin{bmatrix} p \\ -p \end{bmatrix} \\ b &= \begin{bmatrix} p_{f,\text{máx}} - \Delta(k) \\ -p_{f,\text{mín}} + \Delta(k) \end{bmatrix} \\ A_{eq} &= [1, \dots, 1] \\ b_{eq} &= u_f = 1 \\ \Delta(k) &= p_f(k-1) - p^T(k-1)u(k-1) \end{aligned}$$

## 5.3. Análisis de robustez del controlador con redes neuronales para mezclado

### 5.3.1. Análisis de robustez del controlador

Considere el siguiente problema de control [12]:

$$\begin{aligned}
 & \text{mín } c^T u \\
 & \text{sujeto a:} \\
 & \sum_{i=1}^n u_i = 1 \\
 & p_{\text{mín}} \leq p^T u + N(u) \leq p_{\text{máx}} \\
 & u_{\text{mín}} \leq u \leq u_{\text{máx}}
 \end{aligned} \tag{5.16}$$

donde:

$$\begin{aligned}
 N(u) &= \text{no linealidad} \\
 u &= [u_1, \dots, u_n]^T = \text{fracción de flujo en la entrada} \\
 c &= [c_1, \dots, c_n]^T = \text{función de costo} \\
 p &= [p_1, \dots, p_n]^T = \text{propiedad del flujo en la entrada} \\
 u_{\text{máx}} &= [u_{\text{máx}1}, \dots, u_{\text{máx}n}]^T = \text{fracción de flujo máxima en la entrada} \\
 u_{\text{mín}} &= [u_{\text{mín}1}, \dots, u_{\text{mín}n}]^T = \text{fracción de flujo mínima en la entrada} \\
 p_{\text{máx}} &= \text{propiedad del flujo máxima en la salida} \\
 p_{\text{mín}} &= \text{propiedad del flujo mínima en la salida}
 \end{aligned}$$

Realizando el siguiente desarrollo matemático se tiene:

$$\begin{aligned}
 u_1 + \dots + u_n = 1 &\rightarrow u_1 + \dots + u_n - 1 = 0 \\
 u \leq u_{\text{máx}} &\rightarrow u - u_{\text{máx}} \leq 0 \\
 u \geq u_{\text{mín}} &\rightarrow -u + u_{\text{mín}} \leq 0 \\
 p^T u + N(u) \leq p_{\text{máx}} &\rightarrow p^T u + N(u) - p_{\text{máx}} \leq 0 \\
 p^T u + N(u) \geq p_{\text{mín}} &\rightarrow -p^T u - N(u) + p_{\text{mín}} \leq 0
 \end{aligned}$$

$$\begin{bmatrix} I_{n \times n} \\ -I_{n \times n} \\ p^T \\ -p^T \end{bmatrix} u + \begin{bmatrix} -u_{\text{máx}} \\ u_{\text{mín}} \\ -p_{\text{máx}} \\ p_{\text{mín}} \end{bmatrix} + \begin{bmatrix} 0_n \\ 0_n \\ N(u) \\ -N(u) \end{bmatrix} \leq 0$$

Por tanto, se puede reescribir el problema de control de la siguiente manera:

$$\begin{aligned} & \text{mín } c^T u \\ & \text{sujeto a:} \\ & D_1^T u + b_1 = 0 \\ & D_2 u + b_2 + \mu(u) \leq 0_{2n+2} \end{aligned} \tag{5.17}$$

donde:

$$\begin{aligned} D_1 &= [1, \dots, 1]^T \in R^n \\ D_2 &= \begin{bmatrix} I_{n \times n} \\ -I_{n \times n} \\ p^T \\ -p^T \end{bmatrix} \in R^{(2n+2) \times n} \\ b_1 &= -1 \in R \\ b_2 &= \begin{bmatrix} -u_{\text{máx}} \\ u_{\text{mín}} \\ -p_{\text{máx}} \\ p_{\text{mín}} \end{bmatrix} \in R^{2n+2} \\ \mu(u) &= \begin{bmatrix} 0_n \\ 0_n \\ N(u) \\ -N(u) \end{bmatrix} \in R^{2n+2} \end{aligned}$$

La ecuación (5.17) se puede escribir en forma de Lagrange [20]. Escribiendo la función de Lagrange como:

$$L(u, \lambda, \mu) = f(u) + \lambda g_1(u) + \mu^T g_2(u) \tag{5.18}$$

donde:

$$\begin{aligned} f(u) &= \text{función objetivo} \\ g_1(u) &= \text{restricción de igualdad} \\ g_2(u) &= \text{restricción de desigualdad} \end{aligned}$$

Sustituyendo los valores de la ecuación (5.17) en la ecuación (5.18) se tiene lo siguiente:

$$L(u, \lambda, \mu) = c^T u + \lambda [D_1^T u + b_1] + \mu^T [D_2 u + b_2 + \mu(u)] \quad (5.19)$$

donde:

$$\mu = [\mu_1, \dots, \mu_{2n+2}]^T \in R^{2n+2}$$

Nota: Darse cuenta que la función de Lagrange de la ecuación (5.19) es un escalar.

Derivando la función de Lagrange con respecto a cada uno de sus argumentos e igualando a cero se obtienen las condiciones necesarias y suficientes para obtener el mínimo de la función objetivo del problema de optimización de la ecuación (5.17), para la restricción de desigualdad además se necesitan satisfacer las condiciones de Kuhn-Tucker:

$$\begin{aligned} \frac{\partial L}{\partial u} &= 0 \\ \frac{\partial L}{\partial \lambda} &= 0 \\ \frac{\partial L}{\partial \mu^T} &= 0, \quad \mu^T [D_2 u + b_2 + \mu(u)] = 0, \quad \mu^T \geq 0 \end{aligned} \quad (5.20)$$

Sustituyendo la función de Lagrangiano de la ecuación (5.19) en las condiciones de la ecuación (5.20) se tiene:

$$\begin{aligned} c + \lambda D_1 + [D_2 + \nabla \mu(u)]^T \mu &= 0 \\ D_1^T u + b_1 &= 0 \\ D_2 u + b_2 + \mu(u) &= 0, \quad \mu^T \geq 0 \end{aligned} \quad (5.21)$$

donde:

$$\nabla \mu(u) = \left[ \frac{\partial \mu(u)}{\partial u_1}, \dots, \frac{\partial \mu(u)}{\partial u_n} \right] \in R^{(2n+2) \times n}$$

Considerando solamente las condiciones referentes a las restricciones de la ecuación (5.21) se tiene:

$$\begin{bmatrix} D_1^T \\ D_2 \\ 0_{n \times n} \end{bmatrix} u + \begin{bmatrix} 0 \\ \mu(u) \\ \nabla \mu(u)^T \mu \end{bmatrix} = \begin{bmatrix} -b_1 \\ -b_2 \\ -c - \lambda D_1 - D_2^T \mu \end{bmatrix}$$

El sistema algebraico se puede escribir como:

$$Ax + M(x) = y_{sp} \quad (5.22)$$

donde se requiere para el análisis de estabilidad que la matriz  $A$  sea invertible, por tanto se completan los términos de la siguiente manera:

$$\begin{aligned}
 & p = 3n + 3 \\
 A &= \begin{bmatrix} D_1^T & & \\ D_2 & 0_{nx(2n+3)} & \\ 0_{n \times n} & I_{(2n+3)x(2n+3)} & \end{bmatrix} \in R^{p \times p} \\
 M(x) &= \begin{bmatrix} 0 \\ \mu(u) \\ \nabla \mu(u)^T \mu \end{bmatrix} \in R^p \\
 x &= \begin{bmatrix} 0_{2n+1} \\ u \end{bmatrix} \in R^p \\
 y_{sp} &= \begin{bmatrix} -b_1 \\ -b_2 \\ -c - \lambda D_1 - D_2^T \mu \end{bmatrix} \in R^p
 \end{aligned}$$

Se asegura que la matriz  $A$  es invertible debido a que:

$$\det(A) = \det \begin{bmatrix} D_1^T & & \\ D_2 & 0_{nx(2n+3)} & \\ 0_{n \times n} & I_{(2n+3)x(2n+3)} & \end{bmatrix} \neq 0$$

De este modo, solamente para la prueba de estabilidad (y no para encontrar el óptimo), resolver el problema de programación no lineal de la ecuación (5.16) es equivalente a resolver el sistema algebraico de la ecuación (5.22) [12]. Para simplificar el análisis, se consideran las siguientes suposiciones:

**A.1**  $x_{op}$  es la única solución del sistema algebraico de la ecuación (5.22) para todo  $t \geq 0$ ; es decir, existe solamente un valor  $x_{op}$  tal que  $Ax_{op} + M(x_{op}) = y_{sp}$ . Esto significa que  $u_{op}$

es un extremo único del programa de la ecuación (5.16). Si  $u_{op}$  no es una solución única, el resultado de la convergencia descrito a continuación se sostendrá solo localmente (i.e. en una región de  $x_{op}$ ).

**A.2** Si  $\bar{A}$  es un estimado de  $A$ , ambas,  $A$  y  $\bar{A}$  son matrices invertibles, esto significa que se supone una regularidad de  $u_{op}$  relativo al problema de la ecuación (5.16). Se asegura que la matriz  $A$  es invertible debido a que:

$$\det(A) = \begin{bmatrix} D_1^T & & \\ & 0_{nx(2n+3)} & \\ D_2 & & \\ & & I_{(2n+3)x(2n+3)} \\ 0_{n \times n} & & \end{bmatrix} \neq 0$$

De este modo, acorde a las suposiciones A.1 y A.2 la solución del problema de programación no lineal de la ecuación (5.16) es equivalente a resolver el sistema de ecuaciones algebraicas de la ecuación (5.22) donde la no linealidad  $M(x)$  es no conocida.

Desde la estructura de la teoría de control, este problema se puede ver como el diseño de una estrategia de control robusto (contra las incertidumbres de la no linealidad)  $u(t)$  tal que  $y(t) \rightarrow y_{op}$  asintóticamente.

Sea la estrategia de control con esquema de actualización correspondiente a la ecuación (5.22).

Se define el error de modelado asociado al sistema algebraico de la ecuación (5.22) de la siguiente manera:

$$B(x) = \left[ A - \bar{A} \right] x + M(x) \quad (5.23)$$

donde  $\bar{A}$  es una matriz invertible estimada de la matriz  $A$  dado que  $A = A(q_k)$  y  $\bar{A}$  es el valor de  $A$  en un instante. De la ecuación (5.23) se tiene:

$$M(x) = B(x) - Ax + \bar{A}x$$

Sustituyendo en la ecuación (5.22) se tiene:

$$Ax + B(x) - Ax + \bar{A}x = y_{sp}$$

$$B(x) + \bar{A}x = y_{sp} \quad (5.24)$$

La correspondiente iteración para la ecuación (5.24) está dada por:

$$\begin{aligned} \bar{A}x_k &= y_{sp} - B_k \\ x_k &= \bar{A}^{-1} (y_{sp} - B_k) \end{aligned} \quad (5.25)$$

La correspondiente iteración para la ecuación (5.23) esta dada por:

$$B_k = \left[ A - \bar{A} \right] x_k + M(x) = Ax_k + M(x) - \bar{A}x_k$$

$$B_k = y_{sp} - \bar{A}x_k \quad (5.26)$$

Este procedimiento lleva a un optimizador impropio debido a que se requiere  $x_k$  para calcular la señal del error de modelado de la ecuación (5.26) el cual se requiere para la optimización y obtener el valor de  $x_k$ , esto es, se requiere  $x_k$  para calcularse él mismo. Para resolver este problema se emplea una estimación, es decir, en aplicaciones se puede emplear el valor anterior de  $x_k$  para calcular su valor actual (bias update) [3][12], de esta manera, de la ecuación (5.26) se tiene:

$$B_k = y_{k-1} - \bar{A}x_{k-1} \quad (5.27)$$

siendo  $B_k$  el esquema de actualización para el optimizador en donde  $y_k$  es la salida medida

en un instante y  $y_{sp}$  es la salida deseada del sistema algebraico de la ecuación (5.22), esto es:

$$\text{Si } y_{sp} = \begin{bmatrix} -b_1 \\ -b_2 \\ -c - \lambda D_1 - D_2^T \mu \end{bmatrix} = \begin{bmatrix} 1 \\ - \begin{bmatrix} -u_{\text{máx}} \\ u_{\text{mín}} \\ -p_{\text{máx}} \\ p_{\text{mín}} \end{bmatrix} \\ - \begin{bmatrix} c_1 \\ \vdots \\ c_n \end{bmatrix} - \lambda \begin{bmatrix} 1 \\ \vdots \\ 1 \end{bmatrix} - \left( \begin{bmatrix} I_{n \times n} \\ -I_{n \times n} \\ p^T \\ -p^T \end{bmatrix} \right)^T \begin{bmatrix} \mu_1 \\ \vdots \\ \mu_{2n+2} \end{bmatrix} \end{bmatrix}$$

$$\rightarrow y_{sp} = y_{sp}(p)$$

como  $y_{sp}$  depende de la propiedad de entrada  $p$ , y como  $p$  varía con cada iteración, entonces si se puede introducir  $y_k$ .

En virtud de la suposición A.1, la afirmación dada por las ecuaciones (5.25) y (5.27) es calcular la trayectoria de la entrada de control  $x_k$  tal que  $y_k \rightarrow y_{op}$  asintóticamente. Las ecuaciones (5.25) y (5.27) se pueden combinar para dar:

$$Ax + M(x) = y_{sp}$$

$$x_k = \bar{A}^{-1} \left( y_{sp} - y_{k-1} + \bar{A}x_{k-1} \right)$$

$$x_k = x_{k-1} + \bar{A}^{-1} (y_{sp} - y_{k-1}) \tag{5.28}$$

La salida medida introduce el término de corrección  $\bar{A}^{-1} (y_{sp} - y_{k-1})$ , con el cual  $x_k$  se actualiza. Entonces es claro que si  $y_k \rightarrow y_{op}$  asintóticamente, el evento de iteración de la ecuación (5.28) es estable. Por la suposición A.1, esto implica que  $x_k \rightarrow x_{op}$  asintóticamente (por tanto  $u_k \rightarrow u_{op}$  asintóticamente.) como se desea.

Se tiene que  $y_k = Ax_k + M(x_k)$ , para todo  $k \geq 1$ . Entonces, de la ecuación (5.28) se tiene lo siguiente:

$$x_k = x_{k-1} + \bar{A}^{-1} (y_{sp} - Ax_{k-1} - M(x_{k-1})) \quad (5.29)$$

Sea  $e_k = x_k - x_{op}$  y dado que  $y_{sp} = \bar{A}x_k + B(x_{op}) = Ax_{op} + M(x_{op})$ , se pueden sustituir en la ecuación (5.29) para obtener:

$$\begin{aligned} x_k &= x_{k-1} + \bar{A}^{-1} [y_{sp} - Ax_{k-1} - M(x_{k-1})] \\ x_k &= x_{k-1} + \bar{A}^{-1} [Ax_{op} + M(x_{op}) - Ax_{k-1} - M(x_{k-1})] \\ x_k &= x_{k-1} - \bar{A}^{-1} [A(x_{k-1} - x_{op}) + M(x_{k-1}) - M(x_{op})] \\ x_k &= x_{k-1} - \bar{A}^{-1} [Ae_{k-1} + M(x_{k-1}) - M(x_{op})] \\ e_k + x_{op} &= x_{k-1} - \bar{A}^{-1} Ae_{k-1} - \bar{A}^{-1} [M(x_{k-1}) - M(x_{op})] \\ e_k &= x_{k-1} - x_{op} - \bar{A}^{-1} Ae_{k-1} - \bar{A}^{-1} [M(x_{k-1}) - M(x_{op})] \\ e_k &= e_{k-1} - \bar{A}^{-1} Ae_{k-1} - \bar{A}^{-1} [M(x_{k-1}) - M(x_{op})] \\ e_k &= (I_n - \bar{A}^{-1} A)e_{k-1} - \bar{A}^{-1} [M(e_{k-1} + x_{op}) - M(x_{op})] \\ e_k &= (I_n - \bar{A}^{-1} A)e_{k-1} - \bar{A}^{-1} [M(e_{k-1} + x_{op}) - M(x_{op})] \end{aligned} \quad (5.30)$$

Debido a la suposición A.1 la ecuación (5.30) tiene un punto de equilibrio único  $e_{eq} = 0_p$  por lo menos en una vecindad del origen. Note que, si  $M(x) = 0$  y  $A = \bar{A}$ , entonces  $e_k = 0_p$ , esto es, si no existe incertidumbre, el esquema de actualización converge en un solo paso; por lo tanto, en presencia de una pequeña incertidumbre, el esquema de actualización convergerá asintóticamente al valor óptimo  $x_{op}$ . Se afirma ésto en la siguiente proposición [12].

**Proposición 5.1** *Considere el esquema de actualización descrito anteriormente. Entonces  $u_k \rightarrow u_{op}$  si:*

$$\left\| I_n - \bar{A}^{-1} (A + J_M(x)) \right\| < 1 \quad (5.31)$$

para todo  $x \in R^p$ , donde  $J_M(x)$  es la matriz jacobiana de la no linealidad  $M(x)$ .

Con éste resultado, se siguen las siguientes comentarios:

1. La principal conclusión de la proposición es que, comparado a la regla de mezclado ideal, las no linealidades podrían ser lo suficientemente pequeñas tal que garanticen la convergencia de la programación lineal con el esquema de actualización. Por otro lado, si la no linealidad de la mezcla afecta demasiado, se podrían presentar inestabilidades. Este hecho limita ampliamente la aplicación de la programación lineal con el esquema de actualización para resolver problemas de control de mezclado con efectos considerables en la no linealidad.
2. En algunas situaciones prácticas, la no linealidad  $N(u)$  tiene una pequeña contribución a la propiedad de la mezcla. Si  $I_n - \bar{A}^{-1}A$  es suficientemente pequeño, de ésto se espera que  $J_M(x)$  actúe como una perturbación a la matriz  $A$ , de esta forma, si se satisface la ecuación (5.31). Un ejemplo de ésto es el caso de la regla de mezclado  $f_{MR}(u, p) = \left[ \sum_{i=1}^n u_i p_i^\beta \right]^{1/\beta}$  para valores pequeños de  $|\beta - 1|$ . En tal caso, la no linealidad de los componentes o desviación de lo ideal está dado por  $p_{b.nl}(u) = \left[ \sum_{i=1}^n u_i p_i^\beta \right]^{1/\beta} - \sum_{i=1}^n u_i p_i$ . Note que  $p_{b.nl}(u) \rightarrow 0$  cuando  $\beta \rightarrow 1$ , así que, implica que  $|p_{b.nl}(u)|$  es pequeño para valores pequeños de  $|\beta - 1|$ .
3. Se ha considerado que la matriz de las propiedades de torrente no es medible, en la práctica, ésta la matriz se puede medir a través de la actualización de datos correspondiente a la matriz  $\bar{A}$ . De esta forma, la ecuación (4.9) se puede generalizar al considerar las fluctuaciones de la matriz  $A$  y las medidas periódicas de la propiedad dando la siguiente expresión:

$$e_k = (I_n - \bar{A}_{k-1}^{-1}A_{k-1})e_{k-1} - \bar{A}_{k-1}^{-1} [M(e_{k-1} + x_{op}) - M(x_{op})]$$

donde  $A_k$  es el valor de  $A$  en un instante y  $\bar{A}_k$  es su correspondiente valor estimado. Note que en los sistemas de mezclado comerciales, las mediciones de las variaciones en

las propiedades se emplean para introducir un esquema de control prealimentado. De este modo, el impacto de las variaciones de las propiedades de los componentes sobre las propiedades del producto se reducen drásticamente a través de la compensación (prealimentación).

4. El esquema de actualización se puede ver como una acción integral sobre el error de regulación  $y_{sp} - y$ . De hecho, de la ecuación (5.28) se puede obtener la siguiente ecuación:

$$x_k = x_0 + \bar{A}^{-1} \sum_{j=1}^{k-1} (y_{sp} - y_j)$$

donde  $\bar{A}^{-1}$  juega el papel de una ganancia integral. En el caso continuo, tal acción de control es reescrita así  $x(t) = x_0 + \bar{A}^{-1} \int_0^t (y_{sp} - y(t)) dt$ . De este modo, el esquema de actualización adiciona una acción de control integral dentro del controlador optimizante para compensar los errores modelo-planta.

### 5.3.2. Análisis de robustez del controlador optimizante con redes neuronales

Cuando se emplean redes neuronales para la no linealidad para la ecuación (5.22) se tiene [12][9]:

$$Ax + M(x) = Ax + \varphi(v) + \delta = y_{sp} \quad (5.32)$$

con:

$$\begin{aligned} y &= \varphi(v) + \delta \\ v &= w^T u \end{aligned}$$

donde:

$\delta$  = error entre la salida deseada y la salida de la red neuronal

$w$  = pesos sinópticos que conectan las neuronas

$y$  = salida de la red neuronal

$\varphi(\cdot)$  = función no lineal (tangente hiperbólica)

De la ecuación (5.32)  $\varphi(v)$  es un término que se puede conocer a través de la red neuronal, por lo tanto se puede considerar como un término constante el cual puede pasar al otro lado de la igualdad, de ésta forma, la ecuación (5.32) se puede reescribir de la siguiente manera:

$$Ax + \delta = y_a \tag{5.33}$$

donde:

$$y_a = y_{sp} - \varphi(v)$$

y además se cumple lo siguiente:

$$\delta \ll M(x) \tag{5.34}$$

De la ecuación (5.34) se tiene que el error entre la salida deseada y la salida de la red neuronal ( $\delta$ ) tiene un valor mucho menor que el término que depende de la no linealidad ( $M(x)$ ), con lo que para el caso donde se emplean redes neuronales el valor no lineal desconocido es menor al caso donde no se emplean y con esto se asegura que tenga una mayor robustez el controlador optimizante con redes neuronales descrito mediante la ecuación (5.33) al caso donde se no emplean éstas descrito mediante la ecuación (5.22) debido a lo analizado en la proposición del tema anterior (donde se encuentra el ecuación (5.31)) y su primer comentario.

## 5.4. Simulación de la optimización fuera de línea del mezclado de crudo con redes neuronales

### 5.4.1. Obtención de los datos y algoritmo empleado

El proceso de mezclado de aceite crudo esta sujeto a un número de operaciones de restricción. Los objetivos del mezclado de petróleo crudo son maximizar el valor de todos los productos mientras se satisfacen sus restricciones, o minimizar el costo del mezclado. El

## 5.4 Simulación de la optimización fuera de línea del mezclado de crudo con redes neuronales

mezclado de aceite crudo se puede considerar como una mezcla de  $n$  componentes con:

$$\begin{aligned}
 q &= [q_1, \dots, q_n]^T = \text{velocidad de flujo en la entrada} \\
 p &= [p_1, \dots, p_n]^T = \text{propiedad en la entrada} \\
 q_f &= \text{velocidad de flujo en la salida} \\
 p_f &= \text{propiedad en la salida} \\
 u &= \left[ \frac{q_1}{q_f}, \dots, \frac{q_n}{q_f} \right]^T = \text{fracción de flujo en la entrada}
 \end{aligned}$$

En este programa se emplea el algoritmo de la optimización fuera de línea del petróleo crudo con redes neuronales y con esquema de optimización de la Figura (5.1) y de la ecuación (5.15) la cual se vuelve a citar a continuación:

$$\begin{aligned}
 &\text{mín } c^T u(k) \\
 &\text{sujeto a:} \\
 &Au(k) \leq b \\
 &A_{eq}u(k) = b_{eq} \\
 &u_{\text{mín}} \leq u(k) \leq u_{\text{máx}}
 \end{aligned}$$

donde:

$$\begin{aligned}
 A &= \begin{bmatrix} \frac{p}{q_f} \\ -\frac{p}{q_f} \end{bmatrix} = \begin{bmatrix} p \\ -p \end{bmatrix} \\
 b &= \begin{bmatrix} p_{f,\text{máx}} - \Delta(k) \\ -p_{f,\text{mín}} + \Delta(k) \end{bmatrix} \\
 A_{eq} &= [1, \dots, 1] \\
 b_{eq} &= u_f = 1
 \end{aligned}$$

$$\Delta(k) = p_f(k-1) - p^T(k-1)u(k-1)$$

Se puede evitar la suposición del analizador en tiempo real con esquema de actualización al emplear la red neuronal para estimar el comportamiento del mezclador con datos históricos, después del aprendizaje de la red neuronal se tiene el valor de salida de la red neuronal como las propiedades cualitativas del proceso de mezclado que se describe mediante la ecuación (5.11) la cual se vuelve a citar a continuación:

$$\hat{p}_f = V_k \phi [W_k X(k)]$$

De una condición inicial para  $\Delta$ , por ejemplo  $\Delta(0) = 0$ ,  $\Delta$  que es el parámetro que describe el esquema de actualización se puede estimar mediante la ecuación (5.8) después de calcular  $u$  (fracción de flujo en la entrada) mediante de la optimización. El esquema de actualización se vuelve a citar a continuación:

$$\Delta(k + 1) = \hat{p}_f(k) - \frac{q^T(k)p(k)}{q_f} = \hat{p}_f(k) - p^T(k)u(k)$$

En este caso la propiedad cualitativa es la densidad a la salida del modelo la cual se obtiene directamente de las estimaciones que se analizaron en el capítulo anterior para los casos del modelo distribuido (mezclador 1, mezclador 2 y mezclador 3), del modelo agregado, para cuando se aplicó la red neuronal para el modelo completo y cuando se aplicó la red neuronal solamente para la no linealidad. Para este programa se emplea la función de Matlab llamada `linprog` (programación lineal) que se define como sigue:

$$x = \text{LINPROG}(c, A1, b1) \tag{5.35}$$

La ecuación (5.35) de Matlab resuelve en un problema de programación lineal:

$$\begin{aligned} & \text{mín } c^T x \\ & \text{sujeto a:} \\ & A1x \leq b1 \end{aligned} \tag{5.36}$$

donde para el programa  $x = u$  (fracción de flujo en la entrada).

$$x = \text{LINPROG}(c, A1, b1, A0, b0) \tag{5.37}$$

La ecuación (5.37) de Matlab resuelve el problema de la ecuación (5.36) mientras adicionalmente satisface las restricciones de igualdad:

$$A0x = b0$$

$$x = \text{LINPROG}(c, A1, b1, A0, b0, xi, xa) \tag{5.38}$$

La ecuación (5.38) de Matlab define un espacio de cotas inferior y superior sobre las variables de diseño (que para nuestro caso son las fracciones de flujo de las entradas del modelo), tal que el programa de una solución dentro de este rango.

$$x = \text{LINPROG}(c, A1, b1, A0, b0, xi, xa, xx) \quad (5.39)$$

La ecuación (5.39) da un punto inicial a  $x$ , denotado por  $xx$ .

A continuación se muestran los resultados de emplear este algoritmo para el controlador de cada uno de los modelos mencionados.

### 5.4.2. Optimización del modelo distribuido

Para el controlador optimizante del mezclado de crudo con redes neuronales del modelo distribuido se consideran cada uno de los 3 mezcladores por separado teniéndose un programa para cada mezclador. Se obtienen los datos de los programas de Matlab para redes neuronales del capítulo anterior los cuales se emplean para realizar cada programa. Se tienen 6 programas para el controlador debido a que hay dos modalidades que son para el modelo completo y para la no linealidad las cuales se explicaron en el capítulo anterior.

#### Optimización con redes neuronales para el modelo completo

A continuación se grafican los resultados, que son cuatro gráficas en una Figura y una Figura para cada mezclador (ver Figuras 5.2, 5.3 y 5.4):

- La primera gráfica proporciona los valores óptimos de las fracciones de flujo de los componentes de entrada del modelo los cuales son proporcionados por el controlador para el mezclado de crudo con redes neuronales.
- La segunda gráfica proporciona los valores óptimos de las fracciones de flujo de los componentes de entrada del modelo los cuales son proporcionados por el controlador para el mezclado de crudo con redes neuronales.

- La tercer gráfica proporciona la función de costo óptima para el controlador empleado en el mezclado de crudo con redes neuronales.
- La cuarta gráfica proporciona la función de costo óptima para el controlador empleado en el mezclado de crudo con datos reales.

Para los programas de las Figuras 5.2, 5.3 y 5.4 se tiene una función de costo de  $c=[34;32]$ , una matriz para la restricción de igualdad de  $A0=[1,1]$ , un elemento para la restricción de igualdad de  $b0=1$ , una cota inferior de  $xi=[0;0]$ , una densidad mínima a la salida del modelo de  $Pi=30$ , una densidad máxima a la salida del modelo de  $Pa=35$ , el valor inicial de la no linealidad para el esquema de actualización es  $\delta(1)=0$ , una condición inicial  $xx=0$  y una matriz para la restricción de desigualdad de  $A1=[DE;-DE]$  donde  $DE$  es la densidad de los elementos de entrada del modelo.

En la Figura 5.2 se simuló el controlador para el mezclado de crudo con redes neuronales, 2 entradas, 1 salida y una cota superior de  $x_a=[0.6;0.8]$ . De la gráfica del inciso (a) de la Figura 5.2 donde se encuentran los valores óptimos para las fracciones de flujo para el modelo completo del mezclador 1 con redes neuronales se observa que ambas señales son estables con valores  $u_1 = 0,2$ ,  $u_2 = 0,8$  y que  $\sum_{i=1}^n u_i = 1$  teniéndose por consiguiente una optimización exitosa. De la gráfica del inciso (b) de la Figura 5.2 donde se encuentran los valores óptimos para las fracciones de flujo para el modelo completo del mezclador 1 con datos reales se observa que ambas señales son estables con valores  $u_1 = 0,2$ ,  $u_2 = 0,8$  y que  $\sum_{i=1}^n u_i = 1$  teniéndose por consiguiente una optimización exitosa. De la gráfica del inciso (c) de la Figura 5.3 donde se encuentra la función de costo óptima para el modelo completo del mezclador 1 con redes neuronales se observa que tiene un valor final de 32,4. De la gráfica del inciso (d) de la Figura 5.3 donde se encuentra la función de costo óptima para el modelo completo del mezclador 1 con datos reales se observa que tiene un valor final de 32,4.

En la Figura 5.3 se simuló el controlador para el mezclado de crudo con redes neuronales, 2 entradas, 1 salida y una cota superior de  $x_a=[0.8;0.6]$ . De la gráfica del inciso (a) de la Figura 5.3 donde se encuentran los valores óptimos para las fracciones de flujo para el modelo

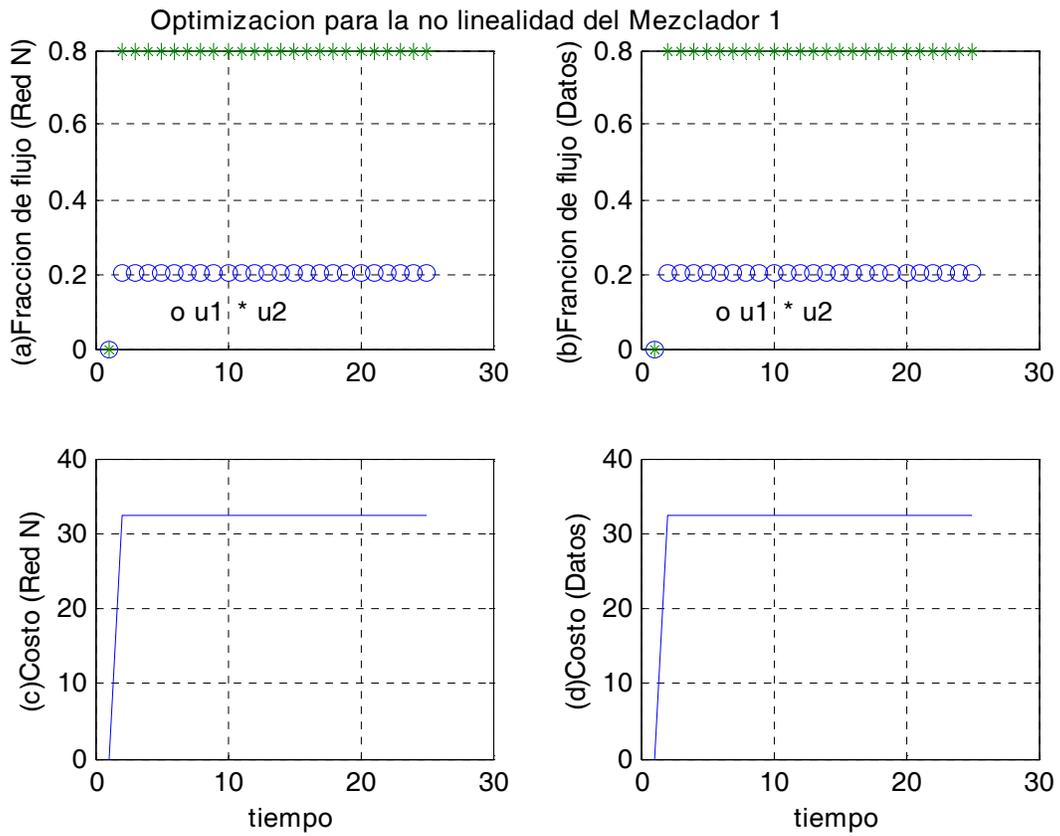


Figura 5.2: Optimización para el modelo completo del Mezclador 1

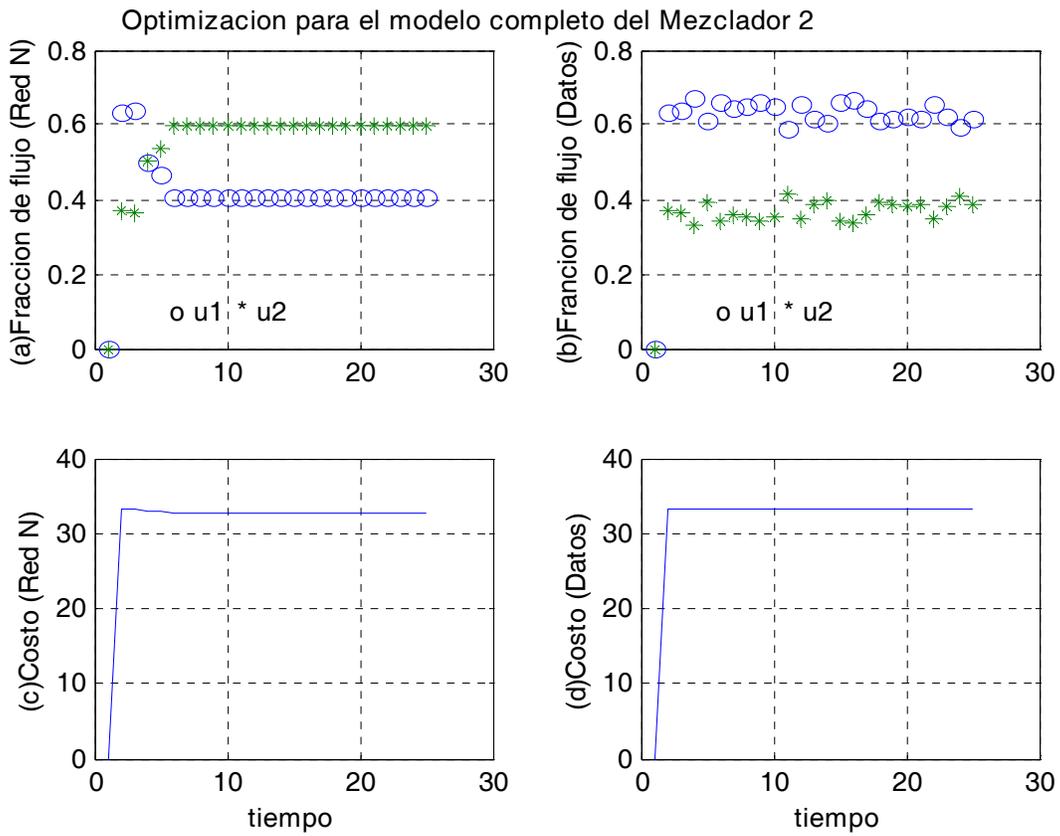


Figura 5.3: Optimización para el modelo completo del Mezclador 2

completo del mezclador 2 con redes neuronales se observa que ambas señales son estables con valores  $u_1 = 0,4$ ,  $u_2 = 0,6$  y que  $\sum_{i=1}^n u_i = 1$  teniéndose por consiguiente una optimización exitosa. De la gráfica del inciso (b) de la Figura 5.3 donde se encuentran los valores óptimos para las fracciones de flujo para el modelo completo del mezclador 2 con datos reales se observa que ambas señales son no estables con valores  $u_1 = 0,6778$ ,  $u_2 = 0,3222$  y que  $\sum_{i=1}^n u_i = 1$  teniéndose por consiguiente una optimización no exitosa. De la gráfica del inciso (c) de la Figura 5.3 donde se encuentra la función de costo óptima para el modelo completo del mezclador 2 con redes neuronales se observa que tiene un valor final de 32,8. De la gráfica del inciso (d) de la Figura 5.3 donde se encuentra la función de costo óptima para el modelo completo del mezclador 2 con datos reales se observa que tiene un valor final de 33,3555.

En la Figura 5.4 se simuló controlador para el mezclado de crudo con redes neuronales, 2 entradas, 1 salida y una cota superior de  $x_a = [0,6; 0,6]$ . De la gráfica del inciso (a) de la Figura 5.4 donde se encuentran los valores óptimos para las fracciones de flujo para el modelo completo del mezclador 3 con redes neuronales se observa que ambas señales son estables con valores  $u_1 = 0,4$ ,  $u_2 = 0,6$  y que  $\sum_{i=1}^n u_i = 1$  teniéndose por consiguiente una optimización exitosa. De la gráfica del inciso (b) de la Figura 5.4 donde se encuentran los valores óptimos para las fracciones de flujo para el modelo completo del mezclador 3 con datos reales se observa que ambas señales son estables con valores  $u_1 = 0,4$ ,  $u_2 = 0,6$  y que  $\sum_{i=1}^n u_i = 1$  teniéndose por consiguiente una optimización exitosa. De la gráfica del inciso (c) de la Figura 5.4 donde se encuentra la función de costo óptima para el modelo completo del mezclador 3 con redes neuronales se observa que tiene un valor final de 32,8. De la gráfica del inciso (d) de la Figura 5.4 donde se encuentra la función de costo óptima para el modelo completo del mezclador 3 con datos reales se observa que tiene un valor final de 32,8.

### Optimización con redes neuronales para la parte no lineal

A continuación se grafican los resultados, que son cuatro gráficas en una Figura y una Figura para cada mezclador (ver Figuras 5.5, 5.6 y 5.7):

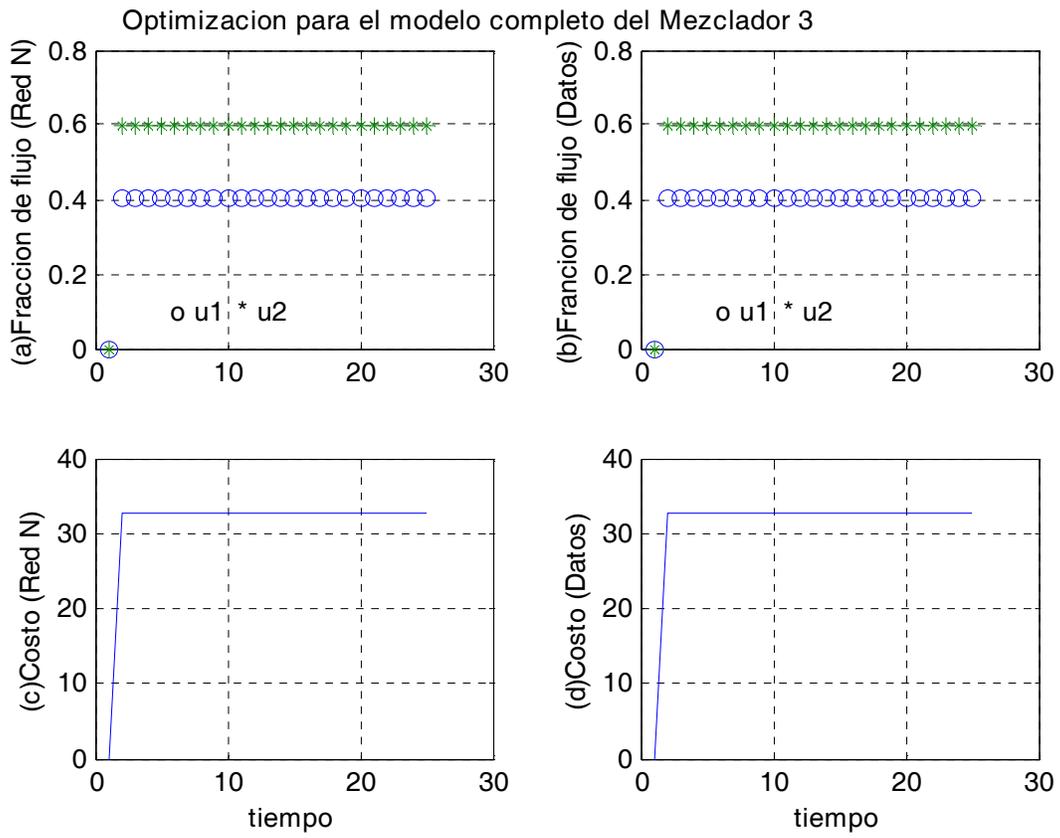


Figura 5.4: Optimización para el modelo completo del Mezclador 3

## 5.4 Simulación de la optimización fuera de línea del mezclado de crudo con redes neuronales

- La primera gráfica proporciona los valores óptimos de las fracciones de flujo de los componentes de entrada del modelo los cuales son proporcionados por el controlador para el mezclado de crudo con redes neuronales.
- La segunda gráfica proporciona los valores óptimos de las fracciones de flujo de los componentes de entrada del modelo los cuales son proporcionados por el controlador para el mezclado de crudo con redes neuronales.
- La tercera gráfica proporciona la función de costo óptima para el controlador empleado en el mezclado de crudo con redes neuronales.
- La cuarta gráfica proporciona la función de costo para el controlador empleado en el mezclado de crudo con datos reales.

Para los programas de las Figuras 5.5, 5.6 y 5.7 se tiene una función de costo de  $c=[34;32]$ , una matriz para la restricción de igualdad de  $A0=[1,1]$ , un elemento para la restricción de igualdad de  $b0=1$ , una cota inferior de  $x_i=[0;0]$ , una densidad mínima a la salida del modelo de  $P_i=30$ , una densidad máxima a la salida del modelo de  $P_a=35$ , el valor inicial de la no linealidad para el esquema de actualización es  $\delta(1)=0$ , una condición inicial  $xx=0$  y una matriz para la restricción de desigualdad de  $A1=[DE;-DE]$  donde  $DE$  es la densidad de los elementos de entrada del modelo.

En la Figura 5.5 se simuló el controlador para el mezclado de crudo con redes neuronales, 2 entradas, 1 salida y una cota superior de  $x_a=[0.6;0.8]$ . De la gráfica del inciso (a) de la Figura 5.5 donde se encuentran los valores óptimos para las fracciones de flujo para la no linealidad del mezclador 1 con redes neuronales se observa que ambas señales son estables con valores  $u_1 = 0,2$ ,  $u_2 = 0,8$  y que  $\sum_{i=1}^n u_i = 1$  teniéndose por consiguiente una optimización exitosa. De la gráfica del inciso (b) de la Figura 5.5 donde se encuentran los valores óptimos para las fracciones de flujo para la no linealidad del mezclador 1 con datos reales se observa que ambas señales son estables con valores  $u_1 = 0,2$ ,  $u_2 = 0,8$  y que  $\sum_{i=1}^n u_i = 1$  teniéndose por consiguiente una optimización exitosa. De la gráfica del inciso (c) de la Figura 5.5 donde

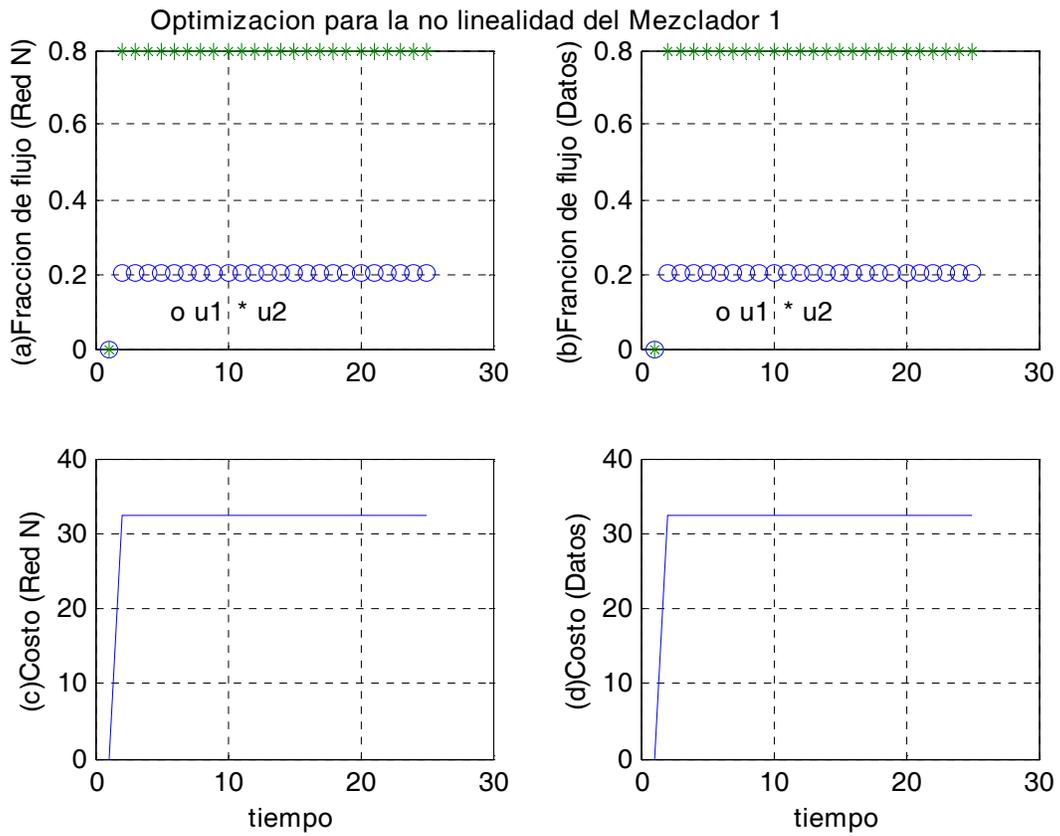


Figura 5.5: Optimización para la no linealidad del Mezclador 1

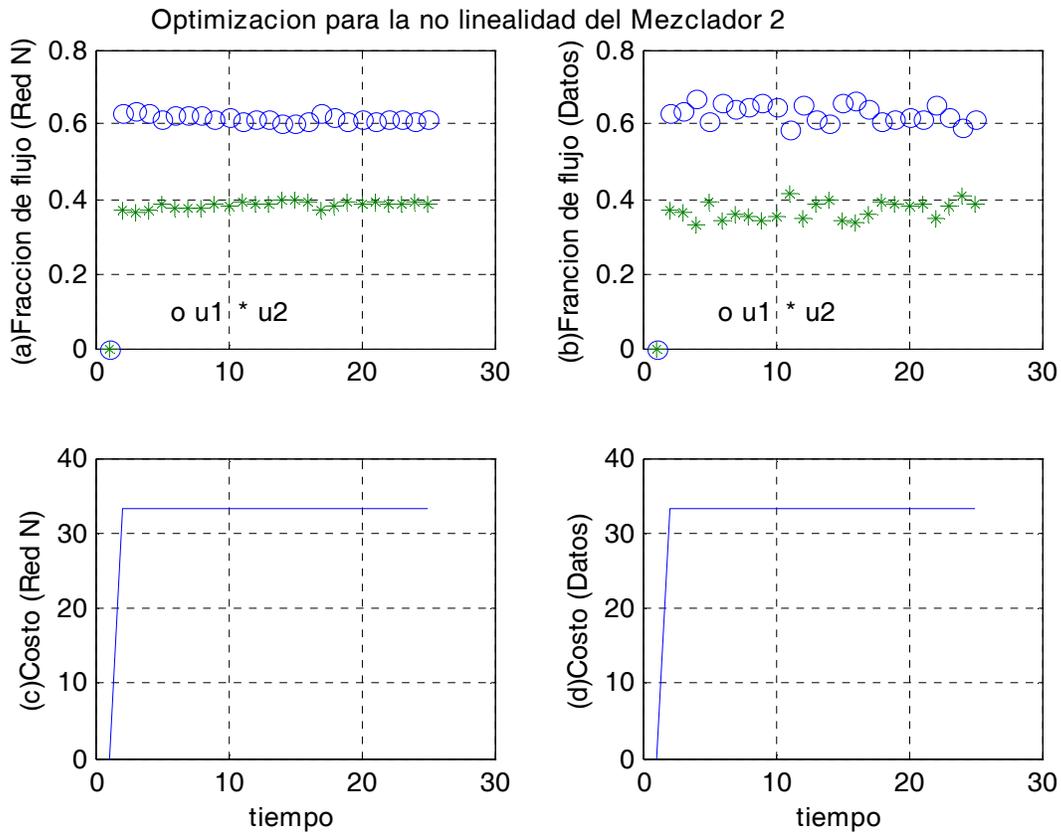


Figura 5.6: Optimización para la no linealidad del Mezclador 2

se encuentra la función de costo óptima para la no linealidad del mezclador 1 con redes neuronales se observa que tienen un valor final de 32,4. De la gráfica del inciso (d) de la Figura 5.5 donde se encuentra la función de costo óptima para la no linealidad del mezclador 1 con datos reales se observa que tienen un valor final de 32,4.

En la Figura 5.6 se simuló el controlador para el mezclado de crudo con redes neuronales, 2 entradas, 1 salida y una cota superior de  $x_a=[0.8;0.6]$ . De la gráfica del inciso (a) de la Figura 5.6 donde se encuentran los valores óptimos para las fracciones de flujo para la no linealidad del mezclador 2 con redes neuronales se observa que ambas señales son estables

con valores  $u_1 = 0,6269$ ,  $u_2 = 0,3731$  y que  $\sum_{i=1}^n u_i = 1$  teniéndose por consiguiente una optimización exitosa. De la gráfica del inciso (b) de la Figura 5.6 donde se encuentran los valores óptimos para las fracciones de flujo para la no linealidad del mezclador 2 con datos reales se observa que ambas señales son no estables con valores  $u_1 = 0,6778$ ,  $u_2 = 0,3222$  y que  $\sum_{i=1}^n u_i = 1$  teniéndose por consiguiente una optimización no exitosa. De la gráfica del inciso (c) de la Figura 5.6 donde se encuentra la función de costo óptima para la no linealidad del mezclador 2 con redes neuronales se observa que tienen un valor final de 33,2537. De la gráfica del inciso (d) de la Figura 5.6 donde se encuentra la función de costo óptima para la no linealidad del mezclador 2 con datos reales se observa que tienen un valor final de 33,3555.

En la Figura 5.7 se simuló el controlador para el mezclado de crudo con redes neuronales, 2 entradas, 1 salida y una cota superior de  $x_a=[0,6;0,6]$ . De la gráfica del inciso (a) de la Figura 5.7 donde se encuentran los valores óptimos para las fracciones de flujo para la no linealidad del mezclador 3 con redes neuronales se observa que ambas señales son estables con valores  $u_1 = 0,4$ ,  $u_2 = 0,6$  y que  $\sum_{i=1}^n u_i = 1$  teniéndose por consiguiente una optimización exitosa. De la gráfica del inciso (b) de la Figura 5.7 donde se encuentran los valores óptimos para las fracciones de flujo para la no linealidad del mezclador 3 con datos reales se observa que ambas señales son estables con valores  $u_1 = 0,4$ ,  $u_2 = 0,6$  y que  $\sum_{i=1}^n u_i = 1$  teniéndose por consiguiente una optimización exitosa. De la gráfica del inciso (c) de la Figura 5.7 donde se encuentra la función de costo óptima para la no linealidad del mezclador 3 con redes neuronales se observa que tienen un valor final de 32,8. De la gráfica del inciso (d) de la Figura 5.7 donde se encuentra la función de costo óptima para la no linealidad del mezclador 3 con datos reales se observa que tienen un valor final de 32,8.

### 5.4.3. Optimización del modelo agregado

Para el modelo agregado se consideran los 3 mezcladores en un modelo teniéndose un programa para todos los mezcladores. Se obtienen los datos de los programa de Matlab para

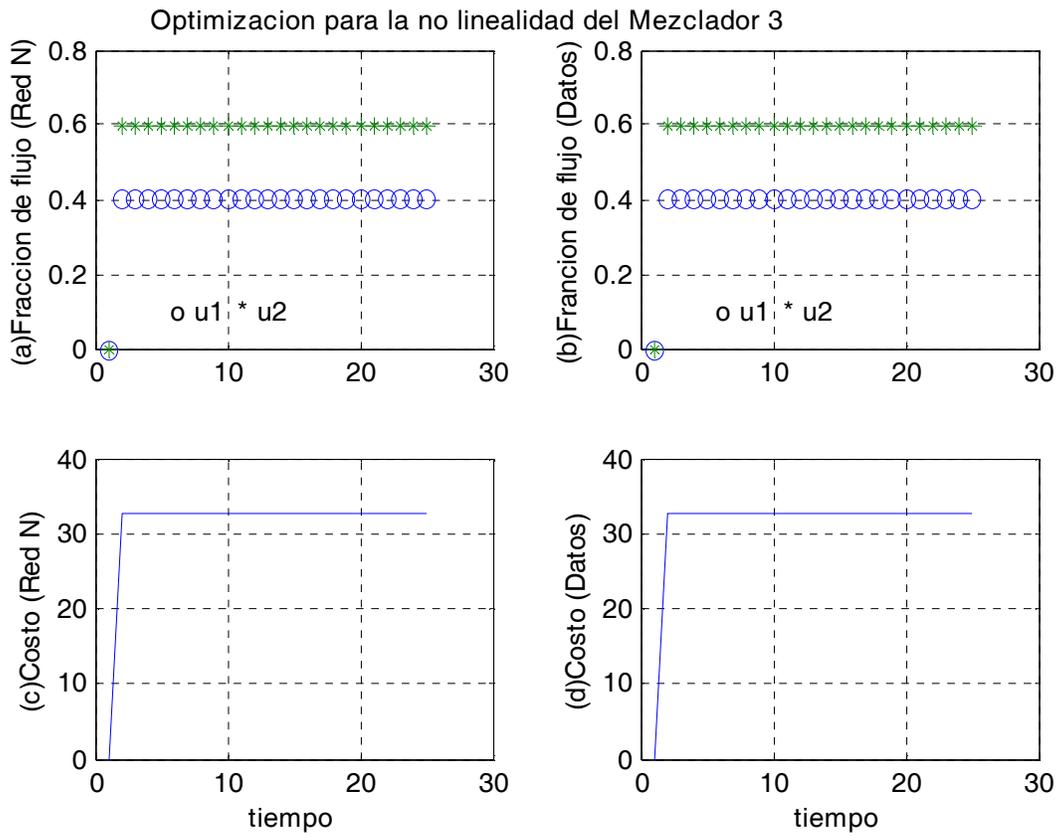


Figura 5.7: Optimización para la no linealidad del Mezclador 3

redes neuronales del capítulo anterior los cuales se emplean para realizar el programa. Se tienen 2 programas para el controlador debido a que hay dos modalidades que son para el modelo completo y para la no linealidad las cuales se explicaron en el capítulo anterior.

### Optimización con redes neuronales para el modelo completo

A continuación se grafican los resultados, que son cuatro gráficas en una Figura (ver Figura 5.8):

- La primer gráfica proporciona los valores óptimos de las fracciones de flujo de los componentes de entrada del modelo lo cuales son proporcionados por el controlador para el mezclado de crudo con redes neuronales.
- La segunda gráfica proporciona los valores óptimos de las fracciones de flujo de los componentes de entrada del modelo lo cuales son proporcionados por el controlador para el mezclado de crudo con redes neuronales.
- La tercer gráfica proporciona la función de costo óptima para el controlador empleado en el mezclado de crudo con redes neuronales.
- La cuarta gráfica proporciona la función de costo para el controlador empleado en el mezclado de crudo con datos reales.

Para el programa de la Figura 5.8 se tiene una función de costo de  $c=[34;32;10;20]$ , una matriz para la restricción de igualdad de  $A0=[1,1,1,1]$ , un elemento para la restricción de igualdad de  $b0=1$ , una cota inferior de  $xi=[0;0;0;0]$ , una densidad mínima a la salida del modelo de  $Pi=30$ , una densidad máxima a la salida del modelo de  $Pa=35$ , el valor inicial de la no linealidad para el esquema de actualización es  $\delta(1)=0$ , una condición inicial  $xx=0$  y una matriz para la restricción de desigualdad de  $A1=[DE;-DE]$  donde DE es la densidad de los elementos de entrada del modelo.

En la Figura 5.8 se simuló el controlador para el mezclado de crudo con redes neuronales, 4 entradas, 1 salida y una cota superior de  $xa=[0.6;0.5;0.5;0.2]$ . De la gráfica del inciso (a)

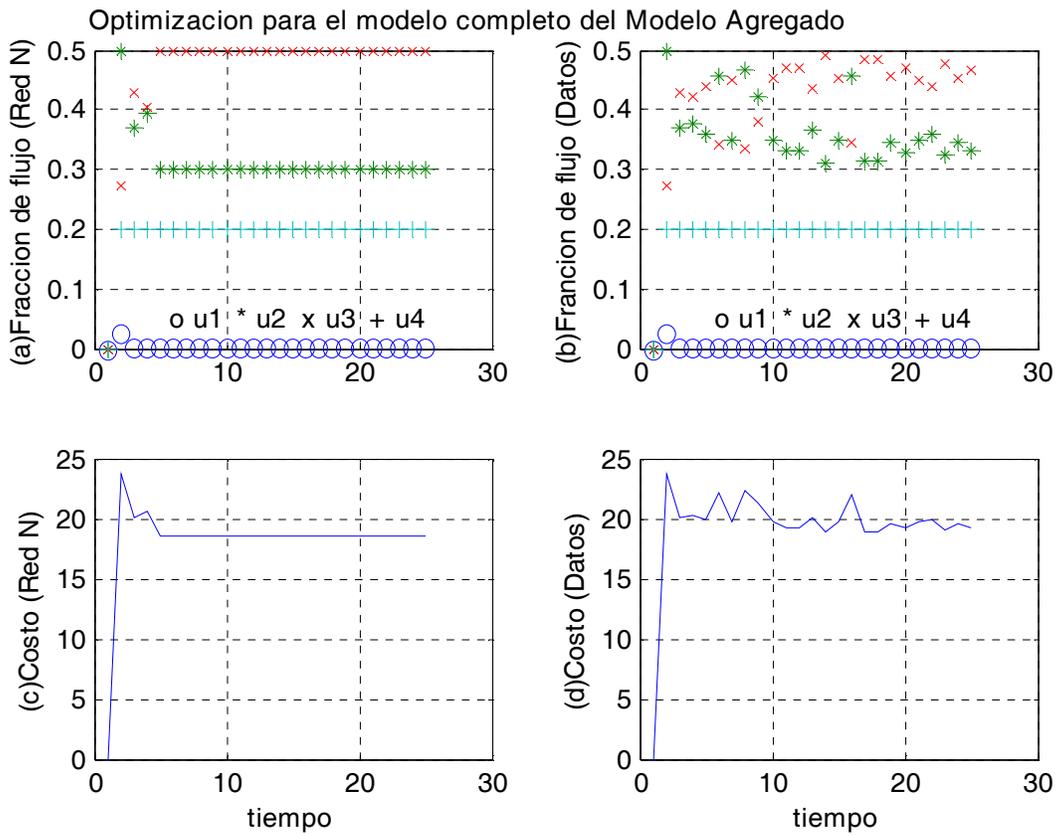


Figura 5.8: Optimización para el modelo completo del Modelo Agregado

de la Figura 5.8 donde se encuentran los valores óptimos para las fracciones de flujo para el modelo completo del modelo agregado con redes neuronales se observa que todas las señales son estables con valores  $u_1 = 0,0$ ,  $u_2 = 0,3$ ,  $u_3 = 0,5$ ,  $u_4 = 0,2$  y que  $\sum_{i=1}^n u_i = 1$  teniéndose por consiguiente una optimización exitosa. De la gráfica del inciso (b) de la Figura 5.8 donde se encuentran los valores óptimos para las fracciones de flujo para el modelo completo del modelo agregado con datos reales se observa que no todas las señales son estables con valores  $u_1 = 0,0$ ,  $u_2 = 0,3725$ ,  $u_3 = 0,4275$ ,  $u_4 = 0,2$  y que  $\sum_{i=1}^n u_i = 1$  teniéndose por consiguiente una optimización no exitosa. De la gráfica del inciso (c) de la Figura 5.8 donde se encuentra la función de costo óptima para el modelo completo del modelo agregado con redes neuronales se observa que tienen un valor final de 18,6. De la gráfica del inciso (d) de la Figura 5.8 donde se encuentra la función de costo óptima para el modelo completo del modelo agregado con datos reales se observa que tienen un valor final de 20,1958.

### Optimización con redes neuronales para la parte no lineal

A continuación se grafican los resultados, que son cuatro gráficas en una Figura (ver Figura 5.9):

- La primer gráfica proporciona los valores óptimos de las fracciones de flujo de los componentes de entrada del modelo los cuales son proporcionados por el controlador para el mezclado de crudo con redes neuronales.
- La segunda gráfica proporciona los valores óptimos de las fracciones de flujo de los componentes de entrada del modelo lo cuales son proporcionados por el controlador para el mezclado de crudo con redes neuronales.
- La tercer gráfica proporciona la función de costo óptima para el controlador empleado en el mezclado de crudo con redes neuronales.
- La cuarta gráfica proporciona la función de costo para el controlador empleado en el mezclado de crudo con datos reales.

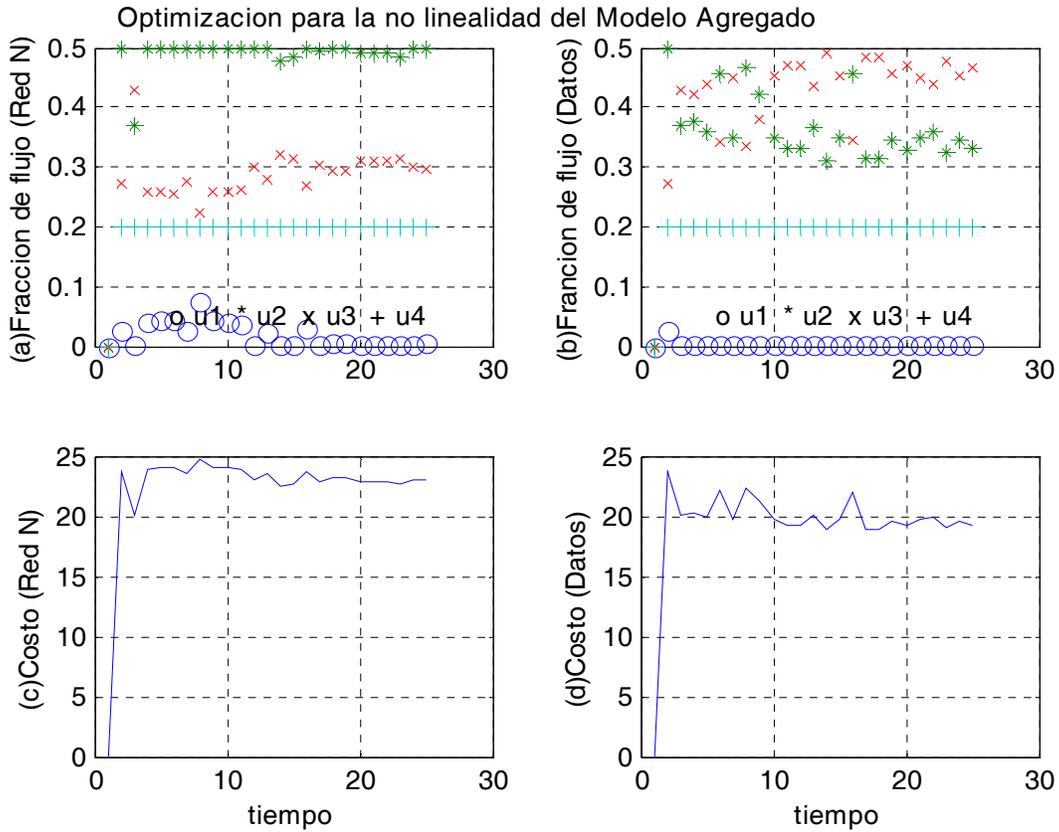


Figura 5.9: Optimización para la no linealidad del Modelo Agregado

Para el programa de la Figura 5.9 se tiene una función de costo de  $c=[34;32;10;20]$ , una matriz para la restricción de igualdad de  $A0=[1,1,1,1]$ , un elemento para la restricción de igualdad de  $b0=1$ , una cota inferior de  $xi=[0;0;0;0]$ , una densidad mínima a la salida del modelo de  $Pi=30$ , una densidad máxima a la salida del modelo de  $Pa=35$ , el valor inicial de la no linealidad para el esquema de actualización es  $\delta(1)=0$ , una condición inicial  $xx=0$  y una matriz para la restricción de desigualdad de  $A1=[DE;-DE]$  donde  $DE$  es la densidad de los elementos de entrada del modelo.

En la Figura 5.9 se simuló el controlador para el mezclado de crudo con redes neuronales, 4 entradas, 1 salida y una cota superior de  $x_a=[0.6;0.5;0.5;0.2]$ . De la gráfica del inciso (a) de la Figura 5.9 donde se encuentran los valores óptimos para las fracciones de flujo para la no linealidad del modelo agregado con redes neuronales se observa que todas las señales son estables con valores  $u_1 = 0,0$ ,  $u_2 = 0,4953$ ,  $u_3 = 0,3047$ ,  $u_4 = 0,2$  y que  $\sum_{i=1}^n u_i = 1$  teniéndose por consiguiente una optimización exitosa. De la gráfica del inciso (b) de la Figura 5.9 donde se encuentran los valores óptimos para las fracciones de flujo para la no linealidad del modelo agregado con datos reales se observa que no todas las señales son estables con valores  $u_1 = 0,0$ ,  $u_2 = 0,3725$ ,  $u_3 = 0,4275$ ,  $u_4 = 0,2$  y que  $\sum_{i=1}^n u_i = 1$  teniéndose por consiguiente una optimización no exitosa. De la gráfica del inciso (c) de la Figura 5.9 donde se encuentra la función de costo óptima para la no linealidad del modelo agregado con redes neuronales se observa que tienen un valor final de 22,8969. De la gráfica del inciso (d) de la Figura 5.9 donde se encuentra la función de costo óptima para la no linealidad del modelo agregado con datos reales se observa que tienen un valor final de 20,1958.

## 5.5. Conclusión

En ésta sección se realiza un análisis de la robustez del controlador optimizante, el principal resultado establece que el controlador optimizante converge al valor óptimo siempre que la parte no lineal (o desviación de lo ideal) de la regla de mezclado y las variaciones en los elementos de alimentación tengan una contribución muy pequeña sobre las propiedades cualitativas de mezclado. Esto restringe el empleo del controlador optimizante para sistemas de mezclado donde las variaciones del sistema son insignificantes. Además, se realiza un análisis de robustez para el controlador optimizante con redes neuronales, donde, se demuestra el se tiene una mayor robustez para el controlador optimizante con redes neuronales que para el que no las emplea dado que para el que emplea redes neuronales se tiene un valor de la no linealidad aún menor que para el caso donde no las emplea.

En el desarrollo de esta sección se realiza una comparación entre el controlador optimizante para el mezclado de crudo con redes neuronales contra el controlador optimizante para el mezclado de crudo con datos reales teniéndose mejores resultados para el que emplea redes neuronales dado que las fracciones de flujo de los elementos de entrada fueron más estables y se tuvo un menor valor en la función de costo, comprobándose con esto que es recomendable el empleo de un controlador optimizante con redes neuronales para tener resultados más estables y un menor costo. Tomando en consideración la función de costo el mejor de todos es el controlador optimizante con redes neuronales para el modelo completo. El resultado anterior se corrobora mediante la siguiente tabla.

	M1 = Mezclador 1
CORN = Controlador optimizante con redes neuronales	M2 = Mezclador 2
CODR = Controlador optimizante con datos reales	M3 = Mezclador 3
	MA = M.Agregado

## Aplicado al modelo completo

	$u_1$	$u_2$	$u_3$	$u_4$	Costo	Estabilidad
CORN1	0,2	0,8	—	—	32,4	Estable
CORN2	0,4	0,6	—	—	32,8	Estable
CORN3	0,4	0,6	—	—	32,8	Estable
CORN4	0,0	0,3	0,5	0,2	18,6	Estable
CODR1	0,2	0,8	—	—	32,4	Estable
CODR2	0,6778	0,3222	—	—	33,3555	No estable
CODR3	0,4	0,6	—	—	32,8	Estable
CODR4	0,0	0,3725	0,4275	0,2	20,1958	No estable

## Aplicado al la no linealidad

	$u_1$	$u_2$	$u_3$	$u_4$	Costo	Estabilidad
CORN1	0,2	0,8	—	—	32,4	Estable
CORN2	0,6269	0,3731	—	—	33,2537	Estable
CORN3	0,4	0,6	—	—	32,8	Estable
CORN4	0,0	0,4953	0,3047	0,2	22,8969	Estable
CODR1	0,2	0,8	—	—	32,4	Estable
CODR2	0,6778	0,3222	—	—	33,3555	No estable
CODR3	0,4	0,6	—	—	32,8	Estable
CODR4	0,0	0,3725	0,4275	0,2	20,1958	No estable

# Capítulo 6

## Conclusiones

El principal resultado obtenido en este trabajo muestra que la aplicación de un control optimizante con redes neuronales en el control de procesos de mezclado de petróleo crudo, garantiza entregar una mezcla con las condiciones de calidad requeridas a un costo mínimo. La convergencia del controlador propuesto a los valores óptimos dependerá de que la no linealidad en la mezcla y los cambios en las corrientes de entrada sean lo suficientemente pequeños en la contribución en la densidad de salida de la mezcla.

En la representación distribuida de un proceso de mezclado el controlador solo tiene información de las capacidades de las corrientes de entrada del mezclador al que se aplica, desconociendo las del que le precede. Esto ocasiona un aumento en la sensibilidad a cambios y una disminución del espacio de solución de los controladores que le preceden. En cuanto a la representación agregada, el controlador tiene información de las capacidades de todas las corrientes de entrada al proceso, esto hace que sea menos sensible a los cambios y estabilizar más rápido la densidad de la mezcla deseada.

En la sección de optimización se realiza un análisis de la robustez del controlador optimizante, el principal resultado establece que el controlador optimizante converge al valor óptimo siempre que la parte no lineal (o desviación de lo ideal) de la regla de mezclado y las variaciones en los elementos de alimentación tengan una contribución muy pequeña sobre las propiedades cualitativas de mezclado. Esto restringe el empleo del controlador optimizante

para sistemas de mezclado donde las variaciones del sistema son insignificantes. Además, se realiza un análisis de robustez para el controlador optimizante con redes neuronales, donde, se demuestra el se tiene una mayor robustez para el controlador optimizante con redes neuronales que para el que no las emplea dado que para el que emplea redes neuronales se tiene un valor de la no linealidad aún menor que para el caso donde no las emplea.

En la sección de modelación se realiza una comparación entre la identificación vía least square contra la identificación vía redes neuronales para el modelo completo y contra la identificación vía redes neuronales para la no linealidad teniéndose mejores resultados para la identificación vía redes neuronales para la no linealidad dado que en éste caso se tiene una generalización excelente ,i.e., las señales identificada y la deseada son muy aproximadas, y además, se ocupa un menor número de épocas para la generalización, comprobándose con esto que es recomendable el empleo de las redes neuronales para tener mejores resultados en la identificación para el proceso de mezclado de crudo.

En la sección de optimización se realiza una comparación entre el controlador optimizante para el mezclado de crudo con redes neuronales contra el controlador optimizante para el mezclado de crudo con datos reales teniéndose mejores resultados para el que emplea redes neuronales dado que las fracciones de flujo de los elementos de entrada fueron más estables y se tuvo un menor valor en la función de costo, comprobándose con esto que es recomendable el empleo de un controlador optimizante con redes neuronales para tener resultados más estables y un menor costo. Tomando en consideración la función de costo el mejor de todos es el controlador optimizante con redes neuronales para el modelo completo.

En el apéndice A se encuentra una Interfase Gráfica de Usuario (GUI) la cual contiene los resultados de las secciones de modelación y de optimización.

Como trabajo futuro se encuentran los siguientes puntos:

- Aunque aquí se presentó la representación de un proceso de mezclado con nodos mezcladores y el proceso en el caso de estudio, se requiere una generalización de un controlador optimizante para procesos de mezclado que contengan otros tipos de nodos y procesos, como columnas de separación, estabilizadores, separación de fases, etc.

- Se requiere una caracterización más completa del comportamiento dinámico del mezclado en tanques de almacenamiento y buque tanques.
- Se requiere una caracterización más completa del comportamiento dinámico de los tanques donde se realiza la operación de mezclado.



# Apéndice A

## Simulación con GUI

### A.1. Estructura del GUI

Las Interfases Gráficas de Usuario (GUI's) contienen varios componentes de interfaces de usuario que habilitan al software para comunicarse con un usuario final. Una de las tareas de implementar el GUI es controlar la forma en que el GUI responde a las acciones del usuario [5][23].

#### A.1.1. Entrada

##### **Botones push**

El botón push genera una acción cuando se presiona (i.e. un OK al botón puede cerrar una caja de diálogo). Cuando se hace click sobre un botón push, éste aparece presionado; cuando se pone en libertad el botón del ratón, la apariencia retorna a su estado no-presionado; éste ejecuta su callback (una acción) sobre un evento.

##### **Botones toggle**

El botón toggle genera una acción y indica un estado binario (i.e. encendido ó apagado). Cuando se hace click sobre un botón toggle, éste aparece presionado y permanece presionado

cuando se pone en libertad el botón del ratón, en éste estado se ejecuta su callback. Un subsecuente click al botón del ratón regresa al botón toggle a su estado no-presionado y de nuevo ejecuta su callback.

## **Menús popup**

Los menús popup se abren para desplegar una lista de elecciones (definidas mediante poner el nombre en una cadena de letras, o mejor conocida como string) cuando el usuario presiona el rectángulo que la define. Cuando no esta abierto, un menú popup despliega la elección actual, el cual está determinado por el índice contenido en el Value property. El primer artículo en la lista tiene un índice 1.

Los menú popup son útiles cuando se requiere proporcionar un número de elecciones mutuamente exclusivas, pero no quiere tomar una gran cantidad de espacio como es el caso de los radio buttons (botones radio).

## **Cajas listas**

La caja lista despliega una lista de artículos (definida usando una propiedad string) y habilita al usuario para seleccionar uno o más artículos. El valor de la propiedad contiene un índice dentro de la lista de cadenas que corresponde con el artículo seleccionado. Si el usuario selecciona múltiples artículos, entonces el valor es un vector de índices, entonces el primer artículo en la lista tiene un índice 1.

## **Textos editables**

El control del texto editable es un campo que habilita al usuario a meter o modificar cadenas de texto. Se emplea el texto editable cuando se quiere el texto como una entrada. La propiedad string (cadena de caracteres) contiene el texto introducido por el usuario.

## **A.1.2. Salida**

### **Ejes**

Los ejes habilitan el GUI para desplegar gráficas de salida (i.e. gráficas e imágenes). Los ejes no son objetos uicontrol, pero se pueden programar para habilitar la interacción del usuario con los ejes y los objetos gráficos desplegados en los ejes.

### **Figuras**

Las figuras son ventanas contenidas en el GUI para el diseño del editor de configuración.

### **Textos estáticos**

El texto estático despliega líneas de texto. El texto estático típicamente se emplea para etiquetar otros controles, guiando al usuario o indicando los valores de un deslizador. El usuario no puede cambiar el texto estático y no hay un camino para invocar la rutina callback asociada con éste.

## **A.1.3. Simulador de las redes neuronales y del controlador optimizante**

Existen muchos elementos los cuales permiten al usuario comunicarse con el software pero el que se emplea en este documento es el menú popup.

Los menús popup se abren para desplegar una lista de elecciones (definidas mediante poner el nombre en una cadena de letras, o mejor conocida como string) cuando el usuario presiona el rectángulo que la define. Cuando no está abierto, un menú popup despliega la elección actual, el cual está determinado por el índice contenido en el Value property. El primer artículo en la lista tiene un índice 1.

Los menús popup son útiles cuando se requiere proporcionar un número de elecciones mutuamente exclusivas, pero no quiere tomar una gran cantidad de espacio como es el caso de los radio buttons (botones radio).

En este desarrollo se presenta el GUI el cual muestra en un solo programa los resultados de la identificación con redes neuronales para el modelo completo y para la no linealidad, también muestra los resultados del controlador optimizante con redes neuronales para el modelo completo y para la no linealidad (los resultados citados se explican con detalle en los dos capítulos anteriores). Este programa consta de cuatro menús los cuales corresponden cada uno a los resultados citados y cada menú consta de la elección de cada uno de los tres mezcladores para el modelo distribuido y para el modelo agregado.

El programa de GUI tiene cuatro etiquetas las cuales tienen el siguiente significado:

RNMC = Red neuronal para el modelo completo

RNMC = Red neuronal para la no linealidad

COMC = Controlador optimizante para el modelo completo

RNMC = Controlador optimizante para la no linealidad

En la Figura A.1 se muestra el programa de GUI para la opción de la red neuronal para la no linealidad del mezclador 2.

Por último, en la Figura A.2 se muestra el programa de GUI para la opción del controlador optimizante para el modelo completo del modelo agregado.

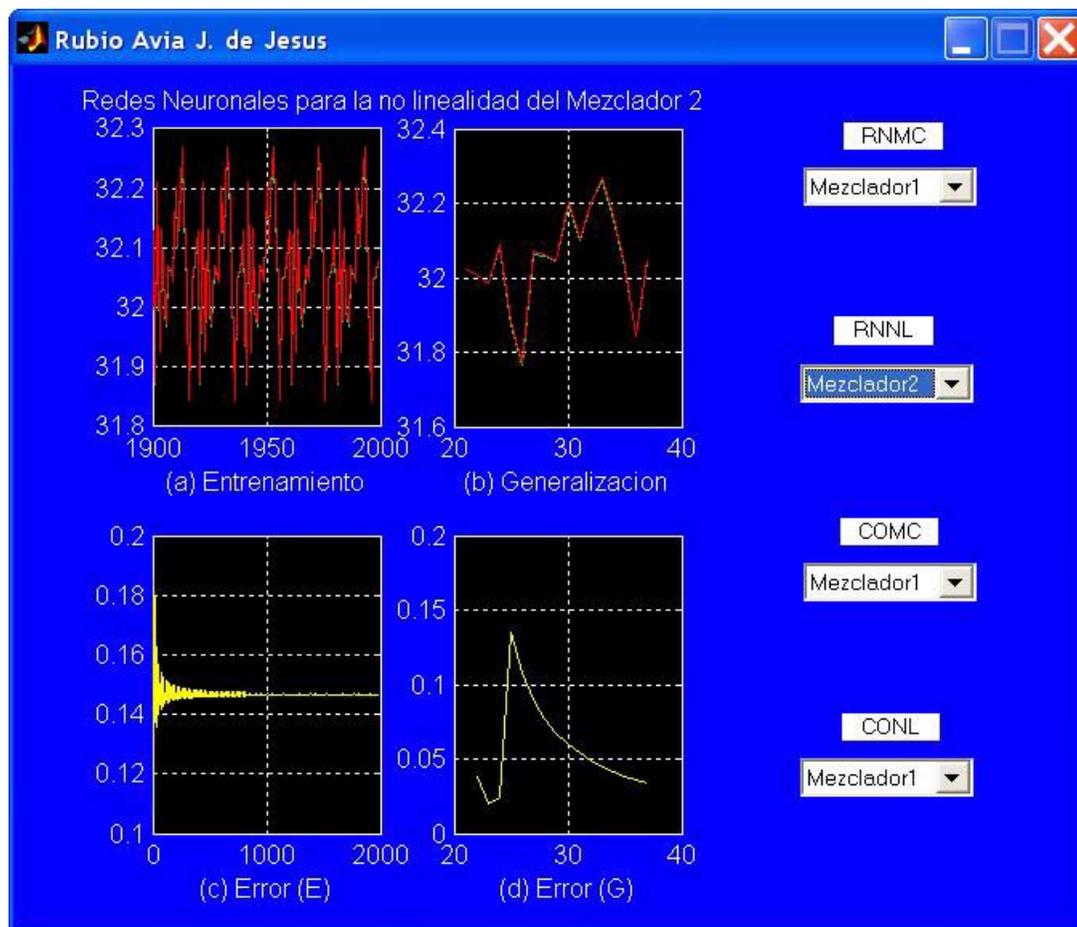


Figura A.1: GUI en la opción de la red neuronal para la no linealidad del mezclador 2

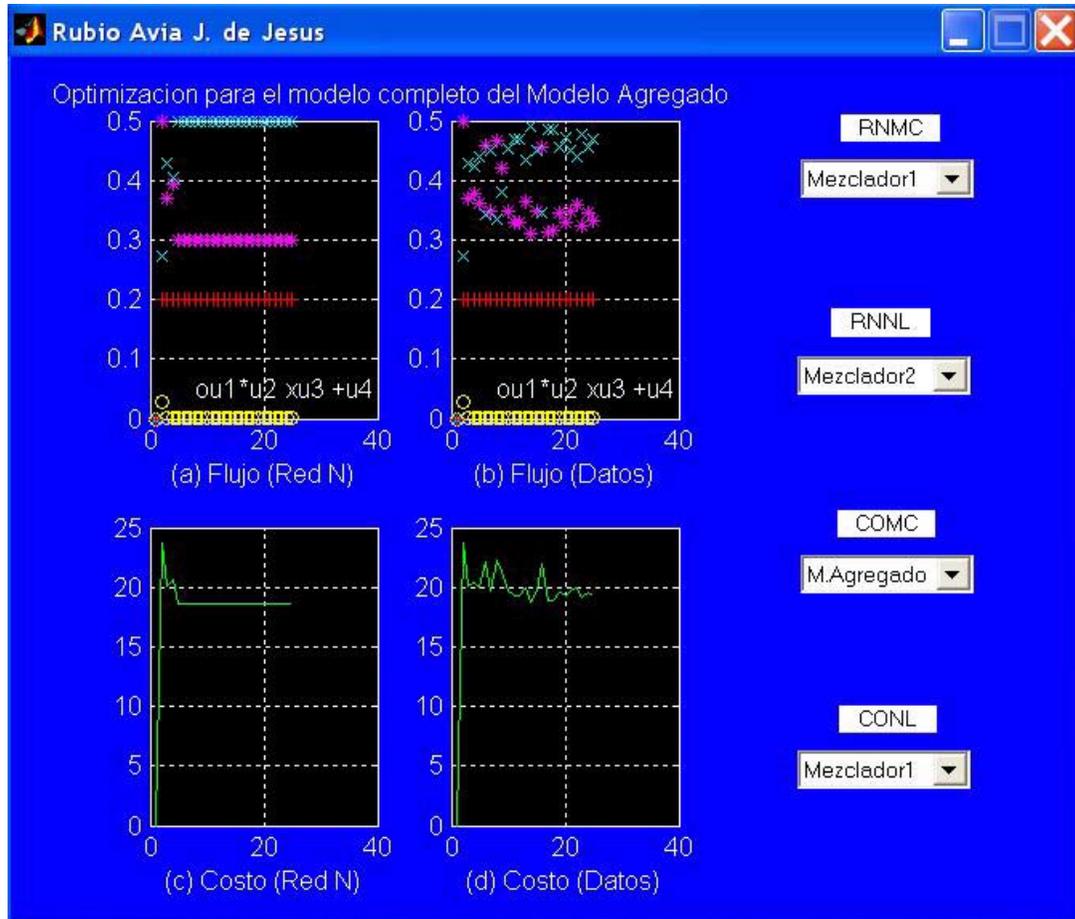


Figura A.2: GUI en la opción del controlador optimizante para el modelo completo del modelo agregado

# Apéndice B

## Método Simplex

La programación lineal es un método de optimización aplicable para solución de problemas en los cuales la función objetivo y las restricciones aparecen como funciones lineales de las variables de decisión. Las ecuaciones de restricción en el problema de programación lineal se pueden expresar como igualdades o como desigualdades [11][20].

### B.1. Forma estándar de un problema de programación lineal

El problema de programación lineal se enuncia de la siguiente forma:

- En forma escalar.

$$\text{Minimizar } f(x_1, x_2, \dots, x_n) = c_1x_1 + c_2x_2 + \dots + c_nx_n \quad (\text{B.1})$$

sujeto a las siguientes restricciones:

$$\left\{ \begin{array}{l} a_{11}x_1 + a_{12}x_2 + \dots + a_{1n}x_n = b_1 \\ a_{21}x_1 + a_{22}x_2 + \dots + a_{2n}x_n = b_2 \\ \vdots \\ a_{m1}x_1 + a_{m2}x_2 + \dots + a_{mn}x_n = b_m \end{array} \right\} \quad (\text{B.2})$$

y

$$\begin{cases} x_1 \geq 0 \\ x_2 \geq 0 \\ \vdots \\ x_n \geq 0 \end{cases} \quad (\text{B.3})$$

■ En forma matricial.

Minimize:

$$c^T X \quad (\text{B.4})$$

sujeto a las siguientes restricciones:

$$aX = b \quad (\text{B.5})$$

y

$$X \geq 0 \quad (\text{B.6})$$

donde:

$$X = \begin{pmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{pmatrix}, b = \begin{pmatrix} b_1 \\ b_2 \\ \vdots \\ b_n \end{pmatrix}, c = \begin{pmatrix} c_1 \\ c_2 \\ \vdots \\ c_n \end{pmatrix}, a = \begin{bmatrix} a_{11} & a_{12} & \cdots & a_{1n} \\ a_{21} & a_{22} & \cdots & a_{2n} \\ \vdots & & & \\ a_{m1} & a_{m2} & \cdots & a_{mn} \end{bmatrix}$$

Las características del problema de programación lineal se enuncian en la forma estándar como sigue:

1. La función objetivo es de tipo minimización.
2. Todas las restricciones son de tipo igualdad.
3. Todas las variables son no negativas.

El problema de programación lineal se puede poner en la forma estándar usando las siguientes transformaciones.

1. La maximización de una función  $f(x_1, x_2, \dots, x_n)$  es equivalente a la minimización del negativo de la misma función. Por ejemplo:

$$\begin{aligned} & \text{minimize} \\ & f = c_1x_1 + c_2x_2 + \dots + c_nx_n \end{aligned}$$

es equivalente a:

$$\begin{aligned} & \text{maximize} \\ & f' = -f = -c_1x_1 - c_2x_2 - \dots - c_nx_n \end{aligned}$$

2. Una variable puede no estar restringida en signo para algunos problemas. En tales casos, una variable no restringida (la cual puede ser positiva, negativa, o cero) se puede describir como una ecuación de diferencia de dos variables no negativas. Así, si  $x_j$  no está restringida en signo, esta se puede escribir como  $x_j = x_j' - x_j''$  donde:

$$x_j \geq 0 \quad \text{y} \quad x_j'' \geq 0$$

$x_j$  será negativa, cero o positiva dependiendo sobre si  $x_j''$  es mayor que, igual o menor que  $x_j'$ .

3. Si aparece una restricción en la forma de "menor que" de la siguiente forma:

$$a_{k1}x_1 + a_{k2}x_2 + \dots + a_{kn}x_n \leq b_k$$

Este se puede convertir en forma de igualdad adicionando una variable no negativa  $x_{n+1}$ .

$$a_{k1}x_1 + a_{k2}x_2 + \dots + a_{kn}x_n + x_{n+1} = b_k$$

Similarmente, si aparece una restricción en la forma de "mayor que" de la siguiente forma:

$$a_{k1}x_1 + a_{k2}x_2 + \dots + a_{kn}x_n \geq b_k$$

Este se puede convertir en forma de igualdad adicionando una variable no negativa  $x_{n+1}$ .

$$a_{k1}x_1 + a_{k2}x_2 + \dots + a_{kn}x_n - x_{n+1} = b_k$$

donde  $x_{n+1}$  es una variable no negativa conocida como variable excedente.

## B.2. Algoritmo simplex

El punto de inicio de un algoritmo simplex es siempre un espacio de ecuaciones, la cual incluye la función objetivo con restricciones de igualdad del problema en forma canónica. Así el objetivo del algoritmo simplex es encontrar el vector  $X \geq 0$  el cual minimice la función objetivo  $f(X)$  y satisfaga las ecuaciones:

$$\left\{ \begin{array}{l} 1 \cdot x_1 + 0 \cdot x_2 + \cdots + 0 \cdot x_m + a_{1,m+1}x_{m+1} + \cdots + a_{1n}x_n = b_1 \\ 0 \cdot x_1 + 1 \cdot x_2 + \cdots + 0 \cdot x_m + a_{2,m+1}x_{m+1} + \cdots + a_{2n}x_n = b_2 \\ \vdots \\ 0 \cdot x_1 + 0 \cdot x_2 + \cdots + 1 \cdot x_m + a_{m,m+1}x_{m+1} + \cdots + a_{mn}x_n = b_m \\ 0 \cdot x_1 + 0 \cdot x_2 + \cdots + 0 \cdot x_m - f + c_{m,m+1}x_{m+1} + \cdots + c_{mn}x_n = -f_0 \end{array} \right\} \quad (\text{B.7})$$

donde  $a_{ij}$ ,  $c_j$ ,  $b_j$  y  $f_0$  son constantes. Note que  $(-f)$  es tratada como una variable básica en la forma canónica de la ecuación (B.7). La ecuación a la que se puede reducir desde ecuación (B.7) es:

$$\left\{ \begin{array}{l} x_i = b_i, \quad i = 1, 2, \dots, m \\ f = f_0 \\ x_i = 0, \quad i = m + 1, m + 2, \dots, n \end{array} \right\} \quad (\text{B.8})$$

Si esta solución básica es también factible, los valores de  $x_i$ ,  $i = 1, 2, \dots, n$  son no negativos y por lo tanto:

$$b_i \geq 0, \quad i = 1, 2, \dots, m \quad (\text{B.9})$$

En la fase I del método simplex, la solución básica corresponde a obtener la forma canónica después de la introducción de variables artificiales y se obtendrá un problema artificial. La fase II del método simplex inicia con la solución factible básica del problema de programación lineal original. Por lo tanto en la forma canónica inicial en el inicio del algoritmo simplex siempre es una solución factible básica.

### B.3. Identificando un punto óptimo

**Teorema B.1** *Una solución factible básica es una solución óptima con el valor mínimo de la función objetivo de  $f_0$  si todos los coeficientes de costo  $c_j$ ,  $j = m + 1, m + 2, \dots, n$  en la ecuación (B.7) son no negativos.*

Para la ecuación (B.7) se puede escribir:

$$f_0 + \sum_{j=m+1}^n c_j x_j = f \quad (\text{B.10})$$

Si  $c_i > 0$  para  $i = m + 1, m + 2, \dots, n$ , entonces al incrementar el valor de cualquier  $x_i$  no puede decrementar el valor de la función objetivo  $f$ . Desde que no se pueden cambiar el valor de las variables básicas tal que causen que  $f$  decrezca, el valor de la función debería ser óptimo en  $f = f_0$ .

#### B.3.1. Mejorando una solución factible básica no óptima

Desde la última fila de la ecuación (B.7), se puede escribir la función objetivo como:

$$f = f_0 + \sum_{i=1}^m c_i x_i + \sum_{j=m+1}^n c_j x_j = f_0 \quad \text{debido a la solución dada en ecuación (B.8)} \quad (\text{B.11})$$

Si a lo menos un  $c_j < 0$ , el valor de  $f$  se puede reducir haciendo el correspondiente  $x_j > 0$ . al mismo tiempo, debido a la operación de pivote, una de las variables básicas corrientes se convierte en no básica y por lo tanto los valores de las nuevas variables básicas se ajustarán para obtener un nuevo valor de  $f < f_0$ . Si hay mas de un  $c_j < 0$ , el índice  $s$  de la variable no básica  $x_s$  el cual la hace básica es:

$$c_j = \text{mínimo} \quad (c_j < 0) \quad (\text{B.12})$$

Habiendo decidido sobre la variable  $x_s$  a convertirse en básica, se incrementa desde cero sosteniendo todas las otras variables como cero y observando el efecto sobre las variables

básicas corrientes.

$$\left\{ \begin{array}{l} x_1 = b_1 - a_{1s}x_s, \quad b_1 \geq 0 \\ x_2 = b_2 - a_{2s}x_s, \quad b_2 \geq 0 \\ \vdots \\ x_m = b_m - a_{ms}x_s, \quad b_m \geq 0 \end{array} \right\} \quad (\text{B.13})$$

$$f = f_0 + c_s x_s, \quad c_s < 0 \quad (\text{B.14})$$

Se puede ver que si todos los coeficientes  $a_{is} \leq 0$ ,  $i = 1, 2, \dots, m$ , entonces  $x_s$  se puede hacer infinitamente grande sin hacer ningún  $x_i < 0$ ,  $i = 1, 2, \dots, m$ . En tal caso el problema tiene una solución no acotada.

Por otro lado, si a lo menos un  $a_{is} > 0$ , el valor máximo que  $x_s$  puede tomar sin hacer  $x_i$  negativo es  $(b_i/a_{is})$ .

El más grande valor  $x_s^*$  que  $x_s$  puede tomar es:

$$x_s^* = \frac{b_r}{a_{rs}} = \text{mínimo} \quad (b_i/a_{is}) \quad (\text{B.15})$$

Sustituyendo el valor de  $x_s^*$  dado por ecuación (B.15) en ecuaciones (B.13) y (B.14) se tiene:

$$\left\{ \begin{array}{l} x_s = x_s^* \\ x_i = b_i - a_{is}x_s \geq 0, \quad i = 1, 2, \dots, m. \quad i \neq r \\ x_r = 0 \\ x_j = 0, \quad j = m+1, m+2, \dots, n. \quad j \neq s \end{array} \right\} \quad (\text{B.16})$$

$$f = f_0 + c_s x_s \leq f_0 \quad (\text{B.17})$$

Realizando la operación de pivote sobre el sistema de la ecuación (B.7) dará una nueva forma canónica desde la cual se deduce la ecuación (B.16). La ecuación (B.17) muestra que la solución factible básica corresponde a un valor de la función objetivo menor comparado con el valor de la ecuación (B.8). En el algoritmo simplex, este procedimiento es repetitivo hasta que el algoritmo encuentra cualquiera de lo siguiente: (i) una clase de soluciones factibles para las cuales  $f \rightarrow -\infty$  o (ii) una solución factible básica para la cual todos los  $c_i \geq 0$ ,  $i = 1, 2, \dots, n$ .

## B.4. Las dos fases del método simplex

El problema es encontrar valores no negativos para las variables  $x_1, x_2, \dots, x_n$  los cuales satisfagan las ecuaciones:

$$\left\{ \begin{array}{l} a_{11}x_1 + a_{12}x_2 + \cdots + a_{1n}x_n = b_1 \\ a_{21}x_1 + a_{22}x_2 + \cdots + a_{2n}x_n = b_2 \\ \vdots \\ a_{m1}x_1 + a_{m2}x_2 + \cdots + a_{mn}x_n = b_m \end{array} \right\} \quad (\text{B.18})$$

$$c_1x_1 + c_2x_2 \cdots + c_nx_n = f \quad (\text{B.19})$$

La fase I del método simplex usa el algoritmo simplex para encontrar si el problema de programación lineal tiene una solución factible. La fase II del método simplex, usa el algoritmo simplex para encontrar si el problema tiene una solución óptima acotada.

El método simplex está descrito por los siguientes pasos.

1. Arreglar el sistema original de la ecuación (B.18) tal que todos los términos  $b_i$  sean positivos o cero cambiando a donde sea necesario, el signo a ambos lados de la ecuación.
2. Introduce a este sistema básico un espacio de variables artificiales  $y_1, y_2, \dots, y_m$  donde cada  $y_i \geq 0$  y el sistema se convierte a:

$$\left\{ \begin{array}{l} a_{11}x_1 + a_{12}x_2 + \cdots + a_{1n}x_n + y_1 = b_1 \\ a_{21}x_1 + a_{22}x_2 + \cdots + a_{2n}x_n + y_2 = b_2 \\ \vdots \\ a_{m1}x_1 + a_{m2}x_2 + \cdots + a_{mn}x_n + y_m = b_m \end{array} \right\} \quad (\text{B.20})$$

$$b_i \geq 0$$

La función objetivo de la ecuación (B.19), se puede escribir de la siguiente manera:

$$c_1x_1 + c_2x_2 \cdots + c_nx_n - f = 0 \quad (\text{B.21})$$

3. **Fase I del método.** Define una cantidad  $w$  como la suma de las variables artificiales:

$$w = y_1 + y_2 + \dots + y_m \quad (\text{B.22})$$

y usar el algoritmo simplex para encontrar  $x_i \geq 0$  ( $i = 1, 2, \dots, n$ ) y  $y_i \geq 0$  ( $i = 1, 2, \dots, m$ ) el cual minimize  $w$  y satisfaga ecuaciones (B.20) y (B.21). Por consiguiente, considerando el arreglo:

$$\left\{ \begin{array}{l} a_{11}x_1 + a_{12}x_2 + \dots + a_{1n}x_n + y_1 = b_1 \\ a_{21}x_1 + a_{22}x_2 + \dots + a_{2n}x_n + y_2 = b_2 \\ \vdots \\ a_{m1}x_1 + a_{m2}x_2 + \dots + a_{mn}x_n + y_m = b_m \\ c_1x_1 + c_2x_2 + \dots + c_nx_n - f = 0 \\ y_1 + y_2 + \dots + y_m - w = 0 \end{array} \right. \quad (\text{B.23})$$

Este arreglo no está en la forma canónica, de cualquier manera, este se puede escribir en la forma canónica como sigue:

$$\left\{ \begin{array}{l} a_{11}x_1 + a_{12}x_2 + \dots + a_{1n}x_n + y_1 = b_1 \\ a_{21}x_1 + a_{22}x_2 + \dots + a_{2n}x_n + y_2 = b_2 \\ \vdots \\ a_{m1}x_1 + a_{m2}x_2 + \dots + a_{mn}x_n + y_m = b_m \\ c_1x_1 + c_2x_2 + \dots + c_nx_n - f = 0 \\ d_1x_1 + d_2x_2 + \dots + d_nx_n - w = -w_0 \end{array} \right. \quad (\text{B.24})$$

donde:

$$d_i = -(a_{1i} + a_{2i} + \dots + a_{mi}), \quad i = 1, 2, \dots, n \quad (\text{B.25})$$

y

$$-w_0 = -(b_1 + b_2 + \dots + b_m) \quad (\text{B.26})$$

La ecuación (B.24) proporciona la solución factible básica para iniciar la fase I.

4. A  $w$  se le llama la forma infinitesimal y tiene la propiedad que si se tiene como resultado de la fase I que el mínimo de  $w$  sea mayor que cero, entonces no existe solución factible

para el problema de programación lineal original de la ecuaciones (B.18) y (B.19) y se termina el proceso. Por otro lado, si el mínimo de  $w$  es igual a cero, entonces el arreglo resultante estará en la forma canónica e iniciará la fase II eliminando la ecuación de  $w$  así como las variables artificiales  $y_1, y_2, \dots, y_m$ .

5. **Fase II del método.** Aplicar el algoritmo simplex al sistema canónico ajustado de la fase I para obtener, si existe una solución finita, la cual optimice el valor de  $f$ .

Ahora, el problema de programación lineal enunciado en forma matricial mediante las ecuaciones (B.4)-(B.6) se puede reescribir de la siguiente manera:

$$\begin{aligned} \text{mín}(c, X), \quad X \in R^n \\ \text{sujeto a:} \\ (a^i, X) \leq b_i, \quad i = 1, \dots, m \end{aligned} \tag{B.27}$$

donde  $a^i$  es una fila de la matriz  $a$  y  $b_i$  es un elemento del vector  $b$ . Entonces, el siguiente teorema dice si la solución encontrada con el método simplex es la única [14].

**Teorema B.2** *Sea  $X^*$  una solución al problema de la ecuación (B.27) y sea  $I^* = \{i : (a^i, X^*) = b_i\}$  el espacio de las restricciones activas. Entonces, si entre los vectores  $a^i$ ,  $i \in I^*$ , no se pueden encontrar  $n$  vectores linealmente independientes, entonces  $X^*$  no es único. Mientras que, si entre los vectores  $a^i$ ,  $i \in I^*$ , si se pueden encontrar  $n$  vectores linealmente independientes y sus correspondientes multiplicadores de Lagrange  $y_i^*$  son positivos, entonces  $X^*$  es único.*



# Apéndice C

## Modelos matemáticos del proceso de mezclado de gasolina

Existen varias propiedades que son importantes en la caracterización de la gasolina de automóvil tales como son el número de octanos (ON), la presión de vapor Reid (RVP), puntos de destilación, de viscosidad y de equilibrio. En la mayoría de las propiedades de la mezcla en forma lineal y no-lineal, necesariamente usan modelos de mezclado complejos para predecir sus propiedades [16][25].

Hay varias características para describir el modelo de mezclado. En este tema la mayoría de esas características son para describir su exactitud.

### C.1. Número de octanos

El número de octanos indica las características anticknock de la gasolina o la habilidad de la gasolina para resistir a la detonación durante la combustión en la cámara de combustión. Los dos más comunes tipos de número de octanos para motores son el número de octanos de investigación (RON) y el número de octanos de motor (MON). Definidos por la Sociedad Americana para la examinación y el diseño de materiales ASTM D-908, el RON representa la ejecución del motor bajo condiciones de manejo en ciudad (aceleración de frecuencia)

mientras el MON (ASTM D-357) representa la ejecución del motor en autopista o carretera. Ya que el RON y el MON del mezclado de gasolina tienen una forma no-lineal, se necesitan modelos complejos para una predicción exacta del número de octanos mezclados.

### C.1.1. Método ideal del número de octanos mezclados

En éste método se usan el número de octanos ficticios BON's. Los BON's se mezclan linealmente sobre una base volumétrica para dar el número de octanos de la mezcla.

$$(ON)_{blend} = \sum_{i=1}^n u_i (BON)_i = \frac{1}{q_f} \sum_{i=1}^n q_i (BON)_i \quad (C.1)$$

donde:  $u_i$  es la fracción volumen del componente  $i$ ,  $(ON)_{blend}$  es el número de octanos (RON o MON) de la mezcla,  $(BON)_i$  es el número de octanos (RON o MON) del componente  $i$ ,  $q_i$  es la velocidad de flujo del componente  $i$ ,  $n$  es el número de componentes en la mezcla.  $q_f$  satisface el balance de masa:

$$q_f = \sum_{i=1}^n q_i \quad u_i = \frac{q_i}{q_f}$$

El número de octanos mezclados se obtienen mediante un análisis de regresión sobre un espacio de datos dados por el usuario. Así, la aplicación se basa frecuentemente sobre la experiencia y el juicio del usuario. Las limitaciones del empleo de este método se han reconocido ampliamente por lo cual se ha tratado con métodos más factibles que no dependan demasiado del juicio del usuario.

### C.1.2. Método Ethil RT-70

El método Ethil es uno de los más viejos en la literatura. El mezclado no lineal se modela explícitamente como función de la sensibilidad de los componentes (RON-MON), en este se tienen que dar el contenido de oleafina y el contenido aromático de los componentes.

$$RON_{blend} = \bar{r} + a_1 (\bar{r}\bar{s} - \bar{r}\bar{s}) + a_2 (\bar{O}^2 - \bar{O}^2) + a_3 (\bar{A}^2 - \bar{A}^2) \quad (C.2)$$

$$MON_{blend} = \bar{m} + a_4(\overline{ms} - \bar{m}\bar{s}) + a_5(\overline{O^2} - \bar{O}^2) + a_6 \left[ \frac{(\overline{A^2} - \bar{A}^2)}{100} \right]^2 \quad (C.3)$$

donde:  $r$  es RON,  $m$  es MON,  $s$  es la sensibilidad (RON o MON),  $s = p - m$ ,  $O$  es el contenido de oleafina (% por volumen) de cada elemento de entrada,  $a_i (i = 1 \dots 6)$  es el coeficiente de correlación.

Las ecuaciones (C.2) y (C.3) contienen un total de 6 parámetros para la mezcla de RON y MON. Para estimar estos parámetros, se necesitan los siguientes datos:

- RON, MON, contenido de oleafina y contenido aromático de cada elemento de alimentación puro.
- RON y MON de cada mezcla.

Una de las ventajas de emplear el método Ethil es que se puede expandir fácilmente al incluir los efectos de otros factores (tales como el contenido de sulfuro) sobre la mezcla de octano. Estos términos se pueden incluir en forma similar a los contenidos aromático y de oleafina usando términos no-lineales adicionales. Las características de la mezcla pueden cambiar considerablemente con los cambios en el número de octanos y con las cantidades de otros elementos de la alimentación. Por lo tanto, como las propiedades de los elementos de entrada cambian con el tiempo debido a los cambios en las operaciones de los procesos anteriores, los parámetros del modelo de mezclado pueden cambiar también. En el modelo Ethil, los parámetros se pueden estimar en línea usando los datos de mezcla históricos. Así, el modelo se puede hacer adaptivo y la exactitud no decrece con el tiempo. De cualquier manera, este requiere del conocimiento de los contenidos aromático y de oleafina los cuales no son fáciles de obtener.

### C.1.3. Método Steward

Este método fue propuesto por Steward más o menos al mismo tiempo que se publicó el método Ethil y es similar al método Ethil en que se contribuye el mezclado no-lineal al

contenido de oleafina de los componentes. El modelo de la mezcla es:

$$RON_{blend} = \frac{\sum_{i=1}^n V_i D_i [RON_i + \tilde{c}(O_i - O_{blend})]}{\sum_{i=1}^n V_i D_i} \quad (C.4)$$

$$MON_{blend} = \frac{\sum_{i=1}^n V_i D_i [RON_i + \tilde{c}(O_i - O_{blend})]}{\sum_{i=1}^n V_i D_i} \quad (C.5)$$

donde  $V_i$  es el volumen del componente  $i$  en la mezcla,  $D_i = \frac{\tilde{a}(O_i - O_{blend})}{1 - \exp[\tilde{a}(O_i - O_{blend})]}$ ,  $O_{blend}$  es el porcentaje de oleafina concentrado de la mezcla y es el promedio volumétrico del contenido de oleafina del componente, y  $\tilde{c}$  y  $\tilde{a}$  son constantes.

Steward ha determinado las constantes  $\tilde{c}$  y  $\tilde{a}$  usando un análisis least-square sobre 102 mezclas; el valor de  $\tilde{a}$  se determinó como 0.01414 para RON y 0.01994 para MON, el valor de  $\tilde{c}$  se determinó como 0.130 para RON y 0.097 para MON. DE cualquier manera, solamente 10 de las mezclas empleadas fueron multicomponente, las demás fueron mezclas binarias.

#### C.1.4. Método de Iteración

Este método está basado sobre el modelo dos-factores donde el efecto promedio se atribuye al efecto principal de cada uno de los dos factores y el término de iteración captura la no-linealidad.

El término de iteración informa el efecto de uno de los factores sobre el otro y se puede determinar mediante ciertos experimentos. Similarmente, en el método de iteración, la no-linealidad de la mezcla de octano se atribuye a la iteración dos-factores entre los componentes en la mezcla y se acumula mediante la adición y el término de iteración del promedio volumétrico del número de octanos. Se considera solamente la iteración dos-caminos entre pares de componentes, las de tres-caminos y términos de iteración más altos generalmente

se ignoran. Para un sistema de  $n$  componentes, el número de octanos (RON o MON) de la mezcla se puede calcular como:

$$(ON)_{blend} = \sum_{i=1}^n [u_i(ON)_i] + \sum_{i=1}^n \sum_{k=i+1}^n (u_i u_k I_{i,k}) \quad (C.6)$$

donde  $I_{i,k}$  es el coeficiente de iteración entre componentes  $i$  y  $k$ .

El coeficiente de iteración,  $I_{i,k}$ , entre un par de componentes se obtiene usando el número de octanos de los componentes puros y que para una mezcla 50:50 está dado como sigue:

$$I_{i,k} = 4(ON)_{i,k} - 2[(ON)_i + (ON)_k] \quad (C.7)$$

donde  $(ON)_{i,k}$  es el número de octanos de la mezcla 50 : 50 de  $i$  y  $k$ .

$I_{i,k}$  depende principalmente de la temperatura y de la presión. La ecuación (C.7) se puede generalizar para tomar en cuenta el coeficiente tres-caminos, de cualquier manera, se pueden obtener una muy pequeña cantidad de datos para coeficiente para tres-caminos de materiales puros. Desde un punto de vista práctico, se espera que el modelo de coeficiente dos-caminos retenga las principales iteraciones no-lineales de las mezclas homogéneas.

Para un sistema de  $n$  componentes, el número de coeficientes de iteración dos-caminos son  $\frac{n(n-1)}{2}$  para el RON o el MON, dando un total de  $n(n-1)$  parámetros por determinar. Los datos experimentales que se requieren para estimar estos parámetros son:

- RON y MON para cada componente puro,
- RON y MON para la mezcla del 50:50 de las combinaciones de todos los componentes.

Así, para el MON o el RON de un sistema de  $n$  componentes, el número total de combustible examinado son  $n$  componentes puros mas  $\frac{n(n-1)}{2}$  mezclas 50:50 que dan un total de  $\frac{n(n+1)}{2}$ . El número total de exámenes requeridos para el NOM y el RON son  $n(n+1)$ .

Desde que las características del octano mezclado puede cambiar con el nivel de octanos y de otras propiedades, el coeficiente de iteración se puede actualizar si ocurren cambios importantes en el octano alimentado. Así, la exactitud predecida del modelo de Iteración puede decrecer con el tiempo tal como las calidades que los elementos de entrada cambian

debido a procesos anteriores o a la variación en la cantidad de crudo. De cualquier manera, a diferencia del método Ethil, el modelo de Iteración no se puede hacer fácilmente adaptivo desde que los coeficientes de iteración  $(I_{i,k})$  no son coeficientes de correlación ya que se determinan experimentalmente. En lugar de determinar los coeficientes de iteración desde mezclas binarias, los coeficientes de iteración se pueden obtener usando un análisis de regresión sobre los datos de la mezcla (los datos de las calidades de todos los elementos de entrada así como la calidad de la mezcla). Los coeficientes de iteración, entonces se podrían actualizar usando datos de mezcla históricos y hacerlo adaptivo. Por ejemplo, 7 componentes de mezcla podría requerir de la determinación de 42 coeficientes de iteración mientras que el método Ethil requiere solamente 6 parámetros. Por lo tanto, se requiere una gran cantidad de datos de mezcla para actualizar los coeficientes de iteración. También, todos los parámetros del modelo de podrían evaluar como nuevos elementos de entrada para una nueva mezcla.

### C.1.5. Método en exceso

Este método trata con desviaciones de los ideal por la aplicación de un término "exceso". Estos valores excedentes se adicionan al promedio volumétrico del número de octanos para predecir el número de octanos de la mezcla (RON y MON) como:

$$(ON)_j = \sum_{i=1}^n u_{ij}(ON)_i + \sum_{i=1}^n u_{ij}(ON)_{ij}^E \quad (C.8)$$

donde:  $(ON)_j$  es el número de octanos de la mezcla  $j$ ,  $(ON)_{ij}^E$  es el número de octanos en exceso asociados con el componente  $i$  en la mezcla  $j$ , y  $u_{ij}$  es la fracción volumen del componente  $i$  en la mezcla  $j$ .

En este modelo lineal, el número de octanos en exceso,  $(ON)_{ij}^E$ , da una contribución de la no-linealidad en la mezcla  $j$  debida al componente  $i$ . Estos parámetros se determinan para cada mezcla por la preparación de una serie de restos-mezcla". Los restos-mezcla"se obtienen por el mezclado de muchas muestras de combustibles de componentes pero omitiendo un componente a la vez.

El número de valores de  $(ON)_{ij}^E$  por determinarse para cada grado de combustible (mezcla) es  $n$  y los datos experimentales que se necesitan para determinar estos parámetros son:

- número de octanos (RON y MON) para cada componente,
- número de octanos (RON y MON) de la mezcla,
- número de octanos (RON y MON) de los  $n$  restos-mezcla".

El número de combustibles que se necesitan para examinar son  $(n + d(n + 1))$ . Así, se necesitarían un total de  $2(n + d(n + 1))$  exámenes para el MON y RON combinados, donde  $d$  es el número de grados de gasolina.

### C.1.6. Método Zahed

Este método correlaciona el número de octanos de los elementos de entrada para predecir el número de octanos de la mezcla. Los parámetros en esta ecuación se estiman usando un análisis de regresión sobre los datos experimentales. El modelo de la mezcla para el RON es:

$$(RON)_{blend} = M_0 + \sum_{i=1}^n M_i [u_i(RO\!N)_i^l]^{k_3} \quad (\text{C.9})$$

Se puede emplear un método análogo para el MON:

$$(MON)_{blend} = Q_0 + \sum_{i=1}^n Q_i [u_i(MON)_i^l]^{k_4} \quad (\text{C.10})$$

donde  $M_0$ ,  $Q_0$ ,  $M_i$ ,  $Q_i$ ,  $k_3$  y  $k_4$  y  $l$  son constantes, y  $n$  es el número de elementos de alimentación en la mezcla.

Así, los parámetros que se necesitan determinar son  $M_0$ ,  $M_1, \dots, M_n$  y  $k_3$  para RON y  $Q_0$ ,  $Q_1, \dots, Q_n$  y  $k_4$  para MON. Estos datos que se necesitan para estimar estos parámetros son RON y MON de cada mezcla y de los  $n$  componentes.

Como con otros modelos, la exactitud de este método está limitada a un espacio de datos usados en la estimación de los parámetros desde las características de la mezcla y por los

tanto los parámetros del modelo pueden cambiar debido a los cambios en las calidades de los elementos de alimentación en el tiempo. Los parámetros del modelo se pueden actualizar en línea usando datos de mezcla históricos.

## C.2. Presión de Vapor Reid

La Presión de Vapor Reid (RVP), definida por la Sociedad Americana para el diseño y examinación de materiales ASTM D-323-56, da una indicación de la volatilidad de la mezcla de gasolina y es aproximadamente la presión vapor de la gasolina a  $100^{\circ}F(38^{\circ}C)$ . La RVP de una mezcla de gasolina afecta la ejecución de la gasolina en términos de su facilidad en el encendido, el calentamiento de motor, y la velocidad de aceleración. RVP es también importante debido a las especificaciones máximas sobre el límite RVP en la cantidad de n-butano, una fuente relativamente barata de octano, que se puede adicionar a la mezcla.

### C.2.1. Método de Iteración

El método de iteración también se ha aplicado al RVP de la mezcla por Morris. El modelo de la mezcla es de la misma forma que el expuesto en la ecuaciones (C.6) y (C.7) exceptuando que se mezcla el RVP en lugar del ON (número de octanos). Aquí también, se consideran solo las iteraciones entre pares de componentes. Las iteraciones para mayor número de elementos se ignoran. El modelo de iteración es:

$$(RVP)_{blend} = \sum_{i=1}^n [u_i(RVP)_i] + \sum_{i=1}^n \sum_{k=i+1}^n \left( u_i u_k \hat{I}_{i,k} \right)$$

$$\hat{I}_{i,k} = 4(RVP)_{i,k} - 2[(RVP)_i + (RVP)_k]$$

donde:  $n$  es el número de elementos de alimentación,  $\hat{I}_{i,k}$  es el coeficiente de iteración entre el elemento de alimentación  $i$  y  $k$ , y  $(RVP)_{i,k}$  es el RVP de una mezcla 50:50 de  $i$  y  $k$ .

El número de coeficientes de iteración requerido podría ser  $\frac{n(n-1)}{2}$ .

### C.2.2. Método de mezcla indexada

Un método empírico fácil de emplear desarrollado por Chevron REsearch Company es el método de mezcla indexada. En este alcance, se predice el RVP de la mezcla usando los índices de Presión de Vapor Reid de la mezcla (RVPBI) los cuales se mezclan linealmente. Los RVPBI's están dados por Gary y Handwerk y son de la siguiente forma:

$$RVPBI = (RVP)^{1,25} \quad (C.11)$$

$$(RVPBI)_{blend} = \sum_{i=1}^n (RVPBI)_i \quad (C.12)$$

## C.3. Destilación ASTM

Las gasolinas, desde que ellas son mezclas de Hidrocarburos con diferentes puntos de burbuja, evaporización sobre rangos de temperatura. Por ejemplo, el Nafta virgen ligero se evapora al rededor de una temperatura de  $30^{\circ}C$  a  $90^{\circ}C$ . Las características de volatilidad de la gasolina están medidas por la destilación ASTM D-86 el cual es un laboratorio estandarizado de destilación en batch la cual es llevada hacia afuera a la presión atmosférica sin una partición. Este da la cantidad (porciento volumen) destilado a una temperatura dada. Varias temperaturas a lo largo de la curva de referencia se emplean como puntos de referencia en la comparación de las propiedades de volatilidad de la gasolina. Equivalentemente, las propiedades de destilación se pueden especificar como las temperaturas a porciento dado de evaporización. Algunos de los métodos para predecir los puntos de destilación de la mezcla son los siguientes.

### C.3.1. Método de Iteración

El método de iteración, descrito para octano mezclado, se puede emplear para predecir los puntos de destilación ASTM como un porciento evaporado a una temperatura dada. Otra vez, solamente se consideran las iteraciones de dos elementos entre los coeficientes de iteración entre todos los pares de componentes los cuales están determinados por cada punto

en la curva de destilación. Como en los casos de los octanos y el RVP, la determinación de los coeficientes de iteración requiere de la calidad de los componentes individuales así como de las mezclas 50:50 y estos se calculan como se describió para el caso de los octanos.

Este método se aplica solamente para puntos intermedios, no para puntos finales de la mezcla de gasolina.

### C.3.2. Modelos empíricos

Frecuentemente se emplean métodos empíricos en la predicción de los puntos de destilación ASTM de la mezcla debido a su fácil empleo. Stanley y Pingrey desarrollaron correlaciones para predecir los puntos finales de las curvas de destilación cuando se conocen solamente las composiciones de la mezcla y los componentes de destilación ASTM. Este método no proporciona estimados para los puntos intermedios en la curva de destilación.

Dhulesia ha reportado algunos resultados usando modelos de la forma:

$$\tilde{v} = \left\{ 1 - e^{\left(\frac{T^*}{\alpha_1}\right)^{\alpha_2}} \right\} 100 \quad (C.13)$$

$$T^* = \frac{T - T_i}{T_f - T_i}$$

donde:  $\alpha_1$  y  $\alpha_2$  son constantes,  $\tilde{v}$  es el porcentaje de volumen evaporado,  $T_f$  y  $T_i$  son los puntos en que hierve final e inicial, respectivamente, de la mezcla.

No se puede predecir la exactitud de los modelos empíricos.

# Bibliografía

- [1] Jesús Noé Campos Favela, *Aplicación de un sistema de control optimizante para mezclado de petróleo crudo*, Guadalajara, 2003.
- [2] Der-Min Chang, Chen-Ching Yu, Coordinated Control of Blending Systems, *IEEE Transactions on Control Systems Technology*, Vol 6, No 4, 495-506, 1998.
- [3] J. Fraser Forbes and Thomas E. Marlin, Model Accuracy for Economic Optimization Controllers: The Bias Update Case, *Industrial Engineering Chemistry Research*, 1919-1929, 1994.
- [4] The Foxboro Company, *I/A Series Application Solution on Crude Oil Blending*, MA, E. U. A., 1999.
- [5] Javier García, José Ignacio Rodríguez y Alfonso Brazález, *Aprenda Matlab 5.3 como si estuviera en primero*, Universidad Politécnica de Madrid, 2001.
- [6] Graham C. Goodwin and Kwai Sang Sin, *Adaptive filtering prediction and control*, Prentice Hall, Englewood Cliffs, NJ07632, 1984.
- [7] M. Guay, S. Kansal and J.F. Forbes, *Industrial Engineering Chemistry Research*, Vol 40, No 9, 2089-2101, 2001.
- [8] Simon Haykin, *Neuronal Networks*, Macmillan Pub.Co., Englewood Cliffs, NJ 07632, 1999.

- [9] José Ramón Hilera Gonzáles y Víctor José Martínez Hernando, *Redes Neuronales Artificiales*, Adison-Wesley Iberoamericana, E.U.A., 45-180, 1995.
- [10] L. Ljung, Lennart, *System Identification: theory for the user*, Prentice Hall, Englewood Cliffs, NJ07632, 1987.
- [11] David G. Luenverger, *Introduction to linear and nonlinear programing*, 1990.
- [12] José Alvarez Ramirez, América Morales y Rodolfo Suares, Robustness of a Class of Bias Update Controllers for Blendidg Systems, *Industrial Engineering Chemistry Research*, 4786-4793, 2002.
- [13] Marco Antonio Moreno Armendariz, *Control adaptable del sistema no-lineal TORA usando redes neuronales dinámicas*, Centro de INvestigación y de ESTudios AVanzados del IPN, México D.F., 1999.
- [14] Boris T. Polyak, *Introduction to Optimization*, Optimizatio Software, New York.
- [15] José de Jesús Rubio Avila, *Control aplicado al proceso de separación de hidrocarburos (aceite, gas, agua)*, ESIME Zacatenco, México D.F., 2002.
- [16] Aseema Singh, *Modeling and Model Updating in the Real-Time Optimization of Gasoline Blending*, University of Toronto, 15-34, 1997.
- [17] A. Sangh, J. F. Forbes, P. J. Vermeer, S. S. Woo, Model-based real-time optimization of automotive gasoline blending operations, *Journal of Process Control* 10, 43-58, 2000.
- [18] J. M. Smith y H. C. Van Ness, *Introduction to chemical engineering thermodynamics*, Third Edition, McGraw-Hill Book Company, 213-279.
- [19] Eronini Umez, *Dinámica de sistemas y control*, Thomson Learning, México, 613-619, 2001.
- [20] John Wiley y Sons, *Optimization, Theory and Applications*, SS. Rao, New York, 1-125, 1978.

- [21] Wei Wu, Adaptive-like control methodologies for a CSTR system with dynamic actuator constraints, *Journal of Process Control* 13, 525-537, 2003.
- [22] The MathWorks, *Excel Link for use with Matlab*, User 's Guide, Version 1.1.2, 1-3 a 1-4, 2-3 a 2-4, 1999.
- [23] The MathWorks, Matlab, The Language of Technical Computing, *Creating Grafical User Interfaces*, 2000.
- [24] Wen Yu, Nonlinear system identification using discrete-time recurrent neural networks with stable learning algorithms, *Information Sciences: An International Journal*, Vol.158, No.1, 131-147, 2004.
- [25] Wen Yu, América Morales, Gasoline blending system modeling via static and dynamic neural networks, *International Journal of Modeling and Simulation*, accepted for publication
- [26] Wen Yu, Passivity Analysis for Dynamic Multilayer Neuro Identifier, *IEEE Trans. Circuits and Systems, Part I*, Vol.50, No.1, 173-178, 2003.
- [27] Wen Yu y América Morales, Off-line optimization of crude oil blending with neural networks and bias update scheme, *International Journal of Computational Engineering Science*, 2004, submitted.
- [28] Wen Yu and America Morales, Multiple Model Neuro Adaptive Control with Application to Blending, *IASTED International Conference on Intelligent Systems and Control (ISC 2003)*, Salzburg, Austria, 96-102, 2003
- [29] Yale Zhang, Dayadeep Monder, J. Fraser Forbes, Results analysis for trust constrained real-time optimization, *Journal of Process Control* 11, 329-341, 2001.
- [30] Yale Zhang, Dayadeep Monder, J. Fraser Forbes, Real-time optimization under parametric uncertainty a probability constrained approach, *Journal of Process Control* 12, 376-389, 2002.