



UNIVERSIDAD NACIONAL AUTÓNOMA DE MÉXICO

FACULTAD DE ESTUDIOS SUPERIORES CUAUTITLAN

**ESTUDIO DEL USO DE UNA RED NEURONAL
ARTIFICIAL PARA HACER AJUSTE FUNCIONAL DE
LAS VARIABLES DE UNA TORRE DE DESTILACIÓN**

TESIS

Que para obtener el título de

INGENIERO QUÍMICO

P R E S E N T A

CRISPINA GARRIDO BECERRIL

ASESOR

DR. RICARDO PARAMONT HERNÁNDEZ GARCÍA

CUAUTITLÁN IZCALLI ,ESTADO DE MÉXICO,2018



Universidad Nacional
Autónoma de México

Dirección General de Bibliotecas de la UNAM

Biblioteca Central



UNAM – Dirección General de Bibliotecas
Tesis Digitales
Restricciones de uso

DERECHOS RESERVADOS ©
PROHIBIDA SU REPRODUCCIÓN TOTAL O PARCIAL

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

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



FACULTAD DE ESTUDIOS SUPERIORES CUAUTITLÁN
 UNIDAD DE ADMINISTRACIÓN ESCOLAR
 DEPARTAMENTO DE EXÁMENES PROFESIONALES

ASUNTO: VOTO APROBATORIO

M. en C. JORGE ALFREDO CUÉLLAR ORDAZ
 DIRECTOR DE LA FES CUAUTITLÁN
 PRESENTE

ATN: I.A. LAURA MARGARITA CORTAZAR FIGUEROA
 Jefa del Departamento de Exámenes Profesionales
 de la FES Cuautitlán.

Con base en el Reglamento General de Exámenes, y la Dirección de la Facultad, nos permitimos comunicar a usted que revisamos el: Trabajo de Tesis

Estudio del uso de una red neuronal artificial para hacer ajuste funcional de las variables de una torre de destilación.

Que presenta la pasante: Crispina Garrido Becerril
 Con número de cuenta: 098516502 para obtener el Título de la carrera: Ingeniería Química

Considerando que dicho trabajo reúne los requisitos necesarios para ser discutido en el EXAMEN PROFESIONAL correspondiente, otorgamos nuestro **VOTO APROBATORIO**.

ATENTAMENTE
"POR MI RAZA HABLARÁ EL ESPÍRITU"
 Cuautitlán Izcalli, Méx. a 21 de Abril de 2017.

PROFESORES QUE INTEGRAN EL JURADO

	NOMBRE	FIRMA
PRESIDENTE	Dr. Ricardo Paramont Hernández García	
VOCAL	M. en C. Gilberto Atilano Amaya Ventura	
SECRETARIO	M. en E. María Teresa Ylizaliturri Gómez Palacio	
1er. SUPLENTE	I.A. Dulce María Hernández Oliver	
2do. SUPLENTE	Dr. Martín Rogelio Cruz Díaz	

NOTA: los sinodales suplentes están obligados a presentarse el día y hora del Examen Profesional (art. 127).

LMCF/cga*

Agradecimientos

A Dios, por permitirme cumplir una de mis metas más anheladas, la experiencia de vida de la que he tenido un gran aprendizaje

A mi familia por su apoyo y paciencia

A mi hermosa hija, quien ha sido mi maestra amorosa, mi motivación para seguir adelante

Gracias a la Universidad Nacional Autónoma de México y profesores, a ellos les debo mi formación, desarrollo profesional y mi amor por la enseñanza

A mi asesor, el Dr. Ricardo Paramont por su paciencia y apoyo

Agradezco también a los maestros Teresa y Gilberto por todo su apoyo.

A todos y cada uno de mis seres queridos, amigos, y profesores por ser parte importante de éste gran logro, gracias, gracias, gracias.

Índice

Descripción del problema que se atacará	6
Objetivos.	8
Objetivo general	8
Objetivos particulares	8
Introducción	9
Capítulo 1. Especificación del problema. Cálculos de equilibrio vapor-líquido en una etapa	10
El Problema	10
Ubicando la zona donde coexisten las fases de vapor y líquida	11
Punto de rocío	11
Punto de burbuja.....	12
Cálculos de <i>flash</i> isotérmico.....	13
Cálculos de <i>flash</i> adiabático	15
Algoritmo 1.1 Cálculos de <i>flash</i> isotérmico.....	16
Algoritmo 1.2 Cálculos de <i>flash</i> adiabático en la etapa <i>i</i>	17
Algoritmo 1.3 Método de la bisección usado para calcular la temperatura de la etapa <i>i</i>	19
Capítulo 2. Simulación de la torre de destilación haciendo cálculos etapa por etapa	21
Cálculos de balance en la torre de destilación.....	21
Haciendo los cálculos etapa por etapa	22
Algoritmo 2.1 Cálculos de balance en la torre de destilación.....	24
Algoritmo 2.2 Balance en el rehervidor	26
Capítulo 3. Otra forma de simular la torre.....	28
Las ecuaciones y la estructura del método numérico usado	28
Algoritmo 3.1 Solución de un sistema de <i>n</i> ecuaciones.....	29
Las ecuaciones que se resuelven simultáneamente	30
Variables calculadas no obtenidas con el método numérico para resolver el sistema de ecuaciones no lineales simultáneas	31
Variables calculadas con el método numérico para resolver el sistema de ecuaciones no lineales simultáneas	33
Tabla 3.1 Variables cuyos valores son calculados por el método numérico.....	33
Tabla 3.2 Ecuaciones que son resueltas por el método numérico	34
El método numérico.....	34
Capítulo 4. Las redes multicapa de perceptrones usadas.....	37
La Arquitectura de la Red.....	38

El entrenamiento de una red neuronal para que funcione como ajustadora de funciones.....	39
Algoritmo 4.1. Propagación hacia atrás en línea	41
Capítulo 5. Desarrollo de los programas	43
1) Programa para hacer los cálculos etapa por etapa de una torre de destilación para una mezcla ideal de alcanos.....	43
2) Programa para hacer los cálculos en una torre de destilación resolviendo iterativamente las ecuaciones de todas las etapas usando el método de Newton con las derivadas estimadas numéricamente.....	43
3) Programa para simular una red multicapa de perceptrones.....	43
4) Programa para obtener las tablas para entrenar y probar la red neuronal.	44
5) Programa para obtener por interpolación los valores de las variables dependientes que no calcula la red neuronal.	46
Capítulo 6. Experimentación y resultados	48
La red MLP.....	48
Entrenamiento de la Red MLP	48
Tabla 6.1 Intervalos en los que se obtuvieron los valores aleatorios	48
Tabla 6.2 Cambio del error.....	49
Prueba de la red MLP entrenada	49
Estimando por interpolación los datos de las variables que no son estimadas por la red.....	60
Tabla 6.3a Valores de las variables que especifican la corriente de entrada a la torre.....	61
Tabla 6.3b Valores del punto inicial estimado para arrancar el método de Newton con derivadas estimadas numéricamente	61
Tabla 6.4a Valores de las variables que especifican la corriente de entrada a la torre (variables independientes) obtenidos al azar por el programa (4)	62
Tabla 6.4b Valores del punto inicial estimado para arrancar el método de Newton con las derivadas estimadas numéricamente	63
Usando el método de Newton con las derivadas estimadas numéricamente para calcular los valores de las variables de todas las etapas.....	66
Tabla 6.5 Resultados de las corridas del método de Newton con las derivadas	66
Tabla 6.6 Número de iteraciones promedio	69
Conclusiones	70
Trabajo Futuro.....	70
Apéndice.....	71
Bibliografía	73

Estudio del uso de una red neuronal artificial para hacer ajuste funcional de las variables de una torre de destilación

Descripción del problema que se atacará

Considérese el problema de hacer los balances de materia y energía en una torre de destilación con dos salidas, operando en estado estacionario, trabajando con mezclas de más de dos componentes y considerando que todas las etapas que la constituyen son etapas en equilibrio. Los datos son: temperatura, presión, composición de la mezcla que se debe separar, número total de etapas de la torre, el número de la etapa donde se alimenta, la relación de reflujo del destilado y la relación del flujo de vapor que sale del rehervidor y del líquido que entra a él. Se hace la suposición de que no hay caídas de presión a lo largo de la torre, y que la presión en ella es igual a la presión de la corriente de alimentación.

Una forma de atacar este problema es realizar los cálculos etapa por etapa. Para obtener la temperatura en cada una de ellas -consideradas como adiabáticas¹- se realizan cálculos del *flash* adiabático, que consisten en hacer cálculos iterativos del *flash* isotérmico, haciendo en cada iteración el balance de energía y manipulando -también en cada una de esas iteraciones- la temperatura hasta lograr que la suma de las entalpías de las corrientes de entrada a la etapa en cuestión sea igual a la suma de las entalpías de la corriente de salida de ella. La temperatura de una etapa de equilibrio adiabática debe estar entre la temperatura de burbuja y la de rocío de la mezcla de las corrientes que entran en ella, así que la temperatura inicial supuesta para la etapa se supone dentro de ese intervalo, lo que obliga, para cada etapa, a realizar los cálculos de esas dos temperaturas. Estos cálculos se deben repetir en todas las etapas de la torre tantas veces como sea necesario, hasta que los valores de los flujos internos líquidos y de vapor de la torre, así como las composiciones y temperaturas de éstos, sean lo suficientemente parecidos a los calculados en la iteración anterior.

Se puede concluir que el número de cálculos para hacer el balance en la torre etapa por etapa, es muy grande. La gran ventaja que se tiene, es que, como se realizan de manera causa-efecto (porque se hacen por etapas sucesivas), los cálculos convergen fuertemente hacia una solución.

Supóngase que el problema no se aborda haciendo los cálculos etapa por etapa, sino que se plantea como el problema de resolver un solo sistema de ecuaciones algebraicas no lineales con términos trascendentes (aparecen variables como exponentes del número «e» -base de los logaritmos naturales- en algunas de ellas) con algún método numérico iterativo (por ejemplo, una aproximación al de Newton generalizado). En ese único sistema de ecuaciones

¹ Es decir, se supone que para las etapas de equilibrio distintas al rehervidor, no existe transferencia de calor entre la etapa y el exterior de la torre. En el rehervidor se hace un cálculo de *flash* isotérmico.

aparecerían las variables de todas las etapas; en cada iteración se irían obteniendo valores para todas las incógnitas, se repetirían las iteraciones hasta llegar a la solución. La ventaja de esta forma de atacar el problema es que se evitan los cálculos iterativos de flash adiabático en cada etapa, así como los cálculos (también iterativos) de las temperaturas de rocío y burbuja también en cada una de las etapas, lo que, en teoría, hará más rápidos los cálculos de balance de materia y energía en la torre. El método numérico requiere de un punto o suposición inicial (valores iniciales para cada una de las incógnitas: todos los flujos, temperaturas y composiciones) el que el converja hacia la solución o no depende fuertemente de ese punto inicial. Como no es fácil (más bien, es muy difícil) encontrar un punto inicial adecuado (muy cercano a la solución), el método numérico casi nunca llegará a la solución, lo que hace esta forma de abordar el problema inviable.

Para hacer viable la forma de solución descrita en el párrafo anterior, se requiere de algún método para encontrar un buen punto inicial para el método que resuelve el sistema de ecuaciones algebraicas con términos trascendentes. Una forma es entrenar una red *multicapa de perceptrones*, red MLP, para que “aprenda” la forma en la que se relacionan las variables de la corriente de alimentación (flujo, temperatura, presión, composiciones) y las relaciones de reflujo en el domo y de las corrientes de vapor y líquido que sale y entra al rehervidor, con los datos de las dos etapas en los extremos de la torre y los de algunas otras etapas intermedias de la misma. Así, una vez que la red MLP calcule los valores de las variables en esas etapas, se pueden estimar por interpolación los valores de las variables en las otras etapas, obteniéndose así los valores que constituyen el punto inicial para que arranque el método numérico que resuelve el sistema de ecuaciones. Por supuesto, es necesario realmente obtener ese punto inicial para varios casos y comprobar que el método numérico converja hacia la solución para cada uno de ellos.

Objetivos.

Objetivo general

Investigar si a partir de los datos obtenidos de las simulaciones llevadas a cabo por un programa que calcula etapa a etapa el funcionamiento de una torre de destilación operando en estado estacionario y suponiendo que cada etapa es una etapa al equilibrio, se puede entrenar una red multicapa de perceptrones, MLP (*Multi-layer Perceptron networks*), para que suministre los valores iniciales que requiere otro programa que hace los cálculos de la torre, utilizando una variación del método de Newton para resolver sistemas de ecuaciones no lineales que modela el funcionamiento de la torre.

Objetivos particulares

- 1.- Diseñar e implementar un programa que haga los cálculos etapa a etapa de una torre de destilación para el caso de una mezcla ideal de hidrocarburos normales de seis a nueve carbonos: n-hexano, n-heptano, n-octano y n-nonano, suponiendo que cada etapa es una etapa al equilibrio.
- 2.- Diseñar e implementar un programa para aplicar una variación del método de Newton generalizado para hacer los cálculos de una torre de destilación.
- 3.- Diseñar e implementar un programa que simule una red MLP para que aprenda las relaciones entre las variables dependientes e independientes de la torre.
- 4.- Diseñar e implementar un programa para que produzca de manera aleatoria los valores de las variables independientes de las tablas de entrenamiento y prueba de la red MLP.
- 5.- Diseñar un programa que interpole los valores de las variables dependientes obtenidas al correr el programa mencionado en (3), para obtener los valores de las variables dependientes que falten.
6. Hacer todas las corridas necesarias de los programas mencionados para obtener los resultados.

Introducción

Una forma de hacer los cálculos en una torre de destilación es el suponer que las corrientes de líquido y vapor que salen de cada plato están en equilibrio entre sí, es decir, considerar que una columna de destilación está constituida por una serie de etapas en equilibrio. Entonces, para hacer los cálculos de la torre (es decir calcular los flujos de las corrientes internas, las composiciones en ambas fases en cada etapa de la torre, el perfil de temperaturas, el calor suministrado al rehervidor, etc.), se puede proceder a llevar a cabo el cálculo del equilibrio vapor-líquido en cada una de las etapas de la torre. Primero es necesario suponer valores iniciales para las variables de la torre, y después ir corrigiendo esos valores mediante sucesivos cálculos en esas etapas hasta alcanzar la convergencia (encontrar los valores de las variables de manera iterativa).

Pensemos en el problema en el que los grados de libertad de la torre se especifican fijando las siguientes variables de la corriente de entrada: temperatura, presión fracción molar total de cada uno de los componentes y el flujo molar total; y estas otras variables: relación de reflujo en el domo, fracción vaporizada en el rehervidor². el número de etapas de equilibrio (incluyendo el rehervidor) y la temperatura del líquido que sale del condensador es la del punto de burbuja. Supóngase que la caída de presión es despreciable a lo largo de la torre y que la presión que hay en ella es igual a la de la corriente de entrada. Las variables que hay que calcular son: todos los flujos internos de líquido y vapor, los flujos de salida, sus composiciones, la temperatura en cada etapa y los calores del condensador y del rehervidor. Para alcanzar la convergencia, es necesario repetir varias veces los cálculos etapa por etapa.

Por otro lado, las redes multicapa de perceptrones, MLP, (*Multi-layer Perceptron networks*), pueden ser usadas como ajustadoras de funciones³ y se ha reportado que funcionan bien para ajustar múltiples funciones que dependen de un mismo conjunto de variables independientes.

Con el objeto de minimizar el número de cálculos en una torre para obtener las variables dependientes, se puede pensar en hacer que una red MLP aprenda la relación entre las variables dependientes y las independientes, de manera tal que cuando se presente un nuevo problema estime *de manera lo suficientemente aproximada* los valores de las variables dependientes en algunas etapas, y que por interpolación se obtengan los valores aproximados de las otras. El término *de manera lo suficientemente aproximada* se refiere a que los valores obtenidos de la manera que se acaba de describir, sirvan como punto de arranque para que un método iterativo que requiera menos cálculos en cada iteración que el método etapa a etapa, como es el método de Newton, para resolver sistemas de ecuaciones no lineales, converjan lo suficientemente rápido (es decir con pocas iteraciones) de manera tal que el obtener el punto de arranque y usar el método de Newton sea más económico en cálculos (y por tanto en tiempo de ejecución) que usar el método etapa a etapa.

² Es la relación entre vapor producido en el rehervidor y el flujo de líquido que entra en él.

³ El otro uso que se les da es el de clasificadoras de elementos por los patrones que éstos presentan.

Capítulo 1. Especificación del problema. Cálculos de equilibrio vapor-líquido en una etapa

El Problema

El problema que se propone es el de separar por medio de una columna de destilación una mezcla de alcanos normales de seis a nueve carbonos, en un producto rico en los alcanos de bajo punto de ebullición (ligeros) y otro rico en alcanos de alto punto de ebullición (pesados). Como los alcanos no presentan grandes dipolos, y los de esta mezcla en particular no tienen moléculas de tamaños muy diferentes entre sí y además forman una serie homóloga (Elliott & Lira, 1999, p.294), supondremos que sus mezclas tanto gaseosas como líquidas se comportan como ideales. Todos ellos son condensables (es decir, son componentes *subcríticos* porque sus temperaturas críticas están por encima de la temperatura de la solución (Prausnitz, p.7)), por lo que supondremos que sus mezclas constituyen soluciones ideales, y en consecuencia calcularemos sus coeficientes de reparto, K , con la ecuación de Raoult:

$$K_i = \frac{P_i^{sat}}{P} \quad (1.1)$$

Donde

K_i coeficiente de reparto del componente i entre la fase gaseosa y líquida,
 P_i^{sat} es la presión de saturación del componente i ,
 P es la presión total del sistema.

La presión de saturación de cada componente se calculará con la ecuación de Antoine:

$$P_i^{sat} = \exp\left(A_i + \frac{B_i}{C_i + T}\right) \quad (1.2)$$

Donde

T temperatura del sistema en escala absoluta

A_i, B_i, C_i constantes del componente i para la ecuación de Antoine.

Substituyendo la ecuación (1.2) en la ecuación (1.1):

$$K_i(T, P) = \frac{1}{P} \exp\left(A_i + \frac{B_i}{C_i + T}\right) \quad (1.3)$$

Por otro lado, el coeficiente de reparto de un componente i , es la relación de las fracciones molares de ese componente en la fase gaseosa y líquida:

$$K_i = \frac{y_i}{x_i} \quad (1.4)$$

Donde

x_i fracción molar de la sustancia i en fase líquida,
 y_i fracción molar de la sustancia i en fase de vapor,

Ubicando la zona donde coexisten las fases de vapor y líquida

En la destilación, el objeto de evaporar parcialmente una mezcla de líquidos de diferentes temperaturas de ebullición, es el lograr su separación: los más volátiles serán los que en mayor medida se evaporen, de esa manera, el vapor será más rico en esos componentes, mientras que en el líquido tenderán a quedarse los que se evaporan menos, es decir, los menos volátiles. Reiterando, es necesario que los líquidos se evaporen parcialmente, esto implica que coexistan las fases de vapor y líquido. Entonces es necesario determinar, para una presión dada, en qué intervalo de temperaturas existen las dos fases (o, en el caso de un sistema isotérmico, para la temperatura dada, hay que determinar el intervalo de presiones en las que coexistirán esas dos fases). A una presión dada, el intervalo de temperaturas donde coexisten las fases de vapor y líquido está acotado por la temperatura de rocío y la temperatura de burbuja (o si lo que se fija es la temperatura, el intervalo de presiones donde coexisten las fases de vapor y líquido, está acotado por las presiones de rocío y burbuja). Es por eso que es conveniente calcular las temperaturas de burbuja y rocío, para conocer, a una presión dada, la menor y la mayor temperatura en las que debe encontrarse una etapa para que exista en ella el equilibrio vapor-líquido (o, si es la temperatura la que se fija. Calcular las presiones de rocío y burbuja).

Punto de rocío

El punto de rocío de una mezcla de vapores condensables, se presenta cuando a una presión y temperaturas dadas, se forma la primera gota de líquido. Consideremos el problema en el que se tiene como datos las fracciones molares totales de todos los componentes de la mezcla, así como la presión a la que se encuentra ésta, y se tiene como incógnita la temperatura del punto de rocío, es decir, la temperatura de rocío. Para esa primera gota de líquido se cumple que:

$$\sum_{i=1}^n x_i = 1 \quad (1.5)$$

Donde

n es el número de sustancias de la mezcla.

Combinando (1.3), (1.4) y (1.5):

$$\sum_{i=1}^n \frac{y_i}{K_i(P, T)} = P \sum_{i=1}^n \frac{y_i}{\exp\left(A_i + \frac{B_i}{C_i + T}\right)} = 1 \quad (1.6)$$

En el punto de rocío prácticamente la mezcla está en estado de vapor (la primera gota que se forma es microscópica y por tanto su masa es despreciable), por eso ahí se tiene que:

$$y_i = z_i \quad (1.7)$$

Donde

z_i fracción molar total del componente i , esto es, las moles del componente i en fase líquida y fase de vapor, divididas entre la masa molar total de esas dos fases: $z_i = \frac{n_{i,V} + n_{i,L}}{n_V + n_L}$, donde $n_{i,V}$ son las moles del componente i en fase de vapor, $n_{i,L}$ son las moles del componente i en fase líquida, n_V son las moles totales en fase de vapor, n_L son las moles totales en fase líquida.

Substituyendo (1.7) en (1.6) y haciendo notar que (1.7) se da en el punto de rocío:

$$P \sum_{i=1}^n \frac{z_i}{\exp\left(A_i + \frac{B_i}{C_i + T_r}\right)} = 1 \quad (1.8)$$

Donde

T_r es la temperatura en el punto de rocío.

La temperatura de rocío se calcula con la ecuación (1.8). Obsérvese que no se puede despejar cuando se tiene dos o más componentes, por lo que es necesario utilizar un método numérico, específicamente uno de los llamados *buscadores de ceros*, para calcular su valor.

Si se aborda el caso en el que se tiene como dato la temperatura y lo que se desconoce es la presión de rocío, se puede despejar de la ecuación (1.8):

$$P_r = \frac{1}{\sum_{i=1}^n \frac{z_i}{\exp\left(A_i + \frac{B_i}{C_i + T}\right)}} \quad (1.9)$$

Punto de burbuja

El punto de burbuja se presenta cuando en una mezcla de líquidos, a una presión y temperatura dadas, se forma la primera burbuja de vapor. Consideremos el problema en el que se tiene como datos las fracciones molares totales de todos los componentes de la mezcla

de líquidos, así como la presión a la que se encuentra ésta, y se tiene como incógnita la temperatura del punto de burbuja, es decir, la temperatura de burbuja. Para esa primera burbuja de vapor se cumple que:

$$\sum_{i=1}^n y_i = 1 \quad (1.10)$$

Combinando (1.3), (1.4) y (1.10):

$$\sum_{i=1}^n x_i K_i(P, T) = \frac{1}{P} \sum_{i=1}^n x_i \exp\left(A_i + \frac{B_i}{C_i + T}\right) = 1 \quad (1.11)$$

En el punto de burbuja. Prácticamente toda la mezcla está en estado líquido, por lo que se cumple que:

$$x_i = z_i \quad (1.12)$$

Substituyendo (1.12) en (1.11) y tomando en cuenta que es el punto de burbuja se obtiene:

$$\frac{1}{P} \sum_{i=1}^n z_i \exp\left(A_i + \frac{B_i}{C_i + T_b}\right) = 1 \quad (1.13)$$

De la ecuación (1.13) se calcula la temperatura del punto de burbuja. Como en el caso de la temperatura de rocío, la temperatura de burbuja tampoco se puede despejar de la ecuación cuando $n > 2$; una de las formas de obtenerla es usando un método numérico del tipo buscador de ceros.

Si se tiene el problema de calcular la presión de burbuja, P_b , para una temperatura T dada, se puede despejar de (1.13):

$$P_b = \sum_{i=1}^n x_i \exp\left(A_i + \frac{B_i}{C_i + T_b}\right) \quad (1.14)$$

Cálculos de *flash* isotérmico

Otra variable que es importante calcular cuando se tiene dos fases en equilibrio, es la fracción de la masa total que está en una fase y la que está en la otra, así como las composiciones de cada una de las fases. Para obtener la fracción de la masa que está en la fase de vapor para un sistema a temperatura y presión conocidas, se hacen los cálculos de *flash* isotérmico.

En la figura 1.1 se muestra un sistema de *separación instantánea* (denominada *flash* en inglés). Constituye una etapa en la que las corrientes de salida están constituidas por fases en equilibrio, es decir, el *flash* es una etapa de equilibrio.

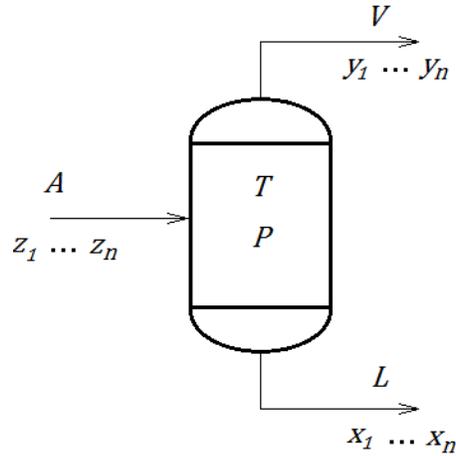


Figura 1.1 Flash isotérmico

Supóngase que se tiene como datos la temperatura, T , la presión, P , a las que opera el *flash*, el flujo molar de la corriente de entrada, A , y las fracciones molares totales de los componentes que la constituyen, z_i para $i = 1, 2, \dots, n-1$; las incógnitas son los flujos molares y las composiciones de las corrientes de salida: V, y_i, L, x_i para $i = 1, 2, \dots, n-1$. Para calcular las incógnitas se puede hacer los balances de materia:

$$A = L + V \quad (1.15)$$

$$Az_i = Lx_i + Vy_i, \quad \text{para } i = 1, 2, \dots, n-1 \quad (1.16)$$

Despejando L de (1.15):

$$L = A - V \quad (1.17)$$

Substituyendo (1.17) en (1.16):

$$Az_i = (A - V)x_i + Vy_i, \quad \text{para } i = 1, 2, \dots, n-1 \quad (1.18)$$

Dividiendo entre A :

$$z_i = (1 - \psi)x_i + \psi y_i, \quad \text{para } i = 1, 2, \dots, n-1 \quad (1.19)$$

Donde

ψ es la fracción de vapor, $\psi = \frac{V}{A}$.

De la ecuación (1.4):

$$y_i = K_i x_i \quad (1.20)$$

Sustituyendo (1.20) en (1.19):

$$z_i = (1 - \psi)x_i + \psi K_i x_i, \quad \text{para } i = 1, 2, \dots, n - 1 \quad (1.21)$$

Despejando x_i :

$$x_i = \frac{z_i}{[(1 - \psi) + \psi K_i]}, \quad \text{para } i = 1, 2, \dots, n - 1 \quad (1.22)$$

Sumando las n ecuaciones en (1.20):

$$\sum_{i=1}^n x_i = \sum_{i=1}^n \frac{z_i}{[(1 - \psi) + \psi K_i]} \quad (1.23)$$

Como la suma de todas las fracciones da 1:

$$\sum_{i=1}^n \frac{z_i}{[(1 - \psi) + \psi K_i]} = 1 \quad (1.24)$$

Los coeficientes de reparto se pueden calcular con la ecuación (1.3), la única incógnita en (1.23) es ψ , como no se puede despejar directamente esta incógnita, se puede aplicar un método numérico para búsqueda de ceros para calcularla.

Una vez que se obtiene el valor de la fracción de vapor, ψ , se puede calcular las fracciones de los componentes en la fase líquida con las ecuaciones (1.22), y posteriormente las fracciones de la fase de vapor con las ecuaciones (1.20).

Los cálculos del *flash* isotérmico se muestran en el algoritmo 1.1.

Cálculos de *flash* adiabático

En una etapa de una torre de destilación (que no sea el rehervidor), se puede considerar que no entra ni sale calor de ella, es decir, que se comporta de manera adiabática, para hacer los cálculos de *flash* hay que tomar en cuenta además de los balances de materia (como en el caso del *flash* isotérmico), el balance de energía.

Algoritmo 1.1 Cálculos de *flash* isotérmico.

Entradas: z_i , para $i = 1, \dots, n$, P , T .

Salidas: ψ , x_i , y_i , para $i = 1, \dots, n$.

Inicio	
1	Fijar z_i , para $i = 1, \dots, n$, P
2	Calcular las temperaturas de rocío y burbuja, T_b , T_r , con las ecuaciones (1.13) y (1.8) respectivamente, usando un método numérico del tipo buscador de ceros.
3	Fijar una temperatura, T , en el intervalo (T_b, T_r)
4	Calcular los coeficientes de reparto, K_i , con la ecuación (1.3) –se deben tener como datos las constantes de la ec. de Antoine para cada componente de la mezcla.
5	Calcular la fracción de la masa total que queda al equilibrio en fase de vapor, ψ , con la ecuación (1.24), usando un buscador de ceros.
6	Calcular las fracciones molares de los componentes en la fase líquida, x_i , con la ecuación (1.22).
7	Calcular las fracciones molares de los componentes en la fase de vapor, y_i , con la ecuación (1.20).
Fin	

En la figura 1.2 se esquematiza una etapa de equilibrio adiabática en una torre de destilación.

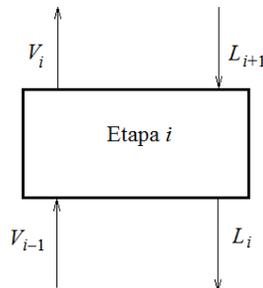


Figura 1.2 Una etapa de equilibrio adiabática en una torre de destilación.

En la etapa de equilibrio que se muestra en la figura 1.2, las corrientes V_i y L_i están en equilibrio.

El balance de energía se muestra en la ecuación (1.25)

$$\dot{Q} = V_{i-1}\underline{H}_{Y,i-1} + L_{i+1}\underline{H}_{L,i+1} - (L_i\underline{H}_{L,i} + V_i\underline{H}_{Y,i}) = 0 \quad (1.25)$$

Donde

\dot{Q} es el flujo de calor que entra/sale de la etapa, en este caso es igual a cero,
 $\underline{H}_{Y,i-1}$, $\underline{H}_{Y,i}$ entalpías molares de las mezclas en fase de vapor,
 $\underline{H}_{L,i}$, $\underline{H}_{L,i+1}$ entalpías molares de las mezclas en fase líquida,

V_{i-1}, V_i son los flujos molares en fase de vapor,

L_i, L_{i+1} son los flujos molares en fase líquida.

Subíndices:

i se refiere a un flujo que sale de la etapa i (de manera similar $i-1$ sale de la etapa $i-1$, $i+1$ sale de la etapa $i+1$)

Tomando en cuenta que $\dot{Q} = 0$, la ecuación (1.25) se puede reacomodar así:

$$V_{i-1}\underline{H}_{Y,i-1} + L_{i+1}\underline{H}_{L,i+1} = L_i\underline{H}_{L,i} + V_i\underline{H}_{Y,i} \quad (1.26)$$

El algoritmo para hacer los cálculos de *flash* adiabático en una etapa de equilibrio de la torre es el algoritmo 1.2.

Algoritmo 1.2 Cálculos de *flash* adiabático en la etapa i

Entradas: $P_i, V_{i-1}, L_{i+1}, \underline{H}_{Y,i-1}, \underline{H}_{L,i+1}, y_{i-1,j}, x_{i+1,j}$ para $j = 1, \dots, n$

Salidas: $T_i, V_i, L_i, \underline{H}_{Y,i}, \underline{H}_{L,i}, y_{i,j}, x_{i,j}$ para $j = 1, \dots, n$

inicio

1 Fijar *tolerancia*

2 **leer** $P_i, V_{i-1}, L_{i+1}, \underline{H}_{Y,i-1}, \underline{H}_{L,i+1}, y_{i-1,j}, x_{i+1,j}$ para $j = 1, \dots, n$

3 Calcular las fracciones molares totales de entrada:

$$z_{e,j} = \frac{V_{i-1}y_{i-1,j} + L_{i+1}x_{i+1,j}}{V_{i-1} + L_{i+1}}$$

4 Calcular la temperatura de burbuja en la etapa $i, T_{b,i}$, aplicando un buscador de ceros a la ecuación:

$$\frac{1}{P_i} \sum_{j=1}^n z_j \exp\left(A_j + \frac{B_j}{C_j + T_{b,j}}\right) - 1 = 0$$

5 Calcular la temperatura de rocío en la etapa $i, T_{r,i}$, aplicando un buscador de ceros a la ecuación:

$$P_i \sum_{j=1}^n \frac{z_j}{\exp\left(A_j + \frac{B_j}{C_j + T_{r,i}}\right)} - 1 = 0$$

6.a **hacer**

6.1 Suponer T_i tal que $T_b < T_i < T_r$

6.2 Calcular los coeficientes de reparto:

$$K_j(T_i, P_i) = \frac{1}{P_i} \exp\left(A_j + \frac{B_j}{C_j + T_i}\right)$$

para $j = 1, \dots, n$

6.3 Calcular la fracción de la masa de entrada que sale como vapor en la etapa i, ψ_i , aplicando un buscador de ceros a la ecuación

$$\sum_{j=1}^n \frac{z_j}{[(1 - \psi_i) + \psi_i K_j]} - 1 = 0$$

- 6.4 Calcular los flujos molares en las fases de vapor y líquido que salen de la etapa i :

$$V_i = \psi_i(V_{i-1} + L_{i+1})$$

$$L_i = (1 - \psi_i)(V_{i-1} + L_{i+1})$$

- 6.5 Calcular la composición del flujo en fase líquida que sale de la etapa i , L_i , con la ecuación:

$$x_j = \frac{z_j}{[(1 - \psi_i) + \psi_i K_j]}, \quad \text{para } j = 1, 2, \dots, n$$

- 6.6 Calcular la composición del flujo que sale de la etapa i en fase de vapor, V_i , con la ecuación

$$y_j = K_j x_j, \quad \text{para } j = 1, 2, \dots, n$$

- 6.7 Calcular las entalpías molares de las corrientes de salida:

$$\underline{H}_{V,i} = \sum_{j=1}^n \left(y_j \int_{T_{ref}}^{T_i} C_{p,V,j}(T) dT + \Delta \underline{H}_{ebull,j} \right)$$

$$\underline{H}_{L,i} = \sum_{j=1}^n \left(x_j \int_{T_{ref}}^{T_i} C_{p,L,j}(T) dT \right)$$

- 6.b **mientras** $|V_{i-1}\underline{H}_{Y,i-1} + L_{i+1}\underline{H}_{L,i+1} - (L_i\underline{H}_{L,i} + V_i\underline{H}_{Y,i})| > \textit{tolerancia}$

7 **regresar** $T_i, V_i, L_i, \underline{H}_{Y,i}, \underline{H}_{L,i}, y_{i,j}, x_{i,j}$ para $j = 1, \dots, n$

fin.

En el paso 6.1 en la descripción del algoritmo 1.2, se menciona que se supone una temperatura T_i (la temperatura de la etapa i), y que esa temperatura debe estar entre las temperaturas de rocío y de burbuja. Obsérvese que este paso está dentro de un ciclo, por lo que se deben hacer varias suposiciones a lo largo de la corrida. Los encargados de hacer estas suposiciones son métodos numéricos del tipo de los llamados buscadores de ceros, pero, para que arranquen, estos métodos necesitan por lo general uno o dos valores iniciales, dependiendo del tipo de método. Es decir, antes de entrar al ciclo, se debe suponer uno o dos valores de T_i , dependiendo del buscador de ceros que se use. Por ejemplo, si se usa un método que requiere un solo valor inicial, se puede usar el promedio aritmético de la temperatura de rocío y la de burbuja; si se usa un método que requiere dos valores iniciales, se puede usar como uno de los valores de T_i la temperatura de rocío, para el otro se puede usar la temperatura de burbuja.

El problema matemático que se debe resolver es:

$$f(T_i) = 0 \quad (1.27)$$

Donde

$$f(T_i) = V_{i-1}\underline{H}_{Y,i-1} + L_{i+1}\underline{H}_{L,i+1} - (L_i\underline{H}_{L,i} + V_i\underline{H}_{Y,i}) \quad (1.28)$$

La condición que se pide que cumpla el valor de T_i en la ecuación (1.27) es muy estricta, se puede hacer menos difícil de cumplir convirtiéndola en:

$$|f(T_i)| \leq \text{tolerancia} \quad (1.29)$$

Donde

tolerancia es un valor fijado por el usuario, es un número real positivo, debe ser lo suficientemente pequeño como para que el valor de T_i hallado se acerque con la precisión buscada al valor de T_i que hace que se cumpla la ecuación (1.27).

El método que se usó en este trabajo para resolver este problema fue el de la bisección. Este método necesita dos valores de la variable independiente: $T_i^{(0)}$ y $T_i^{(1)}$, también requiere que se cumpla que $\text{signo}(T_i^{(0)}) \neq \text{signo}(T_i^{(1)})$ y que la función sea continua en el intervalo $[T_i^{(0)}, T_i^{(1)}]$. Como sabemos que T_i se debe encontrar entre la temperatura de burbuja, T_b , y la temperatura de rocío, T_r , y la función $f(T_i)$ es continua en ese intervalo⁴, concluimos que esas dos temperaturas cumplen con las restricciones que se acaba de mencionar. El algoritmo 1.3 muestra cómo trabaja el método de la bisección para este caso.

Algoritmo 1.3 Método de la bisección usado para calcular la temperatura de la etapa i

Entradas: T_b , T_r , *tolerancia*

Salidas: T_i

inicio	
1	$T_i^{(0)} = T_{b,i}$
2	$y_0 = f(T_i^{(0)})$
3	$T_i^{(1)} = T_{r,i}$
4	$y_1 = f(T_i^{(1)})$
5a	hacer
5.1	$T_i^{(2)} = \frac{T_i^{(0)} + T_i^{(1)}}{2}$
5.2	$y_2 = f(T_i^{(2)})$
5.3	si $ y_2 \leq \text{tolerancia}$ entonces
5.3.1	regresar $T_i^{(2)}$
5.4	si no
5.4.1	si $\text{signo}(y_2) = \text{signo}(y_0)$ entonces
5.4.1.1	$x_0 = x_2$
5.4.1.2	$x_0 = y_2$

⁴ Y además se sabe que la $f(T_i)$ no tiene raíces múltiples. El que se cumpla esta última condición es importante, ya que, cuando existe multiplicidad en una raíz, dependiendo de cuántas veces se repita la raíz, puede ocurrir que la función no atravesase, sino que sólo “roce” (que pase *tangencial* a) el eje de las abscisas.

<p>5.4.2 si no 5.4.2.1 $x_1 = x_2$ 5.4.2.2 $y_1 = y_2$ 5b mientras $y_2 > \textit{tolerancia}$ fin</p>

En los pasos 4, 5 del algoritmo 1.2 se indica que se debe usar un buscador de ceros, el buscador usado fue la bisección. También en el paso 6.3 de ese mismo algoritmo se indica que se debe calcular el valor de ψ_i , usando un buscador de ceros, para este caso también fue el de la bisección, pero obsérvese que la ecuación a la que se aplica tiene una raíz en $\psi_i = 0$, pero la raíz que se busca está en el intervalo $(0, 1)$, es decir no incluye el 0. Para buscarla se usaban como puntos iniciales del método $\psi_i^{(0)} = 0 + \varepsilon$, $\psi_i^{(1)} = 1$; donde ε es un número real positivo mucho menor que 1 (por ejemplo, $\varepsilon = 1 \times 10^{-6}$).

Capítulo 2. Simulación de la torre de destilación haciendo cálculos etapa por etapa

Cálculos de balance en la torre de destilación

Una torre de destilación se puede visualizar como una serie de etapas en equilibrio, como se muestra en la figura 2.1.

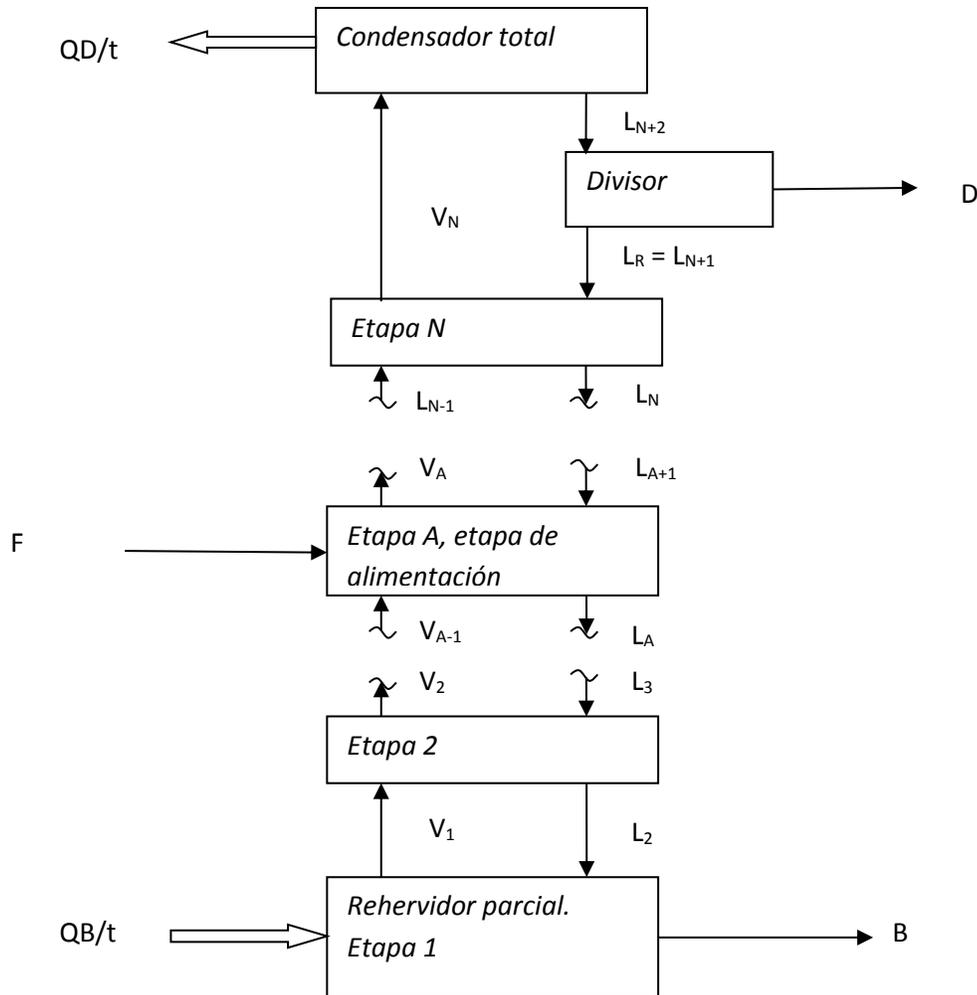


Figura 2.1 Una torre de destilación vista como etapas en equilibrio conectadas entre sí formando un arreglo vertical. Los flujos de vapor suben por fuerza boyante, los de líquido descenden por la fuerza de gravedad. (Esta figura es una adaptación de la figura 7.12 de Henley y Seader, 1990, p. 321)

En la figura 2.1 se muestra una torre de N etapas de equilibrio. Una forma de calcular la temperatura de cada etapa, sus flujos de entrada y salida y las composiciones de éstos, es haciendo los cálculos de *flash* para cada una de las etapas. De la etapa 2 a la etapa N , se puede considerar que se comportan como *flashes* adiabáticos, la etapa 1, el rehervidor, se puede considerar que se comporta como un *flash* isotérmico al que se alimenta el calor para evaporar parcialmente el flujo de líquido que recibe, L_2 .

Así para las etapas 2 a N se usa el algoritmo 1.2, con una modificación para la etapa A, la de alimentación. En esa etapa además de los flujos de líquido y vapor provenientes de otras etapas (o del condensador, como es el caso de la etapa N), hay una corriente que proviene del exterior: la corriente de alimentación, F , ésta se debe agregar a las corrientes de entrada al hacer los cálculos en esa etapa. La etapa 1 tiene una sola corriente de entrada y se comporta como un *flash* isotérmico, pero no se puede usar directamente el algoritmo 1.1 para hacer los cálculos en esta etapa, porque se desconoce la temperatura; en este capítulo se muestra la variación del algoritmo del *flash* isotérmico necesaria para hacer esos cálculos.

Haciendo los cálculos etapa por etapa

Una forma de proceder para hacer los cálculos en la torre es la siguiente: primero hay que suponer que todas las corrientes internas y de salida de la torre valen cero. Supongamos que la alimentación está en fase líquida o en fase líquida y de vapor. Se inicia la alimentación continua a la etapa de alimentación. La única entrada a ella es la corriente de alimentación, se hacen los cálculos de *flash* adiabático, con éstos se calcula las salidas de esa etapa. Se continúa con la etapa que está inmediatamente por debajo de la de alimentación, a esta etapa sólo entra el líquido que le cae de la etapa de alimentación, se repiten los cálculos de *flash* adiabático y se obtienen las salidas. Se repite el proceso para todas las etapas que siguen abajo hasta llegar al rehervidor. Uno de los datos fijados previamente es la relación de fracción vaporizada en el fondo de la torre, $R_b = V_1/B$, con el balance de materia se puede obtener qué parte del flujo de alimentación al rehervidor, L_2 , sale en forma de vapor en la corriente V_1 y cuál en forma de líquido en la corriente B : $L_2 = B + V_1$, dividiendo entre B : $L_2/B = 1 + V_1/B$, que o sea $L_2/B = 1 + R_b$, despejando B : $B = L_2/(1 + R_b)$, el flujo de vapor es entonces: $V_1 = L_2 - B$. De esta manera se tiene ya una primera estimación de la corriente de líquido que sale por el fondo de la torre, B , y del vapor que asciende por el interior de ésta, V_1 . Ahora se procede a repetir los cálculos del *flash* adiabático en la etapa por encima del rehervidor parcial. A esta etapa ahora entra no sólo el flujo de líquido que le cae de la etapa que está sobre ella, sino que también le entra el flujo de vapor que proviene del rehervidor, por eso los cálculos darán por resultado otros valores para la temperatura, los flujos y sus composiciones, distintos a los que se obtuvieron por primera vez. Se repite el proceso hasta llegar a la etapa de alimentación. Hasta aquí se han recalculado todas las etapas por debajo de la etapa de alimentación, es decir de la sección de agotamiento de la torre. Ahora se procede de manera similar con la etapa inmediatamente por arriba de la etapa de alimentación. A ella entra sólo el flujo de vapor que proviene de la etapa de alimentación. Se

hacen los cálculos de *flash* adiabático en esa etapa y se calculan sus flujos de salida (en este momento saldrá sólo vapor). Se repite el proceso con todas las etapas que están por arriba hasta llegar al condensador total. Previamente se ha fijado que la temperatura de salida del condensador es la temperatura de burbuja de la mezcla que sale de él; también se ha fijado la relación de reflujo en el domo: $R_d = L_R/D$. Con el balance de materia en el domo se puede calcular los flujos L_R y D : $V_N = L_R + D$, dividiendo entre D : $V_N/D = L_R/D + 1$, o sea $V_N/D = R_d + 1$, despejando D : $D = V_N/(R_d + 1)$, ya se puede calcular L_R : $L_R = V_N - D$. Ya se tiene una primera aproximación del flujo que sale por el domo de la torre así como del flujo de líquido que cae por el interior de ella. Ahora se procede a hacer los cálculos de *flash* adiabático en la etapa N . En esa etapa ya no sólo entra un flujo de vapor, también entra el flujo de líquido (el reflujo) que le cae del condensador que está arriba. Los cálculos darán diferentes resultados. Se procede de igual manera con todas las etapas hasta llegar a la etapa de alimentación. Se han hecho los cálculos en la sección por arriba del plato de alimentación, es decir en la sección de rectificación de la torre.

Se repiten los cálculos en la sección de agotamiento y luego otra vez en la de rectificación sucesivamente hasta que las temperaturas, flujos y composiciones ya no sufran cambios apreciables en las sucesivas iteraciones (es decir, hasta que se alcanza la convergencia).

En el algoritmo 2.1 se presentan los cálculos de balance en la torre de destilación descritos arriba. En este algoritmo

- F es flujo molar total de la corriente de entrada,
- ψ_F es la fracción de vaporizada de la corriente de entrada,
- P_F es la presión de la corriente de entrada,
- T_F es la temperatura de la corriente de entrada,
- z_F es la fracción molar total de la corriente de entrada,
- x_F es la fracción molar de la fase líquida de la corriente de entrada,
- y_F es la fracción molar de la fase de vapor de la corriente de entrada,
- N es el número total de etapas de la torre,
- A es el número de la etapa de alimentación; las etapas se cuentan de abajo hacia arriba empezando por el rehervidor (ver figura 1.3),
- R_d es la relación de reflujo en el domo de la torre de destilación:

$$R_d = \frac{L_R}{D}$$

- L_R es el flujo molar del líquido que sale del condensador y se recircula hacia la torre de destilación (también es llamado L_{N+1} , ver figura 2.1),
- D es el flujo molar que sale de la torre por el domo (producto de domo),
- R_b es la relación de vaporización en el fondo de la torre:

$$R_b = \frac{V_1}{B}$$

- V_1 es el flujo molar de vapor que sale del rehervidor (etapa 1 de equilibrio de la torre) y asciende hacia la etapa 2 de equilibrio,
- B es el flujo molar de líquido que sale del rehervidor por el fondo de la torre. Nótese que V_1 y B son fases en equilibrio,
- T_{cond} es la temperatura del condensador. Es igual a la temperatura de burbuja de la mezcla que se condensa en ese equipo. $T_{N+2} = T_{N+1} = T_{cond}$ (ver figura 1.3),
- \dot{Q}_D es el flujo de calor retirado de la mezcla por el condensador,
- \dot{Q}_B es el flujo de calor alimentado a la mezcla por el rehervidor,
- tol es una tolerancia; es el máximo error relativo que puede haber entre dos iteraciones de los cálculos de los flujos internos de vapor, líquido y de las temperaturas de las etapas de equilibrio de la torre.

Nótese que de la corriente que entra a la torre, la corriente de alimentación, se deben conocer la fracción vaporizada, ψ_F y las composiciones de sus fases líquida, $x_{F,i}$, y de vapor $y_{F,i}$. Para obtener esta información se debe hacer un cálculo de *flash* isotérmico (ver algoritmo 1.1) para la corriente de alimentación, dando como datos la presión, P_F , la temperatura, T_F , y la fracción molar total, z_F , de esa corriente. También se deben dar como datos la entalpía de la fase de vapor, $\underline{H}_{V,F}$, y la de la de la fase líquida, $\underline{H}_{L,F}$, de esa corriente; éstas se pueden estimar usando las capacidades caloríficas y la entalpía de ebullición:

$$\underline{H}_{V,F} = \sum_{i=1}^n \left(y_{i,F} \int_{T_{ref}}^{T_F} C_{p,V,i}(T) dT + \Delta \underline{H}_{ebull,i} \right) \quad , \text{ para } i = 1, \dots, n \quad (2.1)$$

$$\underline{H}_{L,F} = \sum_{i=1}^n \left(x_{i,F} \int_{T_{ref}}^{T_F} C_{p,L,i}(T) dT \right) \quad , \text{ para } i = 1, \dots, n \quad (2.2)$$

Donde

- $\underline{H}_{V,F}$ es la entalpía de la fase de vapor de la corriente de alimentación a la torre,
- $\underline{H}_{L,F}$, es la entalpía de la fase líquida de la corriente de alimentación a la torre,
- $\Delta \underline{H}_{ebull,i}$ es la entalpía de ebullición del componente i de la mezcla,
- $C_{p,V,i}$ es la capacidad calorífica a presión constante en fase de vapor del componente i ,
- $C_{p,L,i}$ es la capacidad calorífica a presión constante en fase líquida del componente i ,
- n es el número de componentes de la mezcla.

Algoritmo 2.1 Cálculos de balance en la torre de destilación

Entradas: $F, P_F, T_F, \psi_F, \underline{H}_{V,F}, \underline{H}_{L,F}; z_{i,F}, y_{i,F}, x_{i,F}$ para $i = 1, \dots, n; A, N, Rd, Rb, tol$

Salidas: V_i, T_i , para $i = 1, \dots, N; L_i$, para $i = 2, \dots, N+2, T_{cond}, D, B, \dot{Q}_D, \dot{Q}_B$

inicio

- 1 Pone en cero todas las corrientes internas de la torre:
 $V_i = 0$ para $i = 1, \dots, N$; $L_i = 0$, para $i = 2, \dots, N+2$
- 2 Se fija la presión en cada etapa de equilibrio igual a la presión de la corriente de alimentación: $P_i = P_F$, para $i = 1, \dots, N$
- 3 $t = 0$
- 4a **hacer**
 - 4.1 $t = t + 1$
 - 4.2 **seguir_iterando = falso**
 - 4.3 Hacer cálculos de *flash* adiabático en la etapa de alimentación, A, para obtener $V_A(t)$, $L_A(t)$, $x_{i,A}(t)$, $y_{i,A}(t)$, $T_A(t)$, $H_{V,A}(t)$, $H_{L,A}(t)$
 - 4.4 **para $i = A-1$ hasta 2 incremento -1**
 - 4.4.1 Hacer cálculos de *flash* adiabático en la etapa i para obtener $V_i(t)$, $L_i(t)$, $x_i(t)$, $y_i(t)$, $T_i(t)$, $H_{V,i}(t)$, $H_{L,i}(t)$
 - 4.5 Hacer cálculos del *flash* isotérmico en la etapa 1, el rehervidor, para obtener $T_1(t)$, $\dot{Q}_B(t)$, $V_1(t)$ y $B(t)$ ($L_1(t) = B(t)$)
 - 4.6 **para $i = 2$ hasta A-1**
 - 4.6.1 Hacer cálculos de *flash* adiabático en la etapa i para actualizar $V_i(t)$, $L_i(t)$, $x_i(t)$, $y_i(t)$, $T_i(t)$, $H_{V,i}(t)$, $H_{L,i}(t)$
 - 4.7 Hacer cálculos de *flash* adiabático en la etapa de alimentación, A, para actualizar $V_A(t)$, $L_A(t)$, $x_{i,A}(t)$, $y_{i,A}(t)$, $T_A(t)$, $H_{V,A}(t)$, $H_{L,A}(t)$
 - 4.8 **para $i = A+1$ hasta N**
 - 4.8.1 Hacer cálculos de *flash* adiabático en la etapa i para obtener $V_i(t)$, $L_i(t)$, $x_i(t)$, $y_i(t)$, $T_i(t)$, $H_{V,i}(t)$, $H_{L,i}(t)$
 - 4.9 Obtener la temperatura de burbuja de la mezcla que constituye la corriente L_{N+2} (que tiene el mismo flujo y composición que la corriente V_N), como se da como dato que esta corriente es líquido saturado, esa temperatura de burbuja será la suya: $T_{N+2} = T_{b,N+2}$
 - 4.10 **para $i = N$ hasta A+1 incremento -1**
 - 4.10.1 Hacer cálculos de *flash* adiabático en la etapa i para actualizar $V_i(t)$, $L_i(t)$, $x_i(t)$, $y_i(t)$, $T_i(t)$, $H_{V,i}(t)$, $H_{L,i}(t)$
 - 4.11 Hacer cálculos de *flash* adiabático en la etapa de alimentación, A, para volver a actualizar $V_A(t)$, $L_A(t)$, $x_{i,A}(t)$, $y_{i,A}(t)$, $T_A(t)$, $H_{V,A}(t)$, $H_{L,A}(t)$
 - 4.12 **si $t = 1$ entonces seguir_iterando = verdadero**
 - 4.13 **si no**
 - 4.13.1
$$e_{rel1,i} = \frac{|V_i(t) - V_i(t - 1)|}{\frac{1}{2}|V_i(t)| + |V_i(t - 1)|}$$
 - 4.13.2
$$e_{rel2,i} = \frac{|L_i(t) - L_i(t - 1)|}{\frac{1}{2}|L_i(t)| + |L_i(t - 1)|}$$
 - 4.13.3
$$e_{rel3,i} = \frac{|T_i(t) - T_i(t - 1)|}{\frac{1}{2}|T_i(t)| + |T_i(t - 1)|}$$

4.13.4 **si** ($e_{rel1,i} > tol$) \forall ($e_{rel2,i} > tol$) \forall ($e_{rel3,i} > tol$) para $i=1, \dots, N$
entonces seguir_iterando = verdadero

4b **mientras** seguir_iterando

5 $T_{cond} = T_{N+2}; T_{N+1} = T_{N+2}$

6 $\dot{Q}_D = L_{N+2} \underline{H}_{L,N+2} - V_N \underline{H}_{V,N+2}$

7 $\dot{Q}_B = V_1 \underline{H}_{V,1} - L_1 \underline{H}_{L,1} = V_1 \underline{H}_{V,1} - B \underline{H}_{B,1}$

fin

En los pasos 4.3, 4.7 y 4.11 del algoritmo 2.1 se indica que se deben hacer cálculos de *flash* adiabático en la etapa de alimentación. Esos cálculos de *flash* adiabático son como en cualquier otra etapa, con la diferencia de que debe tomarse en cuenta que se alimenta a esa etapa la corriente F , así la ecuación que aparece en el paso 3 del algoritmo 1.2, se debe cambiar por la ecuación (2.3).

$$z_{e,j} = \frac{V_{i-1} y_{i-1,j} + L_{i+1} x_{i+1,j} + F x_{A,j}}{V_{i-1} + L_{i+1} + F} \quad (2.3)$$

En el paso 4.5 del algoritmo 2.1 se indica que se debe hacer un cálculo de flash isotérmico en la etapa 1 (el rehervidor), para calcular el valor de la temperatura en esa etapa. Nótese que este cálculo difiere del mostrado en el algoritmo 1, ya que en éste último se fijan la temperatura y la presión para calcular la fracción vaporizada, es decir, no calcula la temperatura, sino que la pide como dato. El algoritmo 2.2 muestra cómo hacer el cálculo del *flash* isotérmico en la etapa 1.

Algoritmo 2.2 Balance en el rehervidor

Entradas: $P_1, L_2, R_b, x_{2,i}$ para $i = 1, \dots, n; \underline{H}_{L,2}$

Salidas: $T_1, V_1, \psi_1, B, \dot{Q}_B, x_{B,i}$ y $y_{1,i}$ para $i = 1, \dots, n$

inicio

1 $B = \frac{L_2}{1 + R_b}$

2 $V_1 = L_2 - B$

3 $\psi_1 = \frac{V_1}{L_2}$

4 Calcular la temperatura de rocío en la etapa 1, $T_{r,1}$, usando la ecuación (1.6) aplicada a la etapa 1:

$$P_1 \sum_{i=1}^n \frac{x_{2,i}}{\exp\left(A_i + \frac{B_i}{C_i + T_{r,1}}\right)} = 1$$

usando un buscador de ceros.

5 Calcular la temperatura de burbuja en la etapa 1, $T_{b,1}$, usando la ecuación (1.13) aplicada a la etapa 1:

$$\frac{1}{P_1} \sum_{i=1}^n x_{2,i} \exp\left(A_i + \frac{B_i}{C_i + T_{b,1}}\right) = 1$$

usando un buscador de ceros.

6a **hacer**

6.1 Suponer T_1 tal que $T_{b,1} < T_1 < T_{r,1}$

6.2 Calcular los coeficientes de reparto para cada sustancia en la etapa 1, $K_{1,i}$, usando la ecuación 1.3 aplicada a la etapa 1:

$$K_{1,i} = \frac{1}{P_1} \exp\left(A_i + \frac{B_i}{C_i + T_1}\right) \quad \text{para } i = 1, \dots, n.$$

6.3 **si** $\left(\sum_{i=1}^n \frac{x_{1,i}}{[(1-\psi_1) + \psi K_{1,i}]} - 1\right) \leq \text{tolerancia}$ **entonces**

6.3.1 seguir_iterando = **falso**

6.4 **si no**

6.4.1 Seguir_iterando = **verdadero**

6b **mientras** seguir_iterando

7 Calcular la composición de la corriente B usando la ecuación (1.22) para la etapa 1:

$$x_{B,i} = \frac{x_{2,i}}{[(1-\psi_1) + \psi_1 K_{1,i}]}, \quad \text{para } i = 1, \dots, n$$

8 Calcular la composición de la corriente V1 usando la ecuación (1.20) para la etapa 1:

$$y_{1,i} = K_{1,i} x_{1,i} \quad \text{para } i = 1, \dots, n$$

fin

Capítulo 3. Otra forma de simular la torre

En el capítulo anterior se vio cómo simular la torre haciendo los cálculos etapa por etapa. Otra forma de hacer los cálculos, es plantear todas las ecuaciones que relacionan las variables que se estudian de toda la torre, y resolverlas simultáneamente mediante algún método numérico. En este capítulo se plantearán primero el sistema de ecuaciones y después se verá el método numérico para resolverlas.

Las ecuaciones y la estructura del método numérico usado

Sea \mathcal{V} el conjunto de todas las variables en estudio, \mathcal{V}_i el subconjunto de las variables independientes, \mathcal{V}_d el de las variables dependientes (es decir, $\mathcal{V} = \mathcal{V}_i + \mathcal{V}_d$). Llámese $\mathbf{x} = (x_1, x_2, \dots, x_n)$ a los valores de las variables dependientes, que son los que se desea calcular.

De manera general, el problema consiste en resolver un sistema de ecuaciones, varias de ellas no lineales. Se puede enunciar así:

“Dado el sistema de ecuaciones:

$$\mathbf{f}(\mathbf{x}) = 0 \quad (3.1)$$

Donde

$\mathbf{x} \in \mathbb{R}^n$ son las variables dependientes cuyos valores se deben calcular,
 \mathbf{f} son las ecuaciones $f_i(\mathbf{x})$, para $i = 1, \dots, n$, que constituyen el sistema de ecuaciones.

Encontrar las componentes x_1, x_2, \dots, x_n del vector \mathbf{x} que satisfaga (3.1)”.

Para el problema que se debe resolver en este trabajo, algunas de las ecuaciones del sistema tienen la forma:

$$f(x_1, x_2, \dots, x_{n_1}, \chi) = 0 \quad (3.2)$$

Donde χ es una variable dependiente, es decir pertenece al conjunto \mathcal{V}_d , que se puede despejar fácilmente de la ecuación. Ordéñese las ecuaciones de manera tal que primero vayan las ecuaciones de la forma mostrada en (3.2):

$$f_1(x_1, x_2, \dots, x_{n_1}, \chi_1) = 0 \quad (3.2a)$$

$$f_2(x_1, x_2, \dots, x_{n_1}, \chi_2) = 0 \quad (3.2b)$$

.

.

$$f_{n_2}(x_1, x_2, \dots, x_{n_1}, \chi_{n_2}) = 0 \quad (3.2n_2)$$

Despejando χ_i , para $i = 1, \dots, n_2$, en la ecuación f_i correspondiente:

$$\chi_1 = g_1(x_1, x_2, \dots, x_{n_1}) \quad (3.3a)$$

$$\chi_2 = g_2(x_1, x_2, \dots, x_{n_1}) \quad (3.3b)$$

$$\chi_{n_2} = g_{n_2}(x_1, x_2, \dots, x_{n_1}) \quad (3.3m_2)$$

El vector de variables dependientes se puede expresar entonces:

$$\mathbf{x} = (\mathbf{x}^*, \boldsymbol{\chi}) \quad (3.4)$$

Donde

$$\mathbf{x}^* \in \mathbb{R}^{n_1}, \boldsymbol{\chi} \in \mathbb{R}^{n_2}, n = n_1 + n_2$$

En otras palabras, se tiene un sistema de n ecuaciones con n incógnitas, pero n_2 de ellas se pueden expresar, mediante n_2 de las ecuaciones, como funciones de las n_1 restantes, esto permite establecer el siguiente algoritmo para resolver el sistema:

Algoritmo 3.1 Solución de un sistema de n ecuaciones de las cuales n_2 dependen de las n_1 restantes

Entradas: n_1 , *tolerancia*, x_1^{sup} , x_2^{sup} , ..., $x_{n_1}^{sup}$

Salidas: x_1, x_2, \dots, x_{n_1}

inicio

- 1 **ha_convergado = falso**
 $x_i^{(0)} = x_i^{sup}$, para $i = 1, \dots, n_1$
 $i = 0$
- 2a **hacer**
 - 2.1 $x_{n_1+1} = f_1(x_1^{(i)}, x_2^{(i)}, \dots, x_{n_1}^{(i)})$,
 - 2.2 $x_{n_1+2} = f_2(x_1^{(i)}, x_2^{(i)}, \dots, x_{n_1}^{(i)})$,
 - ...
 - 2.3 $x_n = x_{n_1+n_2} = f_{n_2}(x_1^{(i)}, x_2^{(i)}, \dots, x_{n_1}^{(i)})$

- 2.4 Calcular $x_1^{(i+1)}, x_2^{(i+1)}, \dots, x_{n_1}^{(i+1)}$
 con el método numérico aplicado a las ecuaciones de la forma
 $f_{n_2+i}(x_1^{(i)}, x_2^{(i)}, \dots, x_{n_1}^{(i)}, x_{n_1+1}, x_{n_1+2}, \dots, x_n)$, para $i = 1, \dots, n_1$
- 2.6 **si** $\frac{\|x^{(i+1)} - x^{(i)}\|}{\frac{1}{2}(\|x^{(i+1)}\| + \|x^{(i)}\|)} \leq \text{tolerancia}$ **entonces** ha_convergió = **verdadero**
 $i = i + 1$
- 2b **mientras no** ha_convergió
- 3 **regresar** $x_1^{(i)}, x_2^{(i)}, \dots, x_n^{(i)}$
- fin**

Las ecuaciones que se resuelven simultáneamente

Para disminuir el tamaño del sistema de ecuaciones, se puede optar por escoger un conjunto de variables cuyos valores deben irse iterando al ir resolviendo un sistema de ecuaciones de manera simultánea (estas son las variables denominadas χ o $x_{n_1+1}, x_{n_1+2}, \dots, x_{n_1+n_2}$ en la sección anterior) y otro conjunto de variables que puede calcularse mediante ecuaciones fáciles de resolverse analíticamente (por no formar sistema con otras, es decir, que no están acopladas a otras, y/o por ser lineales o no ser fuertemente no lineales), o que se resuelven con otro método numérico no anidado en el que resuelve el sistema de ecuaciones simultáneas (como lo son las variables de la corriente de alimentación).

Primero listemos las variables de la torre que se fijarán (variables independientes):

Datos

- n número de componentes. En este trabajo son 4: n-hexano, n-heptano, n-octano y n-nonano, que serán denominados componente 1, 2, 3 y 4 respectivamente,
- $z_{F,i}$ fracción molar total de los componentes en la corriente de alimentación. Se fijan las fracciones de tres de ellos: $z_{F,1}, z_{F,2}$ y $z_{F,3}$, la del cuarto se calcula: $z_{F,4} = 1 - (z_{F,1} + z_{F,2} + z_{F,3})$,
- F flujo molar de la corriente de alimentación, (mol/h),
- T_F temperatura del flujo de alimentación, ($^{\circ}\text{C}$),
- P_F presión del flujo de alimentación, (kPa). En este trabajo, se supone que no hay cambio de presión a lo largo de la torre (caída de presión despreciable) y que la presión en ella es igual a la presión de la corriente de alimentación. Para todas las corridas se fija un valor de P_F de 200 kPa,

- N es el número de etapas de la torre. En este trabajo este número fue siempre impar. Las etapas se cuentan de abajo hacia arriba, iniciando por el rehervidor, que es la etapa 1,
- A es el número de la etapa de alimentación. En este trabajo fue siempre la etapa justo en medio de todas las etapas (p. ej.: para $N=7$, $A=4$; para $N=9$, $A=5$, etc.),
- R_d relación de recirculación en el domo de la torre: $R_d = \frac{L_R}{D}$ (ver figura 2.1),
- R_b fracción de vaporización en el rehervidor: $R_b = \frac{V_1}{B}$ (ver figura 2.1),
- T_{cond} temperatura de salida del condensador. En realidad, no se fija esta temperatura, lo que se fija es que el líquido que sale del condensador esté saturado, y, por tanto, que la temperatura de la mezcla que sale del él esté a la temperatura de burbuja de esa mezcla: $T_{cond} = T_{b,N+2}$.

Ahora listemos las

Variables calculadas no obtenidas con el método numérico para resolver el sistema de ecuaciones no lineales simultáneas

Variables de la corriente de entrada a la torre. Estas variables se calculan sólo una vez en cada corrida del programa, es decir, no entran en el ciclo iterativo del método numérico para resolver el sistema de ecuaciones. Estas variables son:

- ψ_F fracción vaporizada de la corriente de alimentación a la torre, F , se calcula por medio de un cálculo de *flash* isotérmico (que requiere el uso de métodos numéricos buscadores de ceros),
- $K_{F,i}$ coeficientes de reparto para cada uno de los componentes en la corriente de alimentación: $K_{F,1}$, $K_{F,2}$ y $K_{F,3}$. Se calculan con la ec. (1.3) aplicada a esa corriente: $K_{F,i}(T_F, P_F) = \frac{1}{P_F} \exp\left(A_i + \frac{B_i}{C_i + T_F}\right)$. Se requieren los coeficientes de reparto, $K_{F,i}$, en cálculo del flash isotérmico para calcular ψ_F ,
- $x_{F,i}$ fracciones molares de los componentes en la fase líquida de la corriente de alimentación a la torre, F , $x_{F,1}$, $x_{F,2}$, $x_{F,3}$ y $x_{F,4}$. Se calculan con la ec. (1.20) aplicada a corriente F : $x_{F,i} = \frac{z_{F,i}}{[(1-\psi_F) + \psi_F K_{F,i}]}$,
- $y_{F,i}$ fracciones molares de los componentes en la fase de vapor de la corriente de alimentación a la torre, F , $y_{F,1}$, $y_{F,2}$, $y_{F,3}$ y $y_{F,4}$. Se calculan con la ec. (1.18) aplicada a la corriente de alimentación, F : $y_{F,i} = K_{F,i} x_{F,i}$

\underline{H}_F entalpía molar de la corriente de alimentación a la torre, F . Es la suma de la entalpía molar de la fase líquida y la de la fase de vapor: $\underline{H}_F = \underline{H}_{L,F} + \underline{H}_{V,F}$, donde $\underline{H}_{V,F}$ se calcula con la ec. (2.1) y $\underline{H}_{L,F}$ se calcula con la ec. (2.2)

Otras variables. Estas variables cambian a lo largo de la ejecución del programa, más específicamente, cada vez que cambian las variables que son iteradas con el método numérico para resolver el sistema de ecuaciones simultáneas; es decir, cambian en cada iteración de este método, sin embargo, no son calculadas con él. Se obtienen resolviendo analíticamente (esto es, se despejan de) las ecuaciones donde aparecen. Estas variables son:

L_{N+2} flujo molar líquido que sale del condensador, (mol/h). Se calcula con la ecuación:
 $L_{N+2} = V_N$,

D corriente de salida del domo de la torre (mol/h). Se calcula con la ecuación: $D = \frac{L_{N+2}}{R_d+1}$,

L_{N+1} flujo líquido que sale del condensador y se recircula a la torre (mol/h). Se calcula con la ecuación: $L_{N+1} = R_d D$,

B es el flujo líquido que sale por el fondo de la torre (mol/h). Se calcula con la ecuación:
 $B = \frac{V_1}{R_b}$. Otra denominación para B es L_1 es decir, $L_1 \equiv B$,

$K_{i,j}$ es el coeficiente de reparto del componente j en la etapa i . Se calcula con las ecuaciones: $K_{i,j}(T_i, P_i) = \frac{1}{P_i} \exp\left(A_j + \frac{B_j}{C_j + T_i}\right)$, para $i=1, \dots, N, j=1, \dots, n$,

$y_{i,n}$ es la fracción molar en fase de vapor del n -ésimo componente en la etapa i . Se calcula con las ecuaciones: $y_{i,n} = 1 - \sum_{j=1}^{n-1} y_{i,j}$, para $i=1, \dots, N$. Para este trabajo: $y_{i,4} = 1 - (y_{i,1} + y_{i,2} + y_{i,3})$,

$x_{i,j}$ es la fracción molar en fase líquida del componente j en la etapa i . Se calcula con las ecuaciones: $x_{i,j} = \frac{y_{i,j}}{K_{i,j}}$, para $i=1, \dots, N, j=1, \dots, n$,

$x_{N+2,j}$ es la fracción molar del componente j en la corriente de salida del condensador. Se calcula con la ecuación: $x_{N+2,j} = y_{N,j}$, para $j=1, \dots, n$,

ψ_i fracción que sale en fase de vapor de las corrientes que entran a la etapa i . Se calcula con la fórmula: $\psi_i = \frac{V_i}{V_{i-1} + L_{i+1}}$, para $i=1, \dots, N \wedge i \neq A \wedge i \neq 1$, $\psi_A = \frac{V_A}{V_{A-1} + L_{A+1} + F}$,
 $\psi_1 = \frac{V_1}{L_1}$,

$z_{i,j}$ es la fracción molar total del componente j en las corrientes de entrada a la etapa i . Se calcula con las ecuaciones: $z_{i,j} = \frac{V_{i-1}y_{i-1,j} + L_{i+1}x_{i+1,j}}{V_{i-1} + L_{i+1}}$, para $i = 1, \dots, N \wedge i \neq A \wedge i \neq 1$,

$$z_{A,j} = \frac{V_{A-1}y_{A-1,j} + L_{A+1}x_{A+1,j} + Fz_{F,j}}{V_{A-1} + L_{A+1} + F}, z_{1,j} = x_{2,j}, \text{ para } j = 1, \dots, n,$$

$z_{N,j}, z_{N+2,j}, z_{N+1,j}$ son las fracciones molares totales de la corriente que entra al condensador, V_N , de la que sale del condensador, L_{N+2} , y del divisor, L_{N+1} , en el domo de la torre (ver figura 2.1). Todas estas fracciones se calculan con las ecuaciones: $z_{N,j} = z_{N+2,j} = z_{N+1,j} = y_{N,j}$, para $j = 1, \dots, n$,

T_{N+2}, T_{N+1} son las temperaturas de la corriente que sale del condensador y de la corriente que sale del divisor en el domo de la torre, son iguales entre sí: $T_{N+2} = T_{N+1}$ (ver figura 2.1), ($^{\circ}\text{C}$). Como se ha fijado que la corriente que sale del domo de la torre es un líquido saturado, su temperatura se calcula con la ecuación de la temperatura del punto de burbuja, que es la ec. (1.13).

$\underline{H}_{V,i}$ entalpía molar de la corriente en fase de vapor, interna de la torre, que sale de la etapa i (J/mol). Se calcula con las ecuaciones: $\underline{H}_{V,i,j} =$

$$\sum_{j=1}^n \left(y_{V,i,j} \int_{T_{ref}}^{T_i} C_{p,V,j}(T) dT + \Delta \underline{H}_{ebull,j} \right), \text{ para } i = 1, \dots, N,$$

$\underline{H}_{L,i}$ entalpía molar de la corriente en fase líquida, interna de la torre, que sale de la etapa i (J/mol). Se calcula con las ecuaciones: $\underline{H}_{L,i,j} = \sum_{j=1}^n \left(x_{L,i,j} \int_{T_{ref}}^{T_i} C_{p,L,j}(T) dT \right)$, para $i = 1, \dots, N$,

\dot{Q}_D flujo de calor que retira el condensador en el domo de la torre, (J/h). Se calcula con la ecuación: $\dot{Q}_D = V_N(\underline{H}_{L_{N+2}} - \underline{H}_{V_N})$.

Variables calculadas con el método numérico para resolver el sistema de ecuaciones no lineales simultáneas

Los valores de las variables de esta sección son calculados con un método numérico para resolver sistemas de ecuaciones no lineales, el método es iterativo. En la tabla 3.1 se muestran las variables y su número, en la 3.2 se muestran las ecuaciones independientes y su número.

Tabla 3.1 Variables cuyos valores son calculados por el método numérico que resuelve sistemas de ecuaciones no lineales simultáneas

Variables	Observaciones	No. de variables
V_i	Son los flujos de vapor internos de la torre; $1 \leq i \leq N$	N
L_i	Son los flujos en fase líquida internos de la torre; $2 \leq i \leq N$	$N - 1$
$y_{i,j}$	Son las fracciones molares de los componentes de las corrientes de vapor internas de la torre; $1 \leq i \leq N, 1 \leq j \leq n-1$	$N(n - 1)$

T_i	Son las temperaturas de las etapas de la torre; $1 \leq i \leq N$	N
\dot{Q}_B	Es el flujo de calor que se alimenta al rehervidor (la etapa 1)	1
Número total de variables =		$N(2 + n)$

Tabla 3.2 Ecuaciones que son resueltas por el método numérico que resuelve sistemas de ecuaciones no lineales simultáneas

Ecuaciones	Observaciones	No. de ecuaciones
$\sum_{j=1}^n \frac{z_{i,j}}{\psi_i(K_{i,j} - 1) + 1} = 0$	Ecuaciones del flash isotérmico; $1 \leq i \leq N$	N
$V_{i-1} + L_{i+1} - (V_i + L_i) = 0,$ para $i=1, \dots, N \wedge i \neq 1 \wedge i \neq A$ $L_2 - (V_1 + L_1) = 0,$ para $i=1$ $V_{A-1} + L_{A+1} + F - (V_A + L_A) = 0,$ para $i=A$	Ecuaciones de balance de materia totales para cada etapa; $1 \leq i \leq N$	N
$V_{i-1}y_{i-1,j} + L_{i+1}x_{i+1,j} - (V_i y_{i,j} + L_i x_{i,j}) = 0$ para $i=1, \dots, N \wedge i \neq 1 \wedge i \neq A$ $L_2 x_{2,j} - (V_1 y_{1,j} + L_1 x_{1,j}) = 0,$ para $i=1$ $V_{A-1}y_{A-1,j} + L_{A+1}x_{A+1,j} + Fz_j - (V_A y_{A,j} + L_A x_{A,j}) = 0,$ para $i=A$	Ecuaciones de balance de materia parciales (de los n componentes menos uno) para todas las etapas; $1 \leq i \leq N,$ $1 \leq j \leq n-1$	$N(n-1)$
$V_{i-1}\underline{H}_{y,i-1} + L_{i+1}\underline{H}_{L,i+1} - (V_i \underline{H}_{y,i} + L_i \underline{H}_{L,i}) = 0$ para $i=1, \dots, N \wedge i \neq 1 \wedge i \neq A$ $L_2 \underline{H}_{L,2} - (V_1 \underline{H}_{y,1} + L_1 \underline{H}_{L,1}) - \dot{Q}_B = 0,$ para $i=1$ $V_{A-1} \underline{H}_{y,A-1} + L_{A+1} \underline{H}_{L,A+1} + F \underline{H}_F - (V_A \underline{H}_{y,A} + L_A \underline{H}_{L,A}) = 0,$ para $i=A$	Ecuaciones de balance de energía para todas las etapas; $1 \leq i \leq N.$	N
Número total de ecuaciones =		$N(2 + n)$

Las tablas 3.1 y 3.2 nos muestran un sistema de ecuaciones independientes con un número de incógnitas igual al número de ecuaciones. El método que se usó para resolver ese sistema se describe a continuación.

El método numérico

En el algoritmo 3.1 se puede observar (ver línea 2.4) que el método numérico resuelve n_1 de las incógnitas, ya que las otras n_2 se pueden calcular analíticamente con las otras n_2 ecuaciones restantes, lo que facilita las cosas para el método numérico.

El método de Newton generalizado (Fausett, 2003, p. 244; Linz y Wang, 2003, pp. 185-187) es:

$$\mathbf{x}^{i+1} = \mathbf{x}^i - (\mathbf{J}^i)^{-1} \mathbf{f}(\mathbf{x}^i) \quad (3.5)$$

Donde \mathbf{J}^i es la matriz jacobiana:

$$\mathbf{J}^i = \begin{bmatrix} \left(\frac{\partial f_1}{\partial x_1}\right)_{\mathbf{x}^i} & \left(\frac{\partial f_1}{\partial x_2}\right)_{\mathbf{x}^i} & \dots & \left(\frac{\partial f_1}{\partial x_{n_1}}\right)_{\mathbf{x}^i} \\ \left(\frac{\partial f_2}{\partial x_1}\right)_{\mathbf{x}^i} & \left(\frac{\partial f_2}{\partial x_2}\right)_{\mathbf{x}^i} & \dots & \left(\frac{\partial f_2}{\partial x_{n_1}}\right)_{\mathbf{x}^i} \\ \vdots & \vdots & \dots & \vdots \\ \left(\frac{\partial f_{n_1}}{\partial x_1}\right)_{\mathbf{x}^i} & \left(\frac{\partial f_{n_1}}{\partial x_2}\right)_{\mathbf{x}^i} & \dots & \left(\frac{\partial f_{n_1}}{\partial x_{n_1}}\right)_{\mathbf{x}^i} \end{bmatrix} \quad (3.6)$$

El método numérico que se empleó para resolver el sistema de ecuaciones fue una aproximación del método de Newton generalizado:

$$\mathbf{x}^{i+1} = \mathbf{x}^i - (\mathbf{A}^i)^{-1} \mathbf{f}(\mathbf{x}^i) \quad (3.7)$$

Donde

\mathbf{A}^i es una matriz que es una estimación de la matriz jacobiana. Se calculó evaluando numéricamente cada una de las derivadas parciales que aparecen en esa matriz. Las derivadas parciales se estimaron usando una aproximación de la definición de la derivada. La definición de la derivada es:

$$\left(\frac{\partial f_i}{\partial x_j}\right)_{\mathbf{x}^k} = \lim_{\Delta x_j \rightarrow 0} \frac{f_i(x_1^k, \dots, x_j^k + \Delta x_j, \dots, x_{n_1}^k) - f_i(x_1^k, \dots, x_j^k, \dots, x_{n_1}^k)}{\Delta x_j} \quad (3.8)$$

La aproximación de la definición de la derivada que se usó fue:

$$\left(\frac{\partial f_i}{\partial x_j}\right)_{\mathbf{x}^k} \approx \frac{f_i(x_1^k, \dots, x_j^k + \Delta x_j^k, \dots, x_{n_1}^k) - f_i(x_1^k, \dots, x_j^k, \dots, x_{n_1}^k)}{\Delta x_j} \quad (3.9)$$

Donde $\Delta x_j = \alpha x_j$, $\alpha = 1 \times 10^{-6}$.

Para arrancar el método numérico se debe hacer una suposición inicial, se debe dar un valor inicial, \mathbf{x}^0 , del vector \mathbf{x} . Luego Se aplica (3.7) hasta que se cumpla que:

$$\frac{\|x^{i+1} - x^i\|}{\frac{1}{2}(\|x^{i+1}\| + \|x^i\|)} \leq \textit{tolerancia} \quad (3.10)$$

En realidad, no se usó (3.7). Se usó la forma alternativa (basado en Fausett, 2003, p.262, quien aplica esta forma para el método de Newton generalizado):

$$\mathbf{A}^i \Delta \mathbf{x}^{i+1} = -\mathbf{f}(\mathbf{x}^i) \quad (3.11)$$

Donde

$$\Delta \mathbf{x}^{i+1} = \mathbf{x}^{i+1} - \mathbf{x}^i \quad (3.12)$$

La ecuación (3.11) es un sistema de ecuaciones lineales:

$$a_{1,1}\Delta x_1^{i+1} + a_{1,2}\Delta x_2^{i+1} + \dots + a_{1,n_1}\Delta x_{n_1}^{i+1} = -f_1(\mathbf{x}^i)$$

$$a_{2,1}\Delta x_1^{i+1} + a_{2,2}\Delta x_2^{i+1} + \dots + a_{2,n_1}\Delta x_{n_1}^{i+1} = -f_2(\mathbf{x}^i)$$

...

$$a_{n_1,1}\Delta x_1^{i+1} + a_{n_1,2}\Delta x_2^{i+1} + \dots + a_{n_1,n_1}\Delta x_{n_1}^{i+1} = -f_{n_1}(\mathbf{x}^i)$$

Se usó el método de Eliminación de Gauss (Fausett, 2003 pp. 98-107; Linz, 2003, pp. 32-36; McCracken y Dorn, 1979 pp. 247-255; Sharma, 2007, pp. 81-82) para resolver este sistema de ecuaciones, obteniendo así el vector $\Delta \mathbf{x}^{i+1} = (\Delta x_1^{i+1}, \Delta x_2^{i+1}, \dots, \Delta x_{n_1}^{i+1})$, y luego a partir de (3.12) calcular \mathbf{x}^{i+1} .

Capítulo 4. Las redes multicapa de perceptrones usadas

Las redes neuronales artificiales que se usaron en este trabajo fueron las redes multicapa de perceptrones.

Las redes multicapa de perceptrones pueden ser usadas como clasificadoras o como ajustadoras de funciones. Es en su segunda aplicación, como ajustadoras, que se emplean en este trabajo.

El modelo de neurona que usan, el perceptrón, está basado en el modelo propuesto por McCulloch y Pitts desde 1943 (Caudill 1992, p. 2). Una descripción del perceptrón se muestra en la figura 3.1.

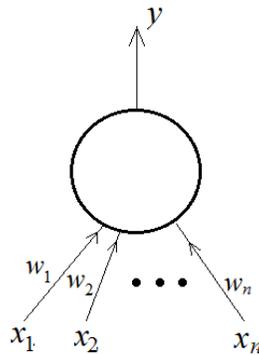


Figura 4.1. Esquema de un perceptrón

Las flechas que apuntan al perceptrón representan los estímulos que recibe la neurona de otras neuronas o del medio ambiente. Esos estímulos pueden ser exitatorios o inhibitorios. Sobre las flechas se representan en el esquema de la figura 4.1 variables nombradas w con diferentes subíndices, representan los *factores de eficiencia sináptica*, también llamados simplemente *pesos*. Éstos representan con qué facilidad pasan los datos por la entrada dada. La salida, y , para este tipo de perceptrón es una variable continua, se calcula de la siguiente manera:

$$\sigma = \sum_i w_i x_i - \theta \quad (4.1)$$

$$y = \frac{1}{1 + e^{-\sigma}} \quad (4.2)$$

Donde θ es el umbral.

$$x_i, w_i, \theta \in \mathbb{R}$$

Obsérvese que $y \in (0, 1)$, esto implica que deben escalarse los valores de las funciones que se deseen ajustar, para que queden en ese mismo intervalo.

En los siguientes párrafos veremos a grandes rasgos cómo están constituidas y cómo funcionan las redes multicapa de perceptrones (redes MLP, *multilayer perceptron networks*).

La Arquitectura de la Red

Una red MLP está constituida por una capa de entrada, capas intermedias (también llamadas capas ocultas) y una capa de salida. En este trabajo se usaron redes MLP con una sola capa intermedia. En la figura 4.2 se muestra un esquema que describe la arquitectura de las redes usadas.

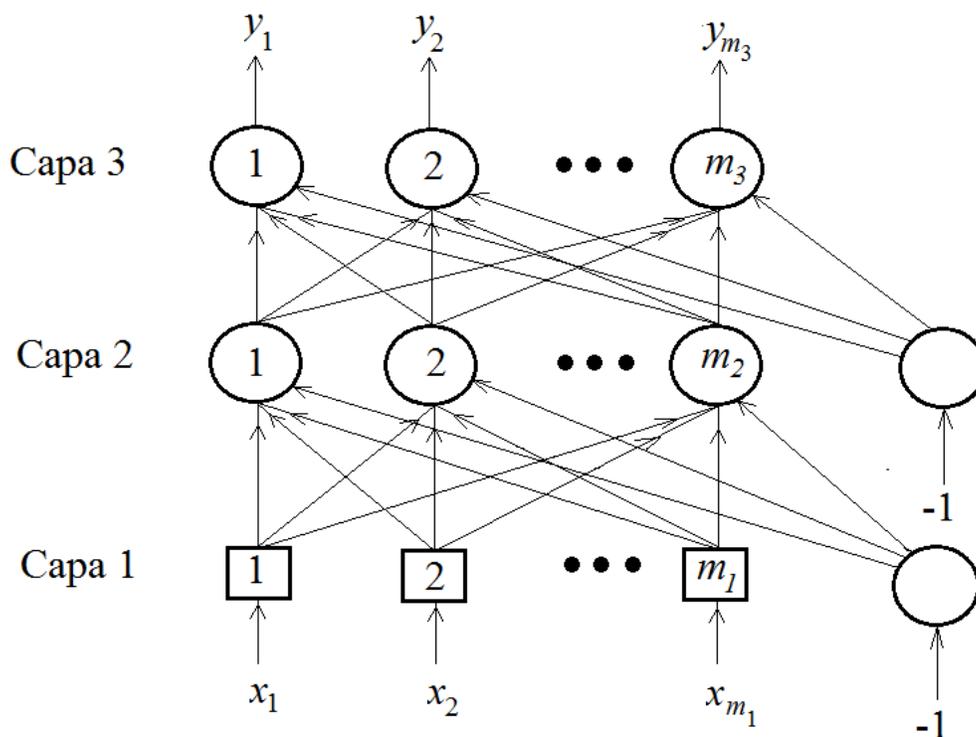


Figura 4.2. Red de perceptrones de tres capas.

Las capas 2 y 3 son de perceptrones, la capa 1 es de distribuidores, elementos que representan a los sensores que mandan la información a las neuronas; éstos distribuyen la señal que reciben a todos los perceptrones a los que están conectados. La capas 1, 2 y 3 son la capa de entrada, la oculta y la de salida respectivamente. Se puede observar que la red MLP se representa con un grafo en el que los nodos de la capa de entrada están totalmente conectados con los de la capa oculta, y los de ésta a su vez están totalmente conectados con los de capa de salida, mediante arcos dirigidos. Estos arcos representan los nervios que interconectan a las neuronas. En este ejemplo, a la derecha se muestran dos neurodos a los que entra el valor de -1, se trata de los neurodos de polarización (*bias neurodes*), de ellos salen conexiones a

las capas de perceptrones, éstas representan los valores de los umbrales de cada perceptrón. El aprendizaje modifica diferentes variables (por ejemplo, el grosor de los nervios y el tipo de excitación –inhibitoria si el valor que pasa por la conexión es negativo, activadora si es positivo) que inciden en la facilidad con la que las neuronas se transmiten información entre sí; para representar esto a cada arco dirigido que une la capa de entrada con la oculta y ésta con la de salida, se le asocia un peso (como se ha indicado arriba en la descripción del perceptrón). El proceso de aprendizaje de una red neuronal artificial consiste en ir modificando progresivamente el valor de sus pesos.

El entrenamiento de una red neuronal para que funcione como ajustadora de funciones

El problema que se desea resolver es el de que una red neuronal ajuste varias funciones que se tienen expresadas en forma tabular. A este proceso de ajuste de la función con la red neuronal, se le llama *entrenamiento de la red*. Se debe construir una tabla de entrenamiento que servirá para entrenar a la red, a esta tabla se la llama *tabla de entrenamiento*. Para probar si la red ya entrenada aprendió correctamente, se debe construir también otra tabla llamada *tabla de prueba*. Cada renglón de estas tablas está constituido de dos partes, el vector de las variables independientes, \mathbf{x} , y el vector de variables dependientes, \mathbf{y} .

Para la red de la figura 2.1 se tiene $\mathbf{x} \in \mathbb{R}^{m_1}$, $\mathbf{y} \in \mathbb{R}^{m_3}$. Las tablas de entrenamiento y de prueba se pueden representar con las ecuaciones (4.3) y (4.4).

$$T_e = ((\mathbf{x}_1, \mathbf{y}_1), \dots, (\mathbf{x}_r, \mathbf{y}_r), \dots, (\mathbf{x}_{R_e}, \mathbf{y}_{R_e})) \quad (4.3)$$

$$T_p = ((\mathbf{x}_1, \mathbf{y}_1), \dots, (\mathbf{x}_r, \mathbf{y}_r), \dots, (\mathbf{x}_{R_p}, \mathbf{y}_{R_p})) \quad (4.4)$$

Donde

T_e es la tabla de entrenamiento,

T_p es la tabla de prueba,

R_e es el número de renglones de la tabla de entrenamiento,

R_p es el número de renglones de la tabla de prueba.

Se le llama *época* al hecho de alimentar (uno a la vez) todos los vectores de variables independientes de la tabla de entrenamiento (y obtener los valores de las variables dependientes estimados por la red cada vez).

Representemos el funcionamiento de la red de perceptrones con la siguiente ecuación:

$$\mathbf{y}_r^{(3)} = P(\mathbf{w}(t), \mathbf{x}_r) \quad (4.5)$$

En la ecuación (4.5) se representa la llamada *propagación hacia adelante* (*forward propagation*). Ésta consiste en alimentar a la red el vector de valores de las variables independientes, pasar por la capa de entrada para que distribuya los valores de sus componentes a los perceptrones de la capa oculta, hacer los cálculos correspondientes para obtener las salidas de los perceptrones de esa capa; alimentar las salidas de esa capa a los perceptrones de la capa de salida, hacer los cálculos correspondientes en los perceptrones de esa capa para obtener las salidas de la red.

Lo que se busca en el proceso de entrenamiento, es que, mediante la modificación de los pesos, el vector de salida de la red, $\mathbf{y}_r^{(3)}$, sea muy similar al vector de variables independientes correspondiente de la tabla, \mathbf{y}_r . Una forma de lograr lo anterior es minimizando la función error:

$$E = \sum_{i=1}^{m_3} (y_i - y_i^{(3)})^2 \quad (4.6)$$

En la ecuación (4.6), y_i es el valor de la i -ésima variable dependiente en el renglón r de la tabla de entrenamiento, y_i^3 es el valor estimado de esa variable obtenido por la red neuronal cuando se le alimentan las variables independientes, \mathbf{x}_r , de ese renglón r . El cálculo de todas las variables dependientes por la red neuronal es lo que se representa en la ecuación (4.5).

El método usado en este trabajo para minimizar la función (4.6) es el método de propagación hacia atrás (*back propagation*), en su modalidad *en línea* (*on line*) (Shepherd, 1997, p. 18). Esta modalidad consiste en modificar los pesos de la red cada vez que se hace una *propagación hacia adelante* (ver arriba), (a diferencia del método *fuera de línea* -*off-line*- que consiste en alimentar una época completa a la red y luego modificar sus pesos).

Henseler (Henseler, 1995, p. 47) nos muestra cómo ir modificando los pesos de la red:

$$w_{i,j}^p(t) = w_{i,j}^p(t-1) - \eta \delta_i^p y_j^{p-1} \quad (4.7)$$

Donde

$$\delta_i^p = \begin{cases} y_i^p (1 - y_i^p) \sum_{j=1}^{m_{p+1}} w_{j,i}^{p+1} \delta_j^{p+1} & 1 < p < 3 \\ y_i^3 (1 - y_i^3) (y_i^3 - y_i) & p = 3 \end{cases} \quad (4.8)$$

Una ecuación que no expresa Henseler de manera explícita es la de la actualización de los umbrales:

$$\theta_i^p(t) = \theta_i^p(t-1) + \eta \delta_i^p \quad (4.7b)$$

Ecuaciones tomadas (con pequeñas modificaciones de notación y aplicándolas a una red de tres capas, como las que se usaron en este trabajo) de Henseler, 1995, p. 47.

Donde

$w_{i,j}^p(t)$ peso de la conexión que va del neurodo (perceptrón o distribuidor) j de la capa $p-1$ al neurodo i de la capa p , en la t -ésima iteración de entrenamiento.

$\theta_i^p(t)$ umbral del i -ésimo neurodo de la capa p . Los umbrales se manejan como pesos de conexiones que van del neurodo de polarización a cada uno de los perceptrones de la red.

η factor de aprendizaje, es fijado por el usuario; por lo general $0 < \eta \leq 1$

y_i valor de la i -ésima variable dependiente en el renglón de la tabla de entrenamiento donde se hizo la más reciente propagación hacia adelante.

El entrenamiento en línea se describe en el algoritmo 4.1.

**Algoritmo 4.1. Propagación hacia atrás en línea (*on line back propagation*)
para entrenar la red MLP**

Inicio	
1	Obtener al azar todos los pesos de la red, $\mathbf{w}(0)$
2	Iniciar un acumulador para almacenar la suma de los errores para cada propagación hacia adelante: $ac_E = 0$
3	Para cada uno de los renglones de la tabla de entrenamiento T_e (4.3)
3.1	Hacer la propagación hacia adelante: $\mathbf{y}_r^3 = P(\mathbf{w}(t), \mathbf{x}_r)$
3.2	Calcular el error con la ecuación (4.6): $E = \sum_{i=1}^{m_3} (y_i - y_i^3)^2$
3.3	Sumar el error en el acumulador de errores: $ac_E = ac_E + E$
3.4	Hacer la propagación hacia atrás para repartir las correcciones del error en cada peso usando las ecuaciones (4.7) y (4.8): $w_{i,j}^p(t) = w_{i,j}^p(t-1) - \eta \delta_i^p y_j^{p-1}$ $\theta_i^p(t) = \theta_i^p(t-1) + \eta \delta_i^p$ $\delta_i^p = \begin{cases} y_i^p(1 - y_i^p) \sum_{j=1}^{m_{p+1}} w_{j,i}^{p+1} \delta_j^{p+1} & 1 < p < 3 \\ y_i^3(1 - y_i^3)(y_i^3 - y_i) & p = 3 \end{cases}$

4	Calcular el error promedio: $\bar{E} = \frac{1}{R_e} \sum_{i=1}^{m_3} (y_i - y_i^3)^2$
5	Si el error promedio es muy grande ir a (2)
	Fin

Capítulo 5. Desarrollo de los programas

Se diseñaron, desarrollaron y depuraron los programas cuyas características se describen a continuación:

1) Programa para hacer los cálculos etapa por etapa de una torre de destilación para una mezcla ideal de alcanos. Lleva a cabo los cálculos de *flash* isotérmico en el rehervidor, y cálculos de *flash* adiabático en las otras etapas. En cada etapa calcula las temperaturas de rocío y burbuja, para obtener el intervalo de temperaturas en las que debe trabajar. Cada iteración la inicia en la etapa de alimentación, desciende haciendo los cálculos etapa por etapa hasta llegar al rehervidor; calcula la cantidad de calor requerida para evaporar en él la fracción fijada de antemano, la fracción restante la saca por el producto del fondo de la torre. Regresa hacia la etapa de alimentación haciendo los cálculos etapa por etapa; al llegar al plato de alimentación sigue hacia el domo, donde está el condensador, haciendo los balances en cada etapa por la que pasa. Al llegar al condensador, calcula la temperatura de burbuja de la mezcla en estado gaseoso que sale por la última etapa, calcula el calor retirado por el condensador y regresa a la torre la fracción de la corriente líquida fijada de antemano, la fracción restante de la corriente la saca como producto de domo. Conforme hace los cálculos en una etapa revisa los valores obtenidos en la iteración anterior, si en una o más etapas difieren en una proporción mayor a la indicada por la tolerancia fijada para tal efecto, se hace una nueva iteración. El algoritmo buscador de ceros que usa para los cálculos de las temperaturas de rocío y burbuja y para la fracción vaporizada es el de la bisección.

2) Programa para hacer los cálculos en una torre de destilación resolviendo iterativamente las ecuaciones de todas las etapas usando el método de Newton con las derivadas estimadas numéricamente. Este programa no considera la torre como un conjunto de etapas en equilibrio colocadas en serie, sino que contempla la totalidad de las ecuaciones que describen el equilibrio fisicoquímico y los balances de materia y energía de todas las etapas como un sistema de ecuaciones algebraicas con términos trascendentes que debe resolver. No calcula las temperaturas de rocío ni de burbuja. Requiere de un valor inicial para todas las variables que debe calcular.

3) Programa para simular una red multicapa de perceptrones. Este programa utiliza el algoritmo de propagación hacia atrás en su versión en línea (*on line, back propagation*), descrito en el capítulo 4 para entrenar y operar en modo de producción una red multicapa de perceptrones. Se puede fijar el número de capas ocultas (en este trabajo siempre se usó solo una capa oculta), el número de perceptrones por capa y el número de épocas con las que se entrena la red. Muestra el error acumulado por época al usar la tabla de entrenamiento y también al usar la tabla de prueba.

4) Programa para obtener las tablas para entrenar y probar la red neuronal. Las variables independientes son:

- a) las fracciones molares totales de los componentes en la corriente de alimentación, $z_{i,A}$ para $i \in \{\text{n-hexano, n-heptano, n-octano, n-nonano}\}$, como son cuatro componentes, sólo se deben fijar tres de estas fracciones, ya que las cuatro deben sumar uno;
- b) la temperatura de la corriente de alimentación, T_F .
- c) el número de etapas de la torre, N ; para que el número de etapas por arriba y por debajo del plato de alimentación fueran iguales, se supusieron siempre valores impares para esta variable y se supuso que el plato de alimentación fue siempre el que se encontraba a la mitad de la torre;
- d) la relación de recirculación en el domo de la torre, R_d , es la relación que existe entre la fracción del flujo que sale del condensador que se recircula a la torre y la fracción de ese flujo que sale de la torre como producto de domo, y
- e) relación de vaporización en el rehervidor, R_f , es la relación que existe entre la fracción de flujo que entra al rehervidor y que es evaporada por él y la fracción de ese flujo que sale de la torre como corriente de fondo.

En total son 7 variables independientes. Se debe alimentar el intervalo entre los que se quiere que el programa obtenga un valor aleatorio para cada una de éstas, es decir, para cada variable se fija un valor máximo y otro mínimo y el programa obtiene un valor aleatorio en ese intervalo incluyendo los extremos.

Se debe alimentar al programa el número de renglones que se desea que tenga la tabla ya sea de entrenamiento o de prueba. El programa obtendrá valores aleatorios de cada una de las variables arriba mencionadas en los intervalos fijados por el usuario para la cantidad de renglones también fijada por él. En ese punto falta completar la tabla con los valores correspondientes para cada renglón de las variables dependientes. Para hacerlo se usa el programa descrito en el inciso (1) de esta sección, el que hace los cálculos etapa a etapa. Este programa es muy estable en cuanto a la forma en la que converge hacia la solución, por lo que fue el que se usó para obtener la parte de las variables dependientes de todas las tablas de entrenamiento y prueba para las redes neuronales que se usaron en este trabajo. Se alimentan al programa (1) los valores de las variables independientes, éste calcula los valores de los flujos de todas las corrientes internas de vapor y de líquido en la torre, así como las dos corrientes líquidas de salida (el producto de domo y el de fondo), también calcula sus composiciones, la temperatura de cada una de las etapas, la temperatura de la corriente de salida del condensador y los flujos de calor del rehervidor y del condensador. De toda esta información se toma sólo una parte, la necesaria para tratar de reconstruir por interpolación la parte que no se toma.

De todos los parámetros que calcula el programa (1), en la tabla se reportan como variables dependientes sólo las siguientes:

- i) Fracciones molares en la fase de vapor del n-hexano, n-heptano y n-octano, en el rehervidor (que es la etapa de equilibrio 1 de la torre), $y_{hex,1}, y_{hep,1}, y_{oct,1}$; flujo molar de la fase de vapor

que sale del rehervidor, V_1 , flujo molar de la fase de líquido que entra al rehervidor, L_2 y la temperatura en el rehervidor, T_1 . Flujo de calor que alimenta el rehervidor al flujo de líquido que entra a él, \dot{Q}_{reherv} . Son en total 7 variables relacionadas con el rehervidor.

ii) Fracciones molares en la fase de vapor del n-hexano, n-heptano y n-octano, en la etapa que está a la mitad o cerca de la mitad de las etapas que van del rehervidor a la etapa de alimentación, $y_{hex, \frac{1}{2} agot}$, $y_{hep, \frac{1}{2} agot}$, $y_{oct, \frac{1}{2} agot}$. El número de esa etapa se calcula con la ecuación:

$$N_{\frac{1}{2} agot} = Entero\left(\frac{A}{2}\right) \quad (5.1)$$

Donde

$N_{\frac{1}{2} agot}$ etapa en la mitad o cercana a la mitad de la sección de agotamiento de la torre,
 $Entero(\cdot)$ es una función que regresa la parte entera del valor del argumento sobre el que opera,
 A es el número de la etapa de alimentación (el número 1 corresponde a la etapa del rehervidor).

En total son 3 variables relacionadas con esta etapa.

iii) Fracciones molares de la corriente de vapor que entra a la etapa de alimentación (A), proveniente de la etapa que está inmediatamente abajo de ella ($A-1$), $y_{hex, A-1}$, $y_{hep, A-1}$, $y_{oct, A-1}$. Flujo molar en fase de vapor que entra a la etapa de alimentación proveniente de la etapa inmediata inferior al plato de alimentación, V_{A-1} . Flujo en fase líquida que sale de la etapa de alimentación, L_A . Fracciones molares de la corriente de vapor que sale de la etapa de alimentación, $y_{hex, A}$, $y_{hep, A}$, $y_{oct, A}$. Flujo molar en fase de vapor que sale de la etapa de alimentación, V_A . Flujo molar del líquido que entra a la etapa de alimentación proveniente de la etapa inmediata superior al plato de alimentación, L_{A+1} . Temperatura de la etapa de alimentación, T_A . En total son 11 variables relativas a la etapa de alimentación.

iv) Fracciones molares de los componentes en el flujo de vapor que sale de la etapa en la mitad o cerca de la mitad de las etapas que van de la etapa de alimentación, A , a la última etapa de la torre, N (la que está inmediatamente abajo del condensador), $y_{hex, \frac{1}{2} rect}$, $y_{hep, \frac{1}{2} rect}$, $y_{oct, \frac{1}{2} rect}$. El número de esta etapa se calcula con la ecuación:

$$N_{\frac{1}{2} rect} = Entero\left(\frac{A + N}{2}\right) \quad (5.2)$$

Donde

$N_{\frac{1}{2} rect}$ etapa en la mitad o cercana a la mitad de la sección de rectificación de la torre,
 N número de la última etapa de la torre (= número de etapas de la torre).

Son en total 3 las variables relativas a la etapa en la mitad o cercana a la mitad de la sección de rectificación de la torre.

v) Fracciones molares de las sustancias en el flujo de vapor que sale de la última etapa de la torre, N , $y_{hex,N}$, $y_{hep,N}$, $y_{oct,N}$. Flujo molar del vapor que sale de la última etapa de la torre, V_N . Flujo molar del líquido que sale de la última etapa de la torre, L_N . Temperatura de la última etapa de la torre, T_N . En total son 6 las variables relativas a la última etapa de la Torre. En total son 30 las variables dependientes que se reportan en las tablas de entrenamiento y de prueba para la red neuronal.

5) Programa para obtener por interpolación los valores de las variables dependientes que no calcula la red neuronal.

Recuérdese que el objetivo de este trabajo es usar algunos de los datos obtenidos de un programa que hace los cálculos de balance en una torre etapa por etapa, para entrenar una red neuronal, que sirva, junto con un programa interpolador, para suministrar los puntos iniciales requeridos por el método de Newton con las derivadas calculadas numéricamente, para resolver otros problemas similares.

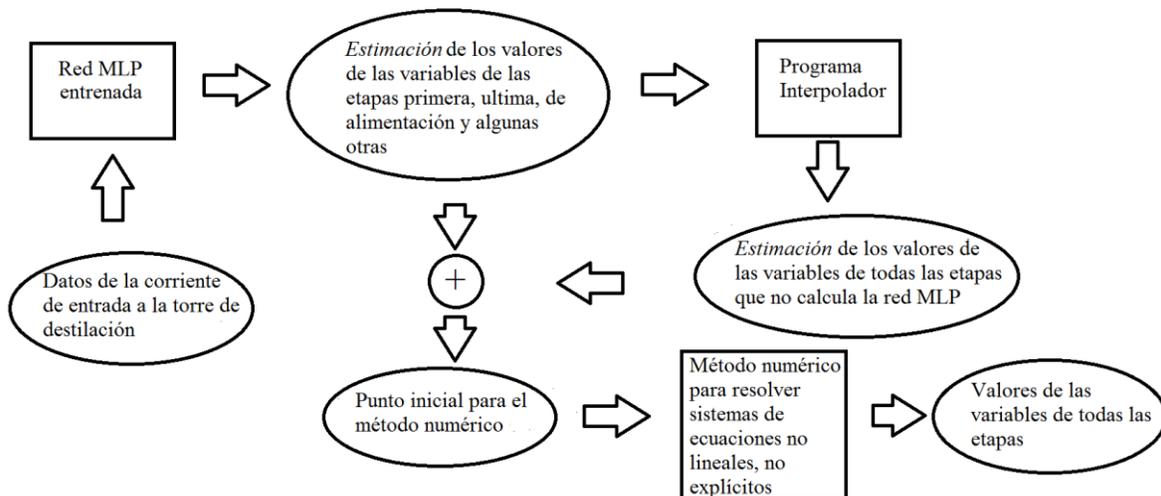


Figura 5.1 Diagrama de flujo de información entre los programas.

Una vez que se tiene una red MLP que ha aprendido el funcionamiento de la torre, se puede usar para estimar los valores de las variables de algunas de sus etapas; el programa interpolador estima los valores de las variables de las etapas restantes. La combinación de los datos calculados por la red MLP y los datos obtenidos por el programa interpolador, sirve para obtener un punto inicial para que un método numérico calcule los valores de las variables de todas las etapas.

Ese punto inicial debe incluir suposiciones iniciales de los valores de *todas* esas variables. Como la red MLP, que es alimentada con los valores de las variables que especifican la corriente de entrada, sólo calcula los valores de las variables de algunas de las etapas, es necesario estimar de algún modo los valores de las variables de las etapas que no calcula la red MLP; esto se hace por interpolación. En la figura 5.1 se esquematiza esto.

Este programa interpolador aplica la interpolación con polinomios de primer orden para estimar los flujos molares internos de la torre en fase líquida y de vapor, así como las temperaturas de las etapas; para estimar los valores de las fracciones molares de las corrientes de vapor usa polinomios de segundo orden. Hace un conjunto de interpolaciones en la sección de agotamiento y otro conjunto de interpolaciones en la sección de rectificación, para calcular los valores de las variables de las etapas en esas secciones. Para hacer las interpolaciones lineales, para obtener los valores de los flujos molares de líquido y vapor y para obtener las temperaturas de las etapas, usa los valores de esas variables en los extremos de esas secciones: para la sección de agotamiento los valores del rehervidor y de la etapa de alimentación, para la sección de rectificación los valores entre la etapa de alimentación y la última etapa de la torre. Para hacer las interpolaciones de segundo orden, para interpolar los valores de las fracciones molares de las sustancias en los flujos en fase de vapor, usa además de los valores de esas variables en las etapas que se acaban de mencionar, aquéllos en las etapas que están a la mitad (o cerca de la mitad) de ambas secciones.

Capítulo 6. Experimentación y resultados

La red MLP

Como ya se mencionó en el capítulo anterior, se manejan 7 variables independientes: $z_{hex,F}$, $z_{hep,F}$, $z_{oct,F}$, T_F , R_d , R_b y N , y 30 variables dependientes: $y_{hex,1}$, $y_{hep,1}$, $y_{oct,1}$, V_1 , L_2 , T_1 , \dot{Q}_{reherv} , $y_{hex,\frac{1}{2}agot}$, $y_{hep,\frac{1}{2}agot}$, $y_{oct,\frac{1}{2}agot}$, $y_{hex,A-1}$, $y_{hep,A-1}$, $y_{oct,A-1}$, V_{A-1} , L_A , $y_{hex,A}$, $y_{hep,A}$, $y_{oct,A}$, V_A , L_{A+1} , T_A , $y_{hex,\frac{1}{2}rect}$, $y_{hep,\frac{1}{2}rect}$, $y_{oct,\frac{1}{2}rect}$, $y_{hex,N}$, $y_{hep,N}$, $y_{oct,N}$, V_N , L_N y T_N . Esto implica que la capa de entrada de la red MLP debe tener 7 y la de salida 30 neurodos. En este trabajo se usaron redes con una sola capa oculta (la capa entre la de entrada y la de salida), el número de neurodos que debe tener esa capa depende de la complejidad de las funciones que se deben ajustar, que en este caso son 30 funciones. El número de neurodos de esa capa se obtuvo por ensayo y error; fue de 44 neurodos.

La red MLP usada tiene entonces 7 neurodos (distribuidores) en la capa de entrada, 44 neurodos (perceptrones) en la capa oculta y 30 neurodos (perceptrones) en la capa de salida.

Entrenamiento de la Red MLP

Se elaboró una tabla de entrenamiento de 200 renglones usando el programa (4) del capítulo anterior. Los intervalos fijados para las 7 variables independientes se muestran en la tabla 6.1.

Tabla 6.1 Intervalos en los que se obtuvieron los valores aleatorios de las diferentes variables independientes para construir las tablas de entrenamiento y de prueba.

$0 \leq z_{hex,F}, z_{hep,F}, z_{oct,F} \leq 1$
$375^\circ C \leq T_F \leq 425^\circ C$
$0.5 \leq R_d \leq 1.5$
$0.5 \leq R_b \leq 1.5$
$7 \text{ etapas} \leq N \leq 23 \text{ etapas}$

Se hizo que los valores de las fracciones $z_{hex,F}$, $z_{hep,F}$ y $z_{oct,F}$ obtenidos aleatoriamente, cumplieran con la restricción:

$$z_{hex,F} + z_{hep,F} + z_{oct,F} \leq 1 \quad (6.1)$$

Ya que se debe cumplir que:

$$z_{hex,F} + z_{hep,F} + z_{oct,F} + z_{non,F} = 1 \quad (6.2)$$

Cada renglón de la tabla de entrenamiento tenía entonces 7 variables independientes y 30 variables dependientes. Para todas las variables que no eran fracciones, se aplicó un escalamiento para que sus valores quedaran en el intervalo [0,1], ya que, como se vio en el capítulo 4, la salida de los perceptrones la da la función de transferencia:

$$y = \frac{1}{1 + e^{-(\sigma-\theta)}} \quad (\text{ec. (4.2) del capítulo 4})$$

donde $y \in (0, 1)$.

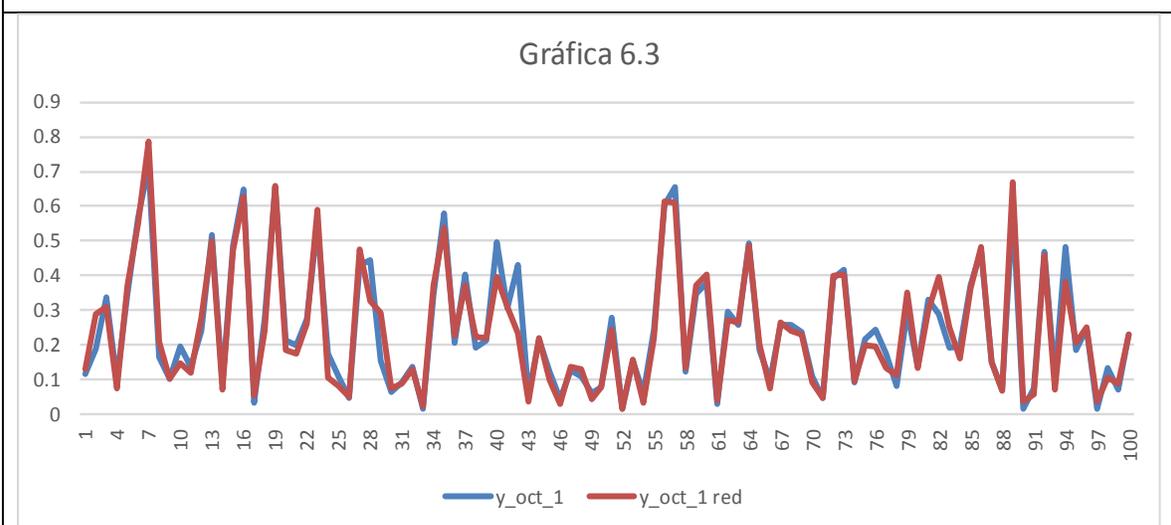
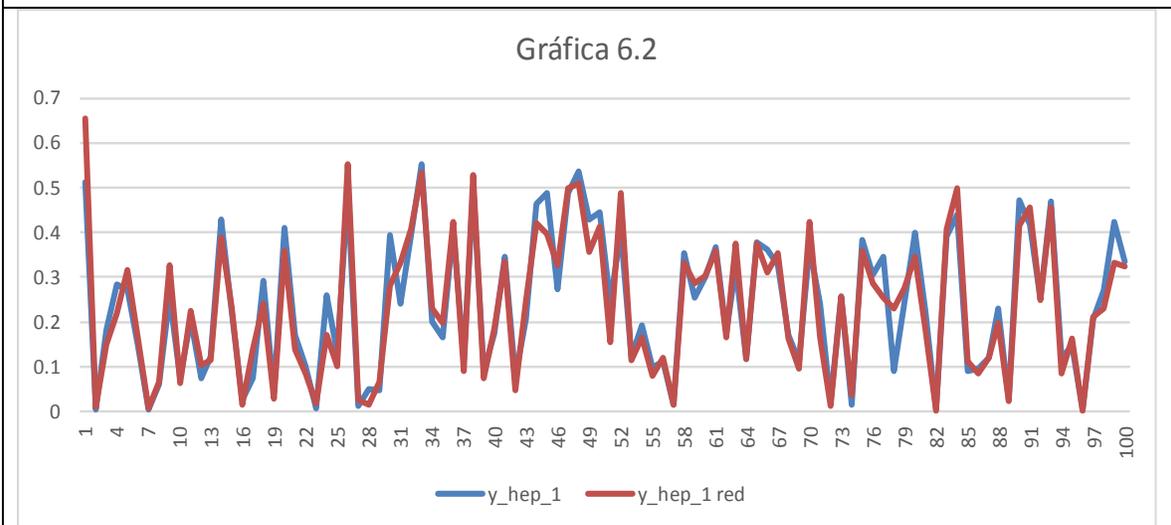
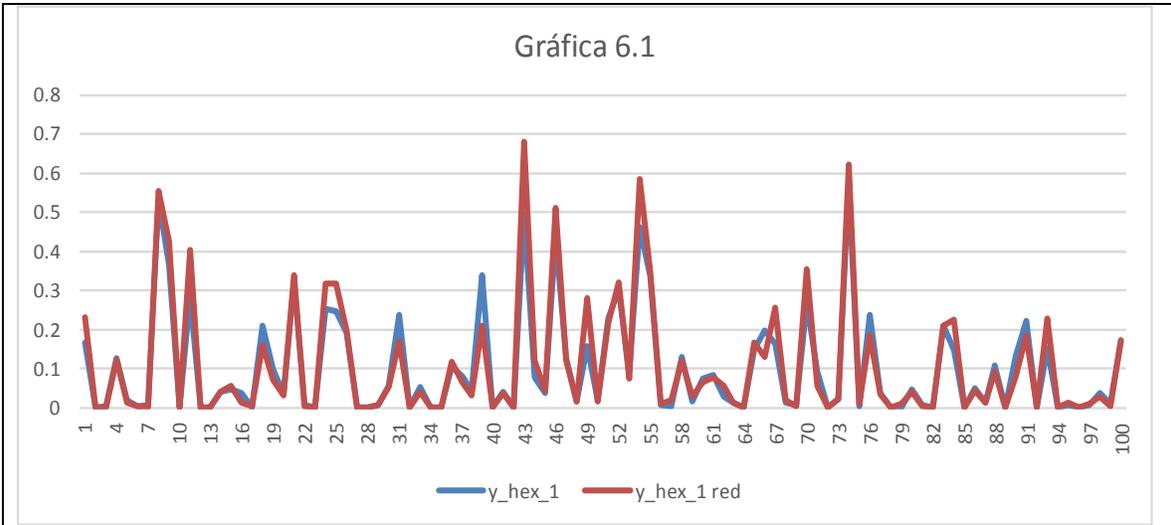
Se entrenó la red MLP aplicando 100,000 épocas de la tabla de entrenamiento. Los valores del error acumulado de una época a lo largo del proceso de entrenamiento se muestran en la tabla 6.2.

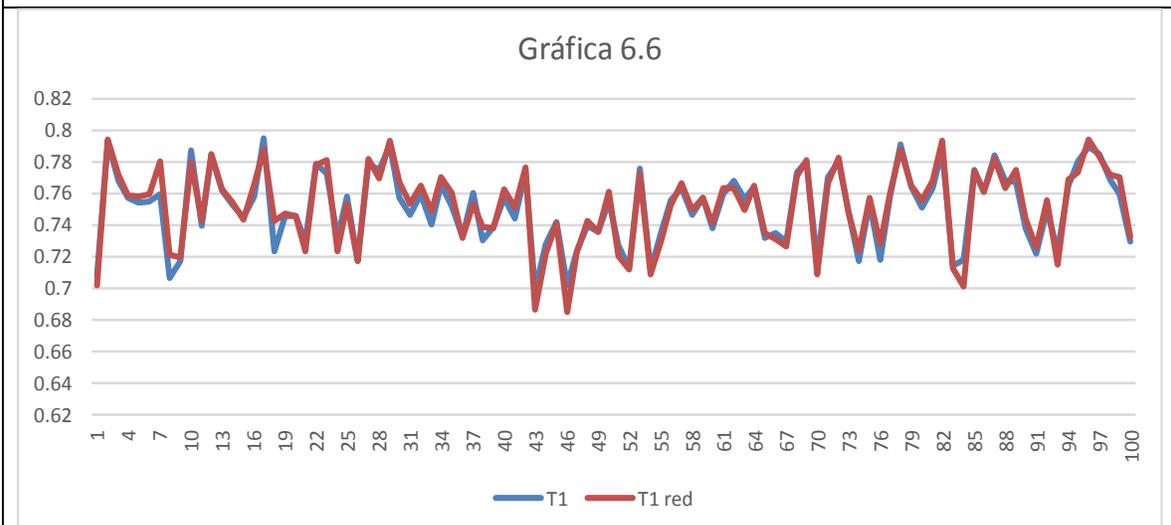
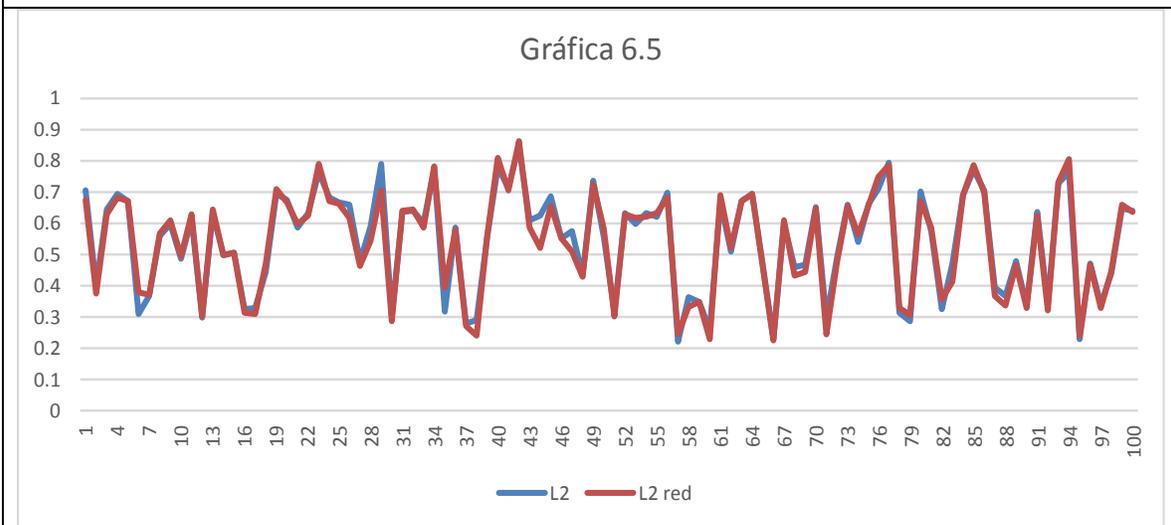
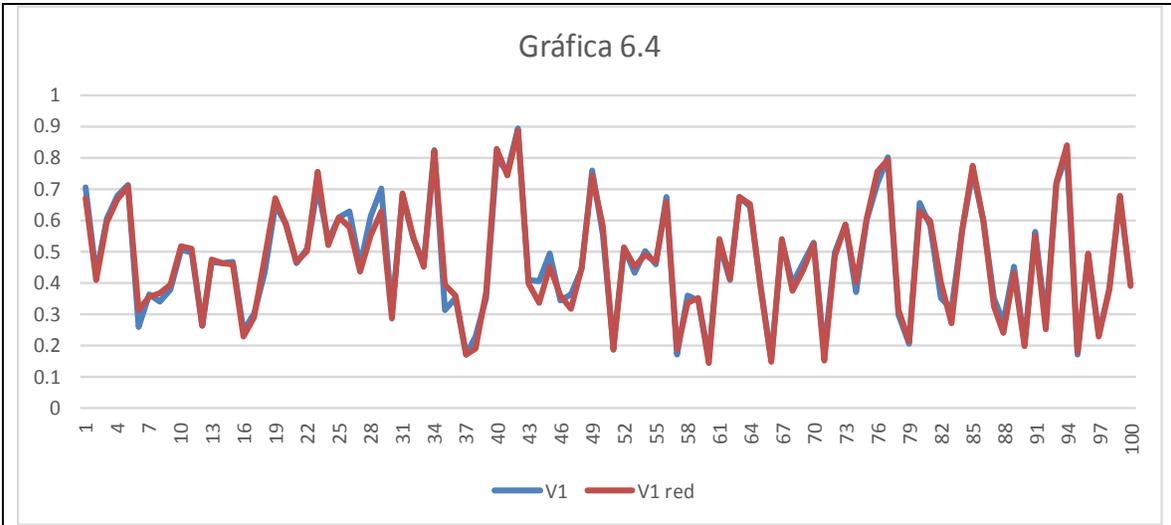
Tabla 6.2 Cambio del error a lo largo del proceso de entrenamiento

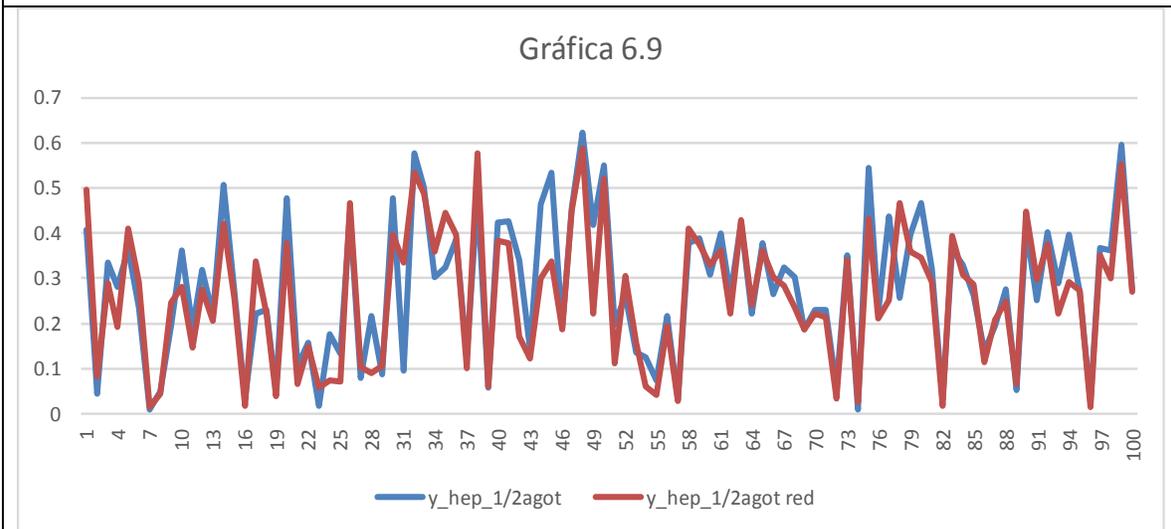
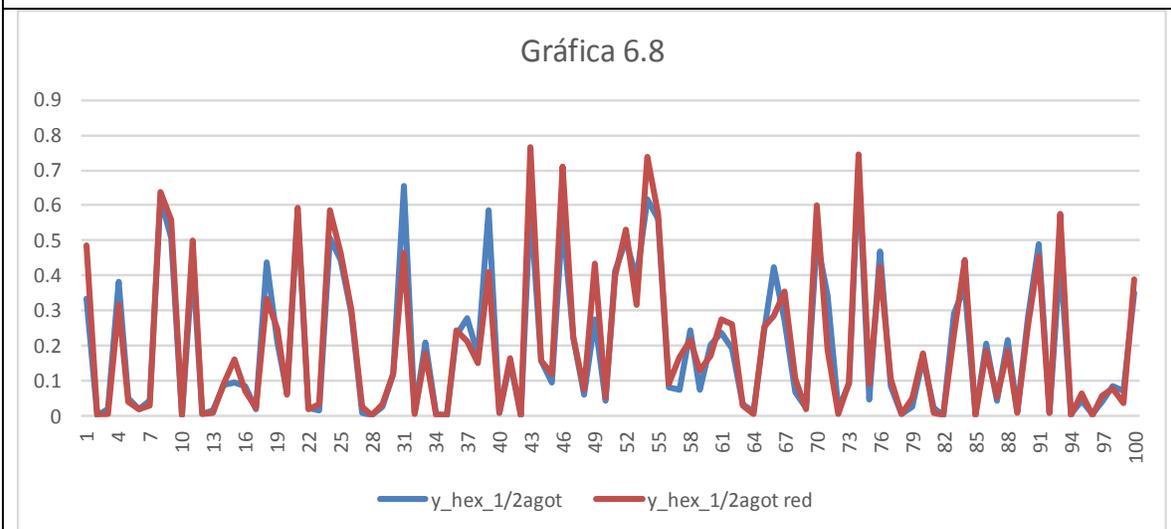
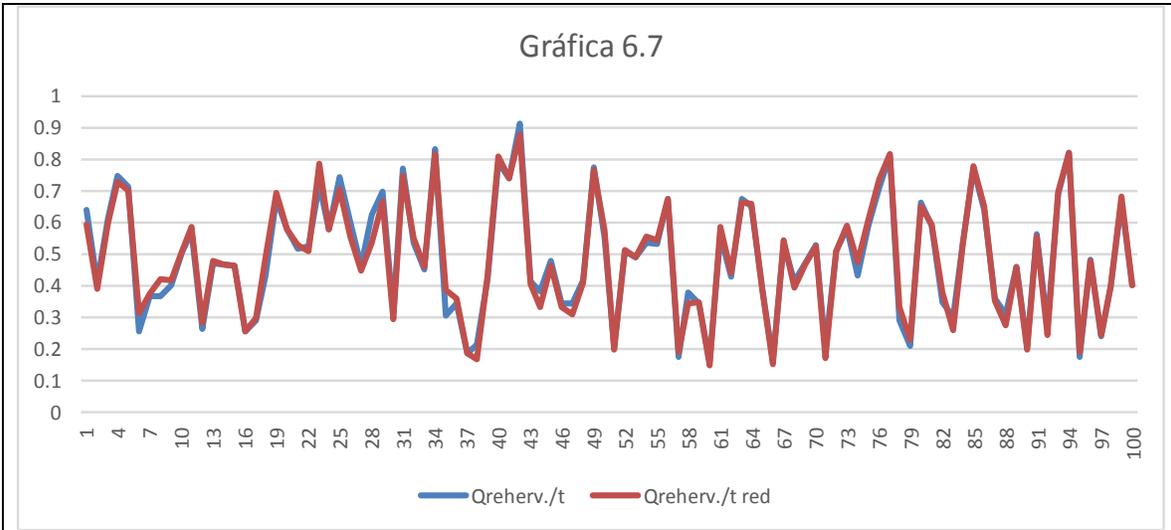
Número de Épocas	Error Absoluto Acumulado de una Época
1	151.0119
10000	2.1980
20000	1.3990
30000	1.0997
40000	1.0320
50000	0.8806
60000	0.8128
70000	0.8565
80000	0.7912
90000	0.8267
100000	0.7875

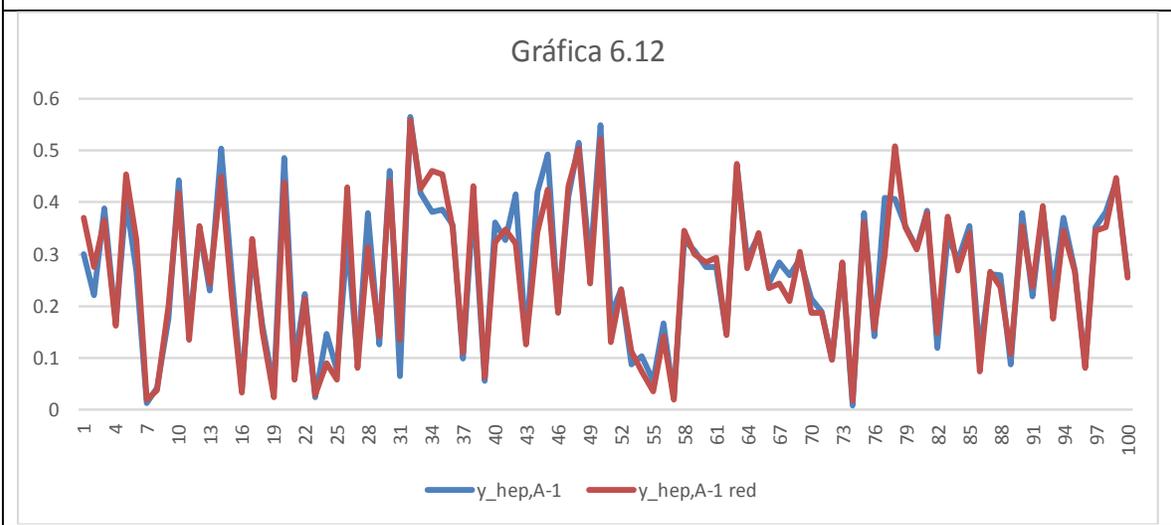
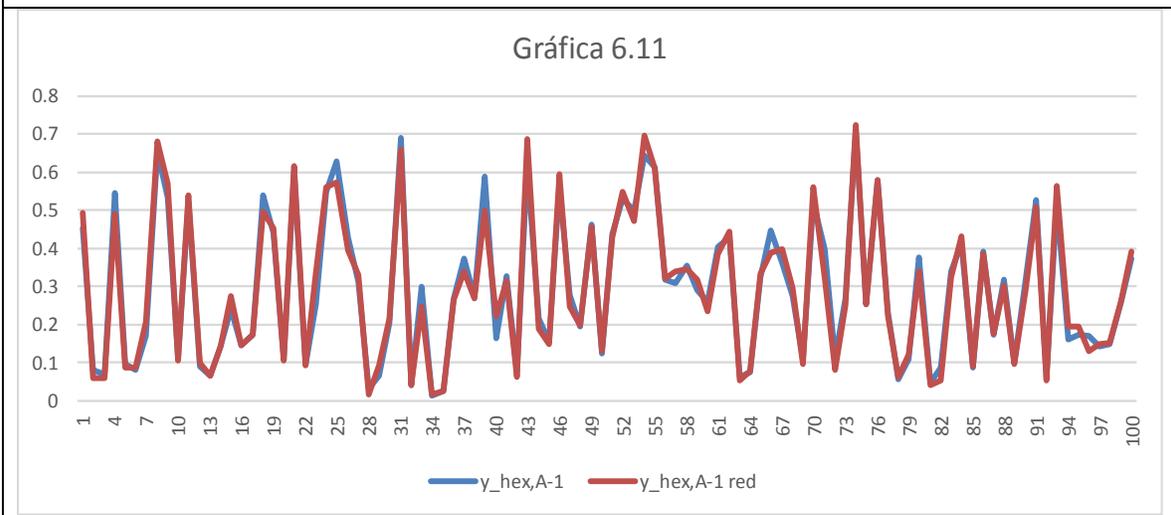
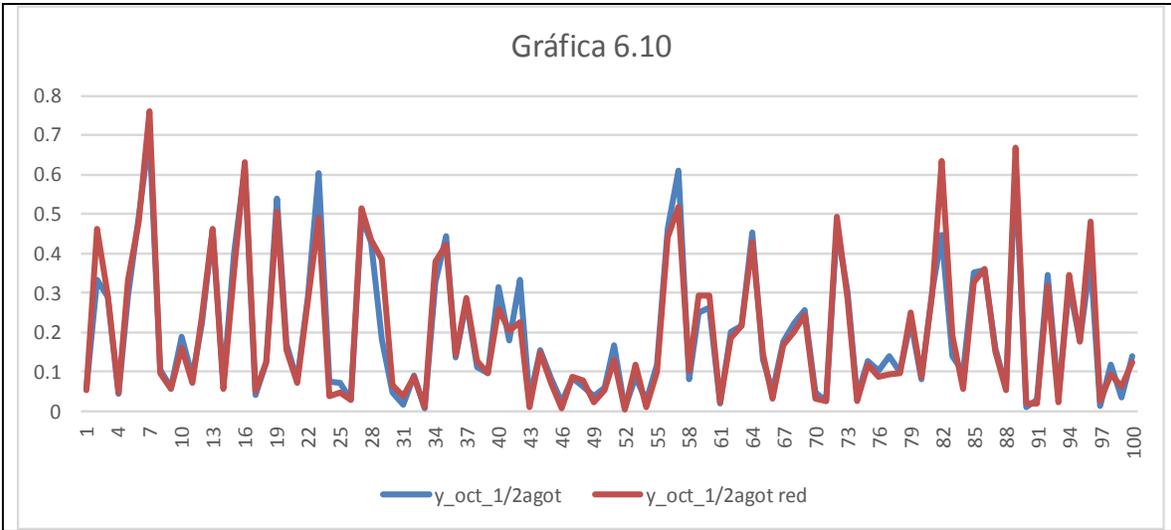
Prueba de la red MLP entrenada

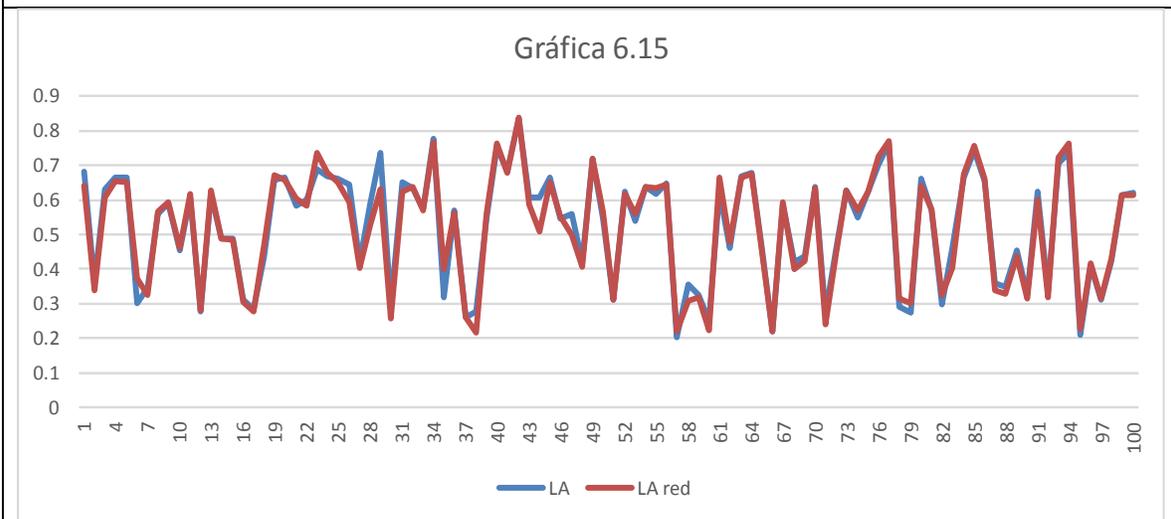
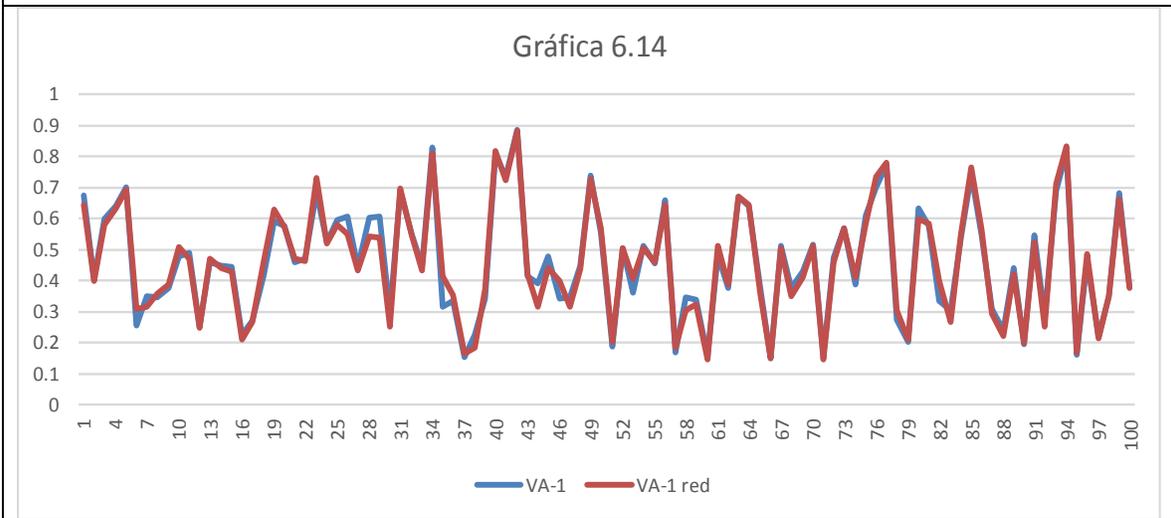
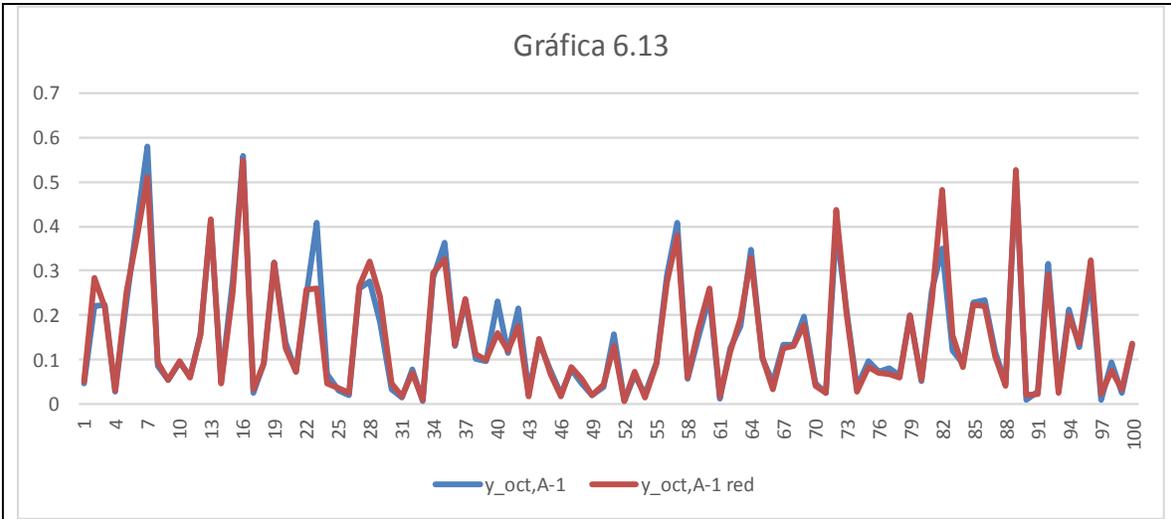
Para probar la red entrenada se elaboró una tabla de prueba de 100 renglones con el programa (4). Se escaló de igual manera que como se hizo con la tabla de entrenamiento para que los valores de todas las variables quedaran en el intervalo [0, 1]. En las gráficas 6.1 a 6.30 se muestran los valores de la tabla de prueba comparados con los valores obtenidos por la red MLP entrenada. En cada una de esas gráficas se muestran cada uno de los cien valores de la variable en cuestión, correspondientes a la tabla de prueba y el valor correspondiente obtenido por la red entrenada.

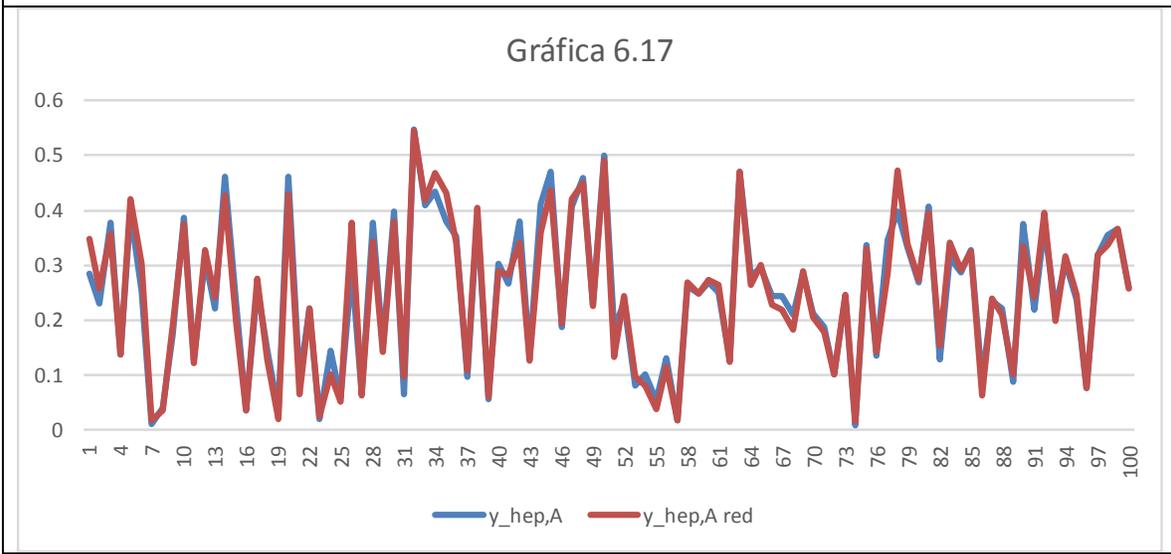
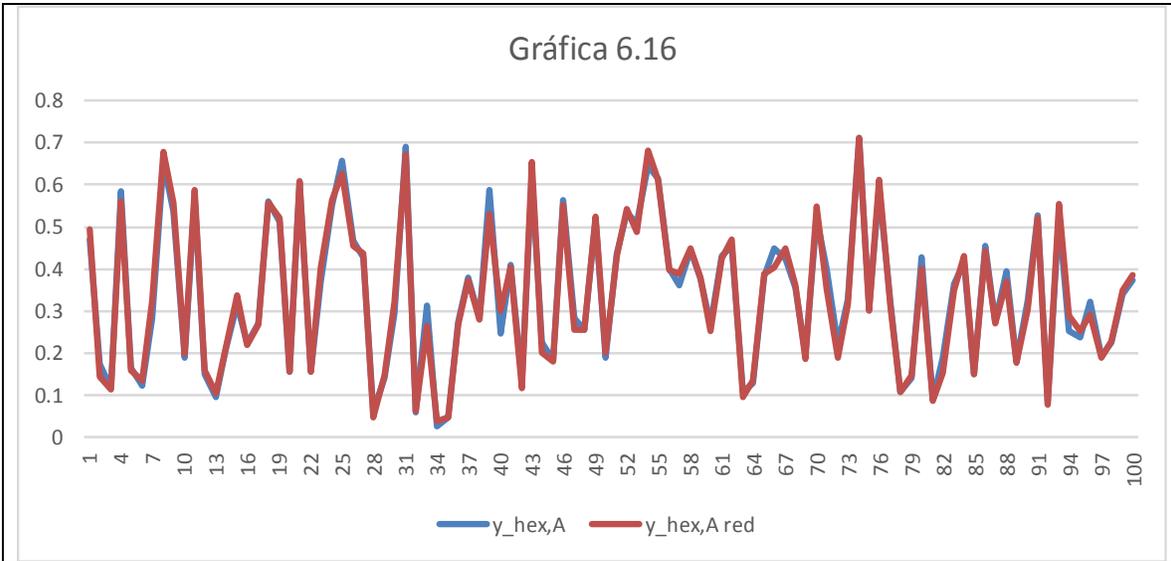


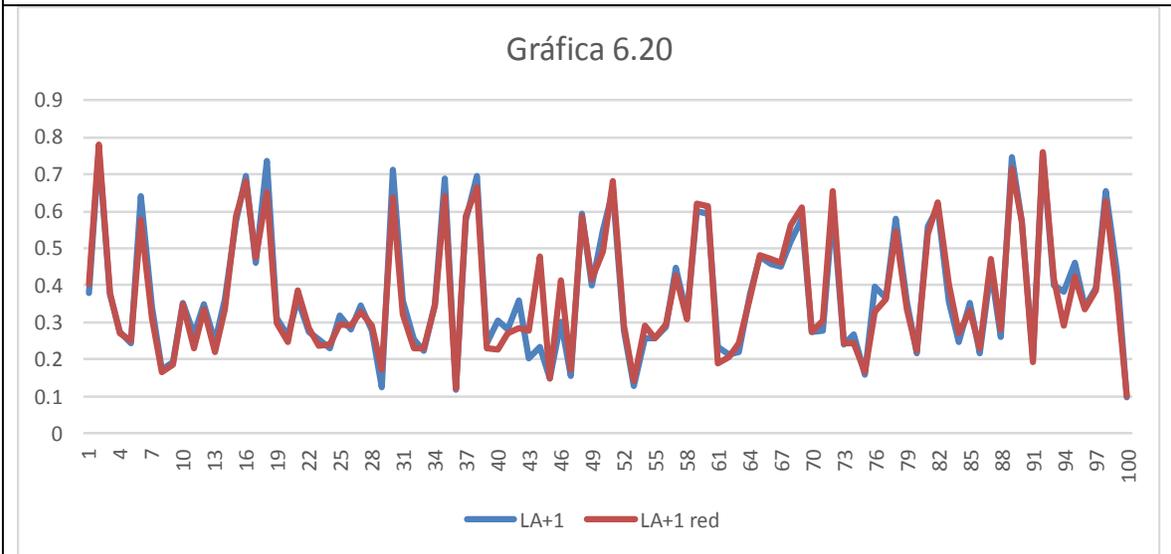
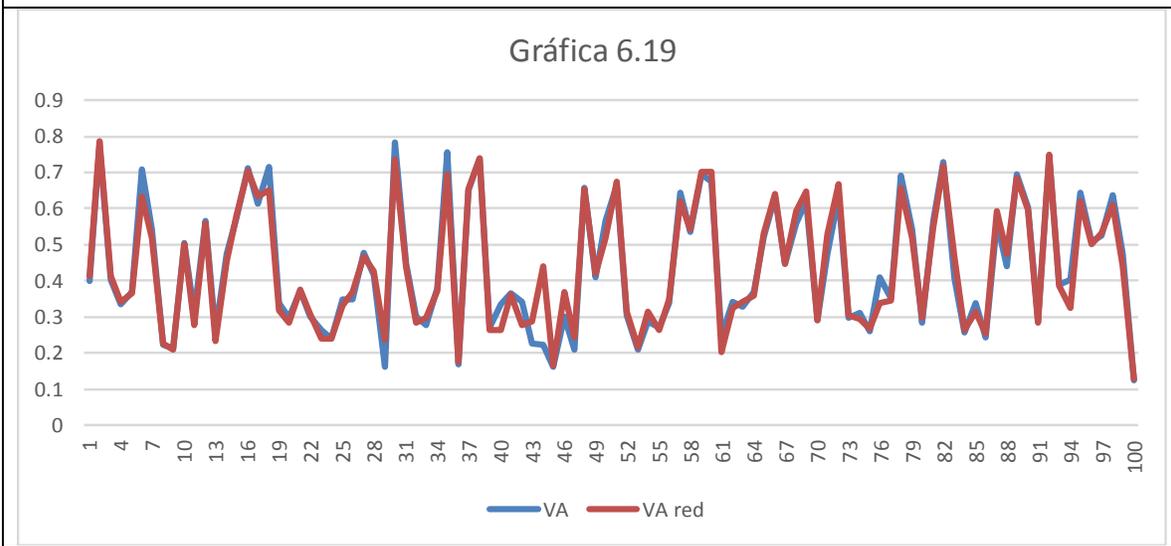
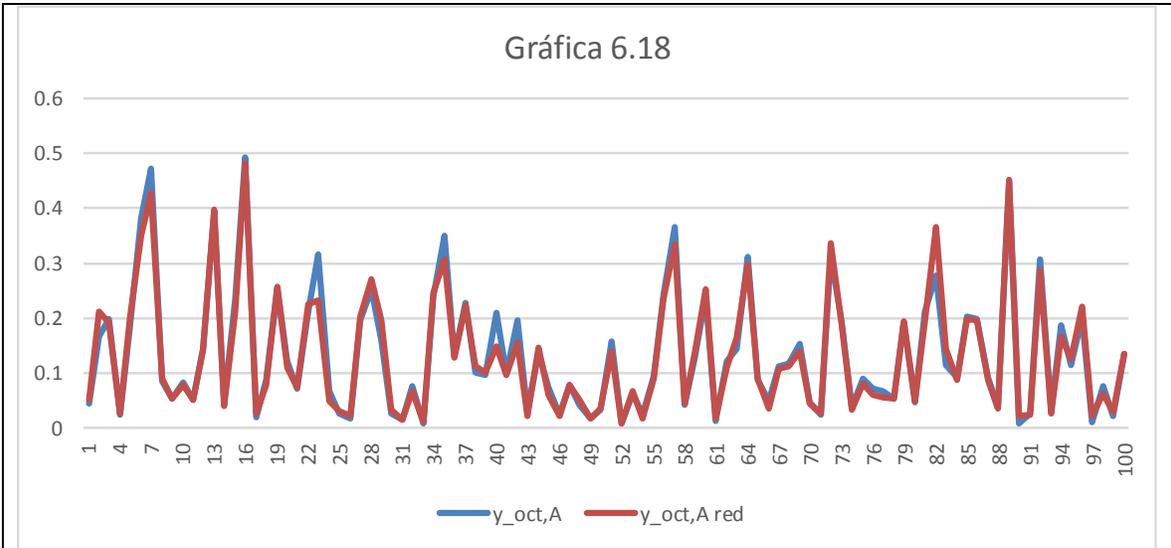


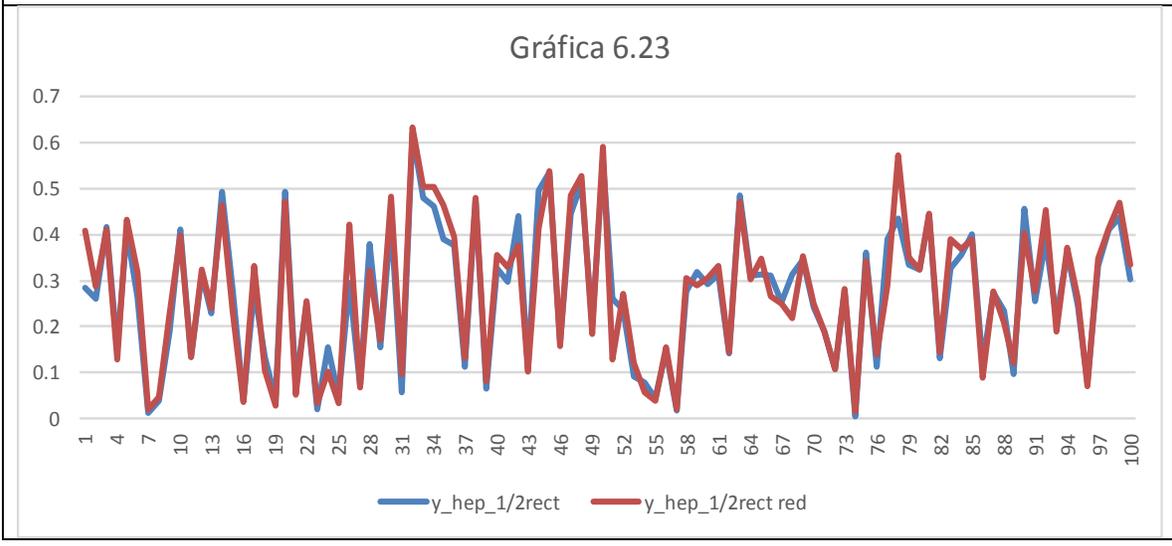
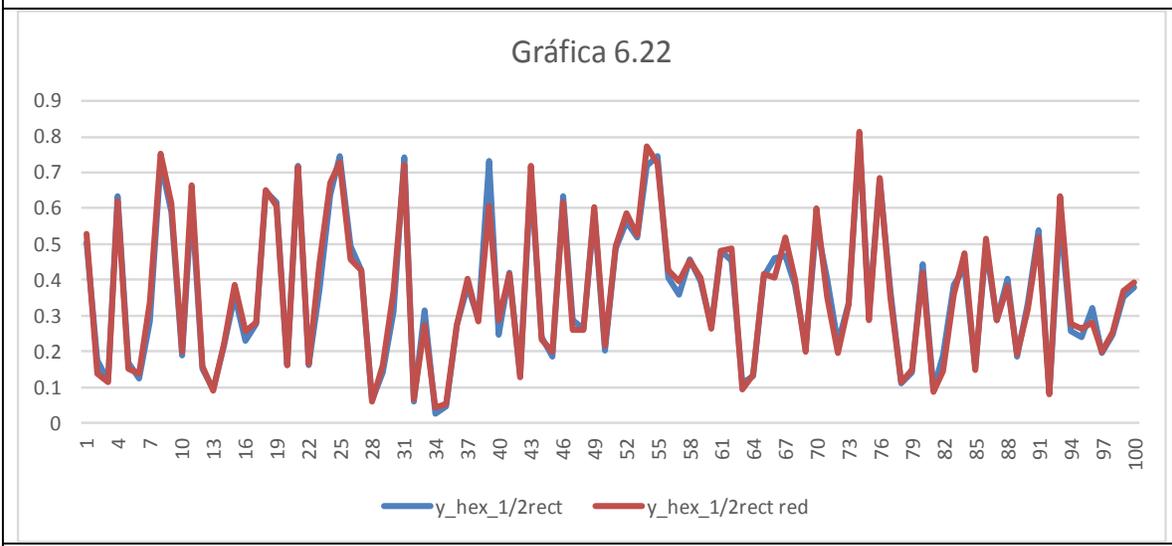
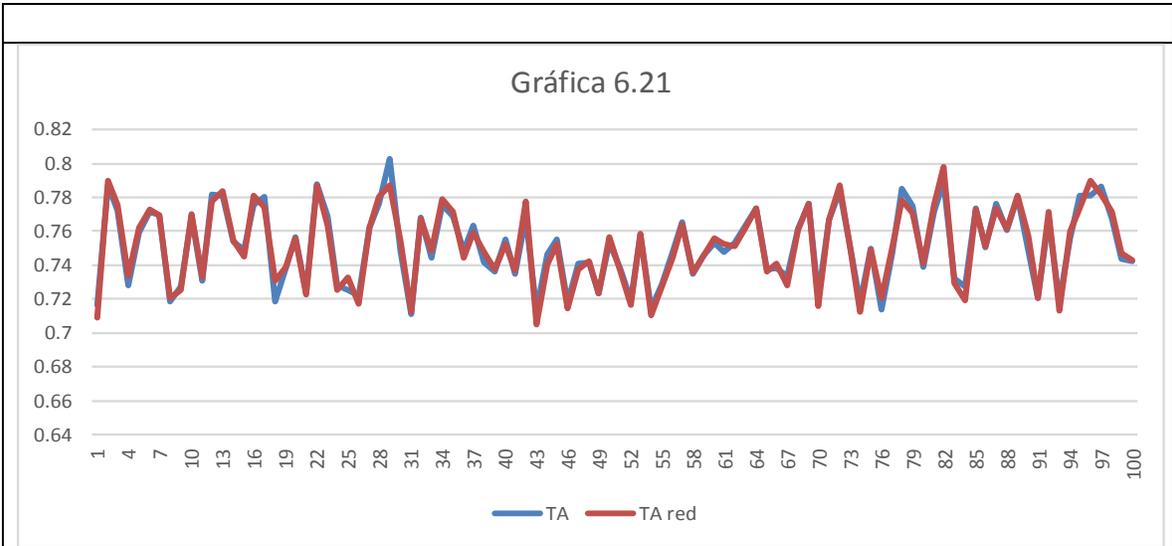


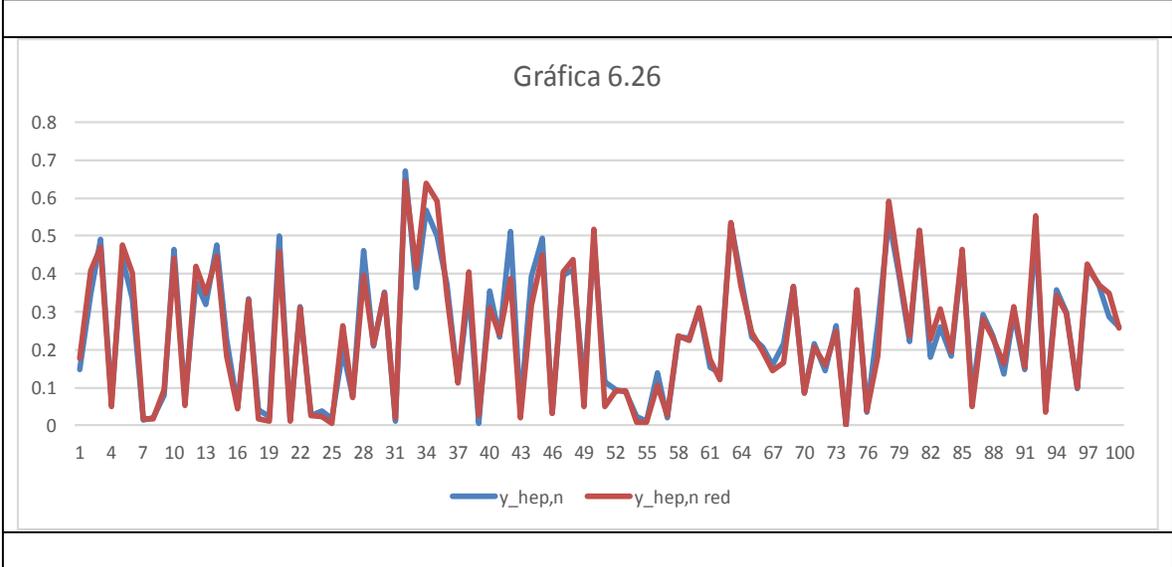
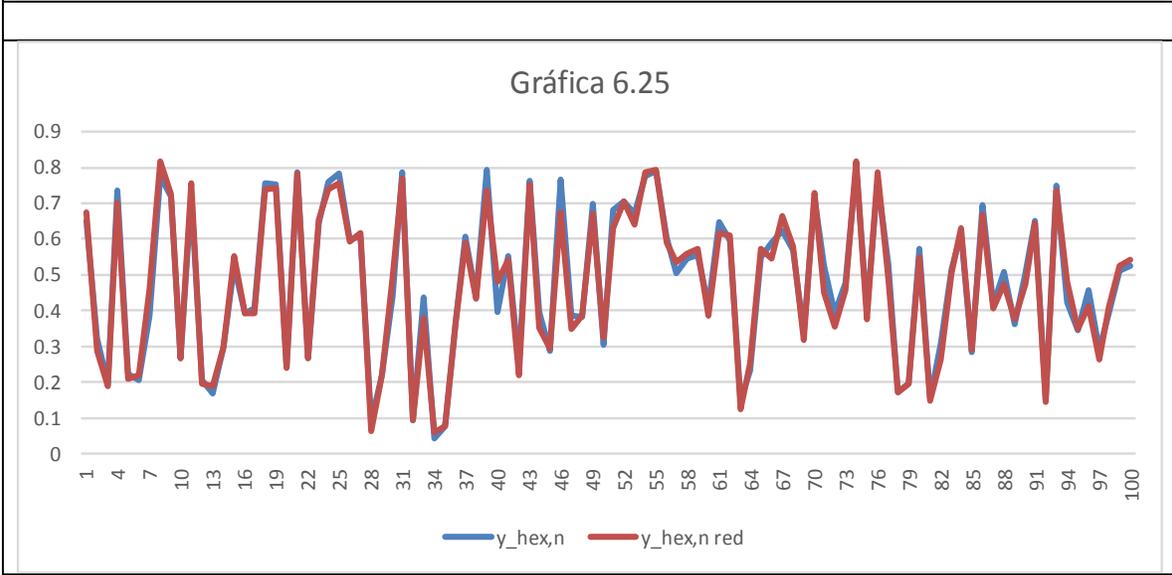
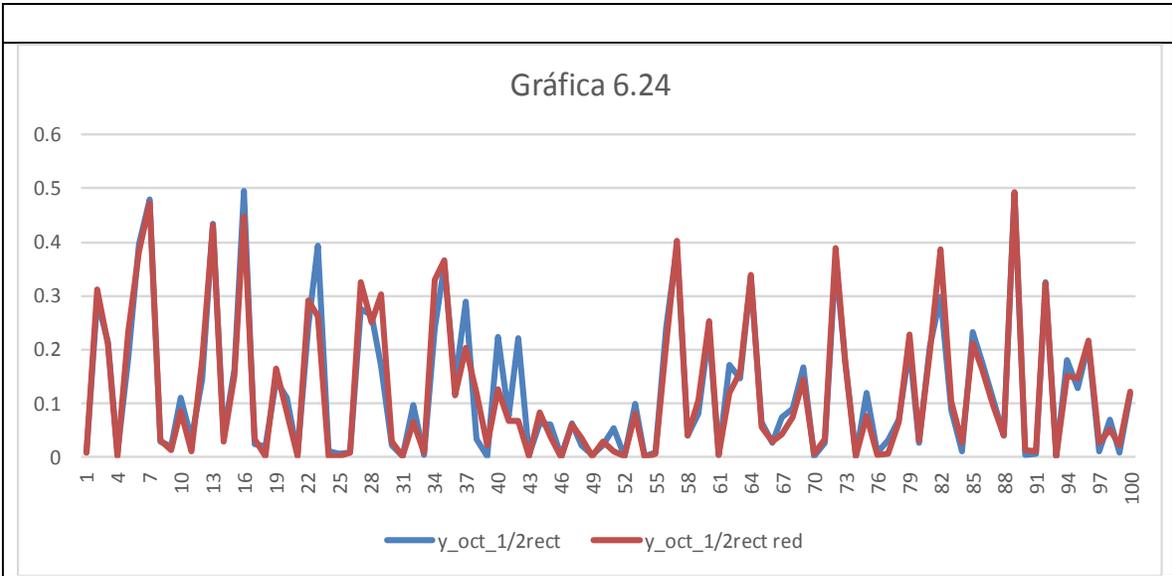


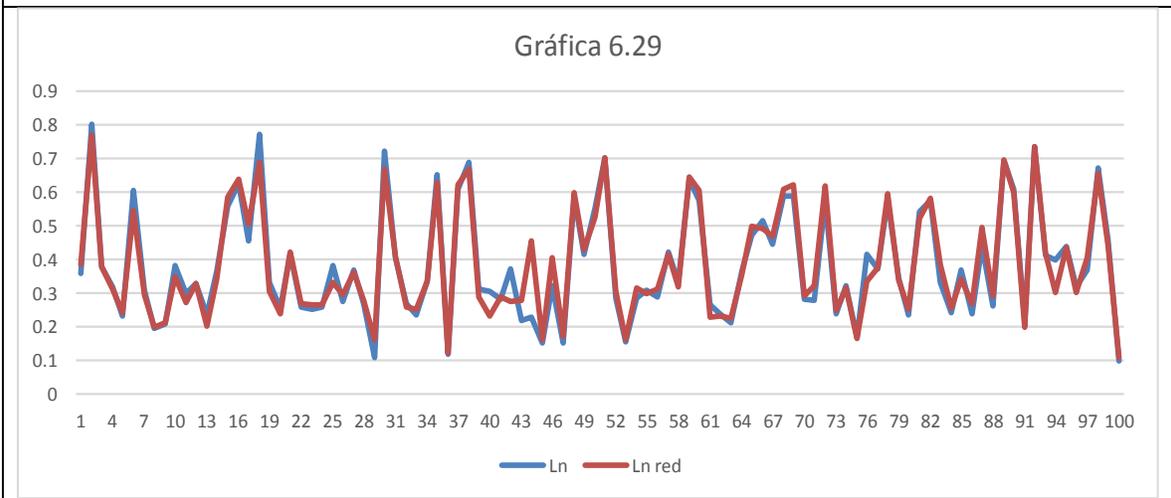
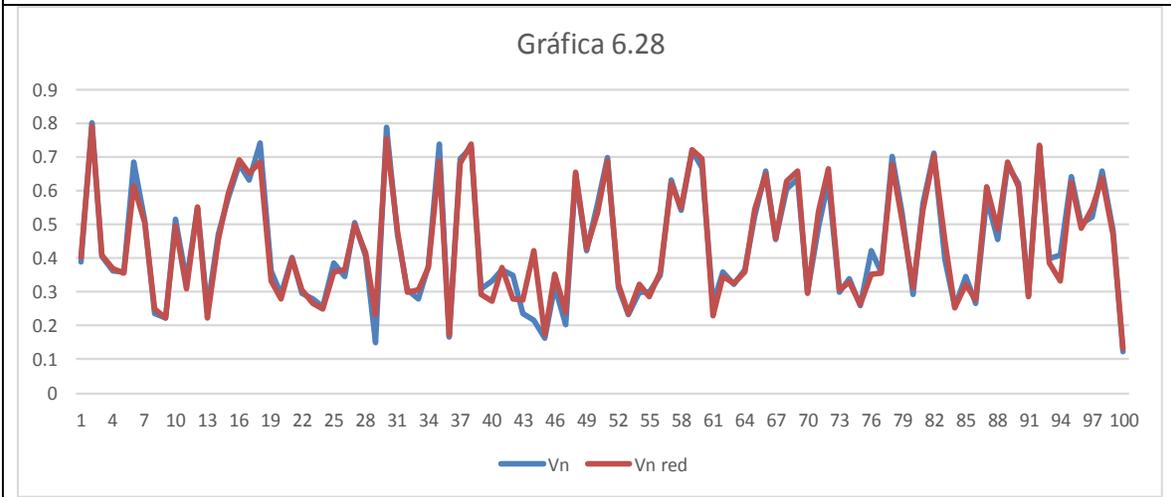
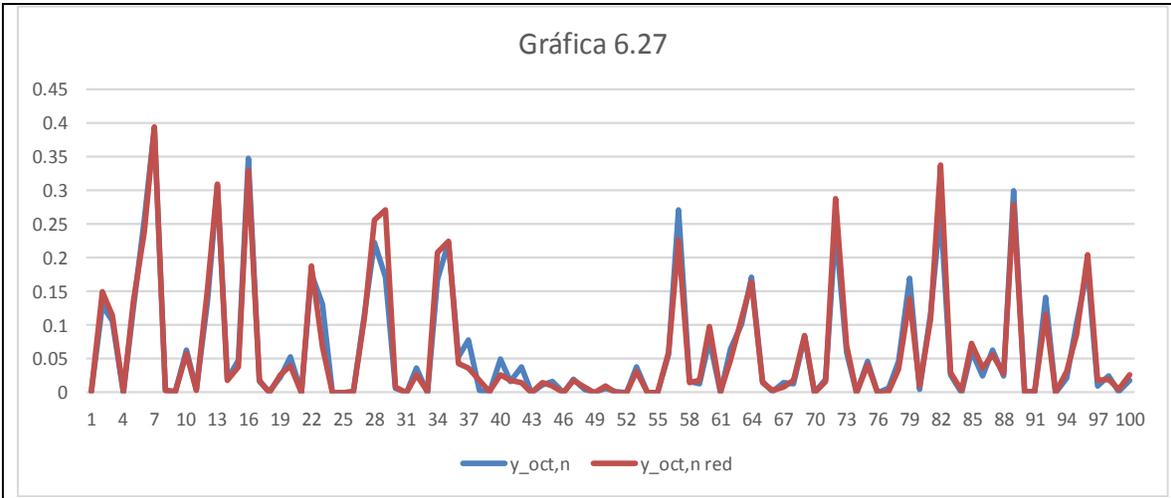


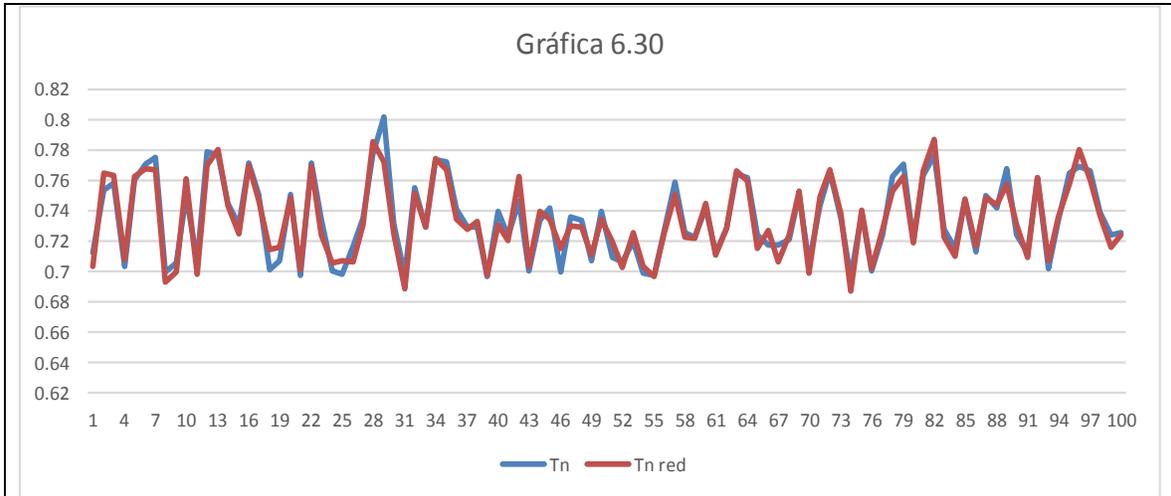












Se puede observar que para cada una de las 30 funciones la red tuvo un buen desempeño. De manera general, la diferencia entre las salidas de red y las variables dependientes de la tabla, no es grande. El promedio de los errores cuadráticos (ver ecuación 4.6) es:

$$E_{prom} = \frac{1}{100} \sum_{i=1}^{m_3} (y_i - y_i^{(3)})^2 = 0.0355$$

Estimando por interpolación los datos de las variables que no son estimadas por la red

Para cada uno de los 100 casos, que corresponden a cada uno de los renglones de la tabla de prueba, la red obtuvo los valores de las variables dependientes que corresponden a los datos de las variables de algunas de las etapas. El programa (5) del que se habla en el capítulo anterior, se usó para estimar los valores de las variables que la red no estimó. Como ya se ha descrito, lo hace usando para algunos variables polinomios de primer orden y para otras, polinomios de segundo orden. En las tablas 6.3 y 6.4 se muestran como ejemplo, dos de los cien casos (renglones de la tabla de prueba); en esas tablas se indica que valores fueron obtenidos por la red MLP y cuales por interpolación.

Tablas 6.3 El caso 5 (renglón no. 5 de la tabla de prueba):

Tabla 6.3a Valores de las variables que especifican la corriente de entrada a la torre (variables independientes) obtenidos al azar por el programa (4)

Variable independiente	Valor obtenido al azar
$Z_{hex,F}$	0.1250
$Z_{hep,F}$	0.3387
$Z_{oct,F}$	0.3151
T_F	390.4014
R_d	0.6081
R_b	1.4594
N	7

Tabla 6.3b Valores del punto inicial estimado para arrancar el método de Newton con derivadas estimadas numéricamente, obtenidos con la red MLP y con el programa interpolador (5)

Variable dependiente*	Valor estimado	Estimado por (r. = red, p.i.= programa interpolador)	Tipo de interpolación (p.= polinomio de primer grado, s. = polinomio de segundo grado)
V 1 (mol/h)	8204.9688	r.	–
V 2 (mol/h)	8197.1165	p.i.	p.
V 3 (mol/h)	8189.2643	r.	–
V 4 (mol/h)	6734.1719	r.	–
V 5 (mol/h)	6792.2723	p.i.	p.
V 6 (mol/h)	6850.3726	p.i.	p.
V 7 (mol/h)	6908.4729	r.	–
L 2 (mol/h)	13911.3783	r.	–
L 3 (mol/h)	13806.0209	p.i.	p.
L 4 (mol/h)	13700.6636	r.	–
L 5 (mol/h)	2509.4154	r.	–
L 6 (mol/h)	2590.7879	p.i.	p.
L 7 (mol/h)	2672.1604	r.	–
y(1, 1) (mol/mol)	0.0166	r.	–
y(2, 1) (mol/mol)	0.3948	r.	–
y(3, 1) (mol/mol)	0.4649	r.	–
y(1, 2) (mol/mol)	0.0479	r.	–
y(2, 2) (mol/mol)	0.5116	r.	–
y(3, 2) (mol/mol)	0.4165	r.	–
y(1, 3) (mol/mol)	0.1099	r.	–
y(2, 3) (mol/mol)	0.5679	r.	–

y(3, 3) (mol/mol)	0.3180	r.	–
y(1, 4) (mol/mol)	0.1974	r.	–
y(2, 4) (mol/mol)	0.5250	r.	–
y(3, 4) (mol/mol)	0.2510	r.	–
y(1, 5) (mol/mol)	0.1863	r.	–
y(2, 5) (mol/mol)	0.5287	r.	–
y(3, 5) (mol/mol)	0.2850	r.	–
y(1, 6) (mol/mol)	0.2034	p.i.	s.
y(2, 6) (mol/mol)	0.5425	p.i.	s.
y(3, 6) (mol/mol)	0.2541	p.i.	s.
y(1, 7) (mol/mol)	0.2540	r.	–
y(2, 7) (mol/mol)	0.5809	r.	–
y(3, 7) (mol/mol)	0.1651	r.	–
T(1) (°K)	426.9644	r.	–
T(2) (°K)	421.1931	p.i.	p.
T(3) (°K)	415.4217	p.i.	p.
T(4) (°K)	409.6504	r.	–
T(5) (°K)	408.3085	p.i.	p.
T(6) (°K)	406.9666	p.i.	p.
T(7) (°K)	405.6246	r.	–
Q/t reherv (J/h)	267382014.5105	r.	–

*En los rótulos de las variables de esta tabla, para las fracciones molares, y, el primer número entre paréntesis corresponde al componente: 1 = hex, 2 = hep, 3 = oct; el segundo número corresponde a la etapa

Tablas 6.4 El caso 12 (renglón no. 12 de la tabla de prueba):

Tabla 6.4a Valores de las variables que especifican la corriente de entrada a la torre (variables independientes) obtenidos al azar por el programa (4)

Variable independiente	Valor obtenido al azar
$Z_{hex,F}$	0.1764
$Z_{hep,F}$	0.3378
$Z_{oct,F}$	0.1801
T_F	419.8189
R_d	0.5554
R_f	0.9801
N	23

Tabla 6.4b Valores del punto inicial estimado para arrancar el método de Newton con las derivadas estimadas numéricamente, obtenidos con la red MLP y con el programa interpolador (5)

Variable dependiente*	Valor estimado	Estimado por (r. = red, p.i.= programa interpolador)	Tipo de interpolación (p.= polinomio de primer grado, s. = polinomio de segundo grado)
V 1 (mol/h)	3066.7475	r.	–
V 2 (mol/h)	3053.3193	p.i.	p.
V 3 (mol/h)	3039.8911	p.i	p.
V 4 (mol/h)	3026.4629	p.i	p.
V 5 (mol/h)	3013.0347	p.i	p.
V 6 (mol/h)	2999.6065	p.i	p.
V 7 (mol/h)	2986.1783	p.i	p.
V 8 (mol/h)	2972.7501	p.i	p.
V 9 (mol/h)	2959.3219	p.i	p.
V 10 (mol/h)	2945.8937	p.i	p.
V 11 (mol/h)	2932.4655	r.	–
V 12 (mol/h)	10285.9219	r.	–
V 13 (mol/h)	10329.0666	p.i	p.
V 14 (mol/h)	10372.2112	p.i	p.
V 15 (mol/h)	10415.3558	p.i	p.
V 16 (mol/h)	10458.5004	p.i	p.
V 17 (mol/h)	10501.6451	p.i	p.
V 18 (mol/h)	10544.7897	p.i	p.
V 19 (mol/h)	10587.9343	p.i	p.
V 20 (mol/h)	10631.0789	p.i	p.
V 21 (mol/h)	10674.2235	p.i	p.
V 22 (mol/h)	10717.3682	p.i	p.
V 23 (mol/h)	10760.5128	r.	–
L 2 (mol/h)	6305.3152	r.	–
L 3 (mol/h)	6267.2850	p.i	p.
L 4 (mol/h)	6229.2548	p.i	p.
L 5 (mol/h)	6191.2246	p.i	p.
L 6 (mol/h)	6153.1945	p.i	p.
L 7 (mol/h)	6115.1643	p.i	p.
L 8 (mol/h)	6077.1341	p.i	p.
L 9 (mol/h)	6039.1039	p.i	p.
L 10 (mol/h)	6001.0737	p.i	p.
L 11 (mol/h)	5963.0435	p.i	p.
L 12 (mol/h)	5925.0133	r.	–
L 13 (mol/h)	3333.0330	r.	–
L 14 (mol/h)	3362.5834	p.i	p.
L 15 (mol/h)	3392.1337	p.i	p.

L 16 (mol/h)	3421.6841	p.i	p.
L 17 (mol/h)	3451.2345	p.i	p.
L 18 (mol/h)	3480.7849	p.i	p.
L 19 (mol/h)	3510.3352	p.i	p.
L 20 (mol/h)	3539.8856	p.i	p.
L 21 (mol/h)	3569.4360	p.i	p.
L 22 (mol/h)	3598.9864	p.i	p.
L 23 (mol/h)	3628.5368	r.	–
y(1, 1) (mol/mol)	0.0005	r.	–
y(2, 1) (mol/mol)	0.1301	r.	–
y(3, 1) (mol/mol)	0.3492	r.	–
y(1, 2) (mol/mol)	0.0000	p.i	s.
y(2, 2) (mol/mol)	0.1824	p.i	s.
y(3, 2) (mol/mol)	0.3448	p.i	s.
y(1, 3) (mol/mol)	0.0000	p.i	s.
y(2, 3) (mol/mol)	0.2301	p.i	s.
y(3, 3) (mol/mol)	0.3380	p.i	s.
y(1, 4) (mol/mol)	0.0000	p.i	s.
y(2, 4) (mol/mol)	0.2730	p.i	s.
y(3, 4) (mol/mol)	0.3288	p.i	s.
y(1, 5) (mol/mol)	0.0000	p.i	s.
y(2, 5) (mol/mol)	0.3112	p.i	s.
y(3, 5) (mol/mol)	0.3170	p.i	s.
y(1, 6) (mol/mol)	0.0054	r.	–
y(2, 6) (mol/mol)	0.3448	r.	–
y(3, 6) (mol/mol)	0.3028	r.	–
y(1, 7) (mol/mol)	0.0202	p.i	s.
y(2, 7) (mol/mol)	0.3736	p.i	s.
y(3, 7) (mol/mol)	0.2862	p.i	s.
y(1, 8) (mol/mol)	0.0396	p.i	s.
y(2, 8) (mol/mol)	0.3977	p.i	s.
y(3, 8) (mol/mol)	0.2670	p.i	s.
y(1, 9) (mol/mol)	0.0636	p.i	s.
y(2, 9) (mol/mol)	0.4171	p.i	s.
y(3, 9) (mol/mol)	0.2454	p.i	s.
y(1, 10) (mol/mol)	0.0921	p.i	s.
y(2, 10) (mol/mol)	0.4319	p.i	s.
y(3, 10) (mol/mol)	0.2213	p.i	s.
y(1, 11) (mol/mol)	0.1253	r.	–
y(2, 11) (mol/mol)	0.4419	r.	–
y(3, 11) (mol/mol)	0.1948	r.	–
y(1, 12) (mol/mol)	0.1987	r.	–
y(2, 12) (mol/mol)	0.4085	r.	–
y(3, 12) (mol/mol)	0.1758	r.	–

y(1, 13) (mol/mol)	0.1957	p.i	s.
y(2, 13) (mol/mol)	0.4002	p.i	s.
y(3, 13) (mol/mol)	0.1914	p.i	s.
y(1, 14) (mol/mol)	0.1941	p.i	s.
y(2, 14) (mol/mol)	0.3956	p.i	s.
y(3, 14) (mol/mol)	0.2042	p.i	s.
y(1, 15) (mol/mol)	0.1940	p.i	s.
y(2, 15) (mol/mol)	0.3949	p.i	s.
y(3, 15) (mol/mol)	0.2141	p.i	s.
y(1, 16) (mol/mol)	0.1953	p.i	s.
y(2, 16) (mol/mol)	0.3979	p.i	s.
y(3, 16) (mol/mol)	0.2211	p.i	s.
y(1, 17) (mol/mol)	0.1980	r.	–
y(2, 17) (mol/mol)	0.4048	r.	–
y(3, 17) (mol/mol)	0.2253	r.	–
y(1, 18) (mol/mol)	0.2021	p.i	s.
y(2, 18) (mol/mol)	0.4154	p.i	s.
y(3, 18) (mol/mol)	0.2266	p.i	s.
y(1, 19) (mol/mol)	0.2077	p.i	s.
y(2, 19) (mol/mol)	0.4299	p.i	s.
y(3, 19) (mol/mol)	0.2251	p.i	s.
y(1, 20) (mol/mol)	0.2147	p.i	s.
y(2, 20) (mol/mol)	0.4482	p.i	s.
y(3, 20) (mol/mol)	0.2207	p.i	s.
y(1, 21) (mol/mol)	0.2231	p.i	s.
y(2, 21) (mol/mol)	0.4702	p.i	s.
y(3, 21) (mol/mol)	0.2134	p.i	s.
y(1, 22) (mol/mol)	0.2329	p.i	s.
y(2, 22) (mol/mol)	0.4961	p.i	s.
y(3, 22) (mol/mol)	0.2033	p.i	s.
y(1, 23) (mol/mol)	0.2442	r.	–
y(2, 23) (mol/mol)	0.5257	r.	–
y(3, 23) (mol/mol)	0.1904	r.	–
T(1) (K)	442.0964	r.	–
T(2) (K)	439.9358	p.i	p.
T(3) (K)	437.7752	p.i	p.
T(4) (K)	435.6146	p.i	p.
T(5) (K)	433.4539	p.i	p.
T(6) (K)	431.2933	p.i	p.
T(7) (K)	429.1327	p.i	p.
T(8) (K)	426.9721	p.i	p.
T(9) (K)	424.8115	p.i	p.
T(10) (K)	422.6509	p.i	p.
T(11) (K)	420.4903	p.i	p.

T(12) (K)	418.3297	r.	–
T(13) (K)	417.7108	p.i	p.
T(14) (K)	417.0918	p.i	p.
T(15) (K)	416.4729	p.i	p.
T(16) (K)	415.8540	p.i	p.
T(17) (K)	415.2351	p.i	p.
T(18) (K)	414.6162	p.i	p.
T(19) (K)	413.9973	p.i	p.
T(20) (K)	413.3784	p.i	p.
T(21) (K)	412.7595	p.i	p.
T(22) (K)	412.1406	p.i	p.
T(23) (K)	411.5217	r.	–
Q/t reherv (J/h)	108180116.5337	r.	–

*En los rótulos de las variables de esta tabla, para las fracciones molares, y, el primer número entre paréntesis corresponde al componente: 1 = hex, 2 = hep, 3 = oct; el segundo número corresponde a la etapa

Usando el método de Newton con las derivadas estimadas numéricamente para calcular los valores de las variables de todas las etapas

Para cada uno de los cien casos (renglones de la tabla de prueba), se obtuvieron las estimaciones del punto inicial, como se ha descrito en las secciones de arriba, que requiere el método casi newtoniano descrito en el capítulo 3. Con esas estimaciones se arrancó el método casi newtoniano y se corrió para los cien casos. Los resultados obtenidos se muestran en la tabla 6.5.

Tabla 6.5 Resultados de las corridas del método de Newton con las derivadas estimadas numéricamente para los cien casos de la tabla de prueba. La tolerancia usada fue de 0.0003

Caso (renglón de la tabla de prueba)	Número de etapas	Convergió	Número de iteraciones usadas por el método casi newtoniano
1	13	sí	3
2	21	sí	74
3	11	sí	4
4	11	sí	4
5	7	sí	3
6	13	sí	4
7	11	sí	3
8	9	sí	4
9	11	sí	3
10	21	sí	97
11	7	sí	4
12	23	sí	37
13	15	sí	4

14	7	sí	4
15	9	sí	4
16	7	sí	4
17	13	sí	4
18	11	sí	4
19	9	sí	4
20	7	sí	3
21	15	sí	5
22	9	sí	4
23	17	sí	66
24	15	sí	5
25	9	sí	5
26	9	sí	3
27	23	sí	16
28	21	sí	42
29	7	sí	4
30	7	sí	4
31	19	sí	5
32	17	sí	7
33	19	sí	19
34	7	sí	3
35	17	sí	5
36	15	sí	3
37	17	sí	6
38	21	sí	6
39	23	sí	8
40	23	sí	13
41	11	sí	3
42	23	sí	21
43	13	sí	4
44	13	sí	4
45	11	sí	4
46	21	sí	7
47	13	sí	3
48	13	sí	3
49	9	sí	4
50	9	sí	4
51	15	sí	6
52	17	sí	5
53	15	sí	6
54	13	sí	4
55	13	sí	4
56	15	sí	4
57	19	sí	14
58	7	sí	4
59	13	sí	4
60	15	sí	4
61	13	sí	4
62	17	sí	44

63	7	sí	3
64	13	sí	3
65	7	sí	3
66	19	sí	6
67	7	sí	3
68	13	sí	5
69	9	sí	4
70	21	sí	4
71	19	sí	7
72	11	sí	4
73	13	sí	4
74	17	sí	9
75	21	sí	7
76	13	sí	4
77	9	sí	4
78	13	sí	4
79	19	sí	4
80	13	sí	4
81	7	sí	3
82	19	sí	11
83	7	sí	3
84	21	sí	5
85	17	sí	4
86	11	sí	4
87	9	sí	4
88	7	sí	4
89	13	sí	3
90	15	sí	5
91	21	sí	8
92	21	sí	13
93	19	sí	6
94	23	no	36 (falló por entrar a los negativos)
95	13	sí	4
96	17	no	9 (falló por entrar a los negativos)
97	13	sí	4
98	7	sí	4
99	15	sí	5
100	21	sí	4

De los 100 casos que se corrieron con el método Newton con las derivadas estimadas numéricamente, en 98 de ellos convergió hacia la solución, es decir, pudo reproducir los valores de las incógnitas hallados por el programa (1) (que fue el que se usó para construir las tablas de entrenamiento y de prueba), lo que indica, que, para los intervalos probados, la red entrenada, combinada con el programa interpolador proporcionaron buenos puntos iniciales para ese método.

En la tabla 6.6 se muestra el número de iteraciones promedio que le tomó al método casi newtoniano para alcanzar la convergencia para las diferentes etapas.

Se puede observar en la tabla 6.6 que entre 7 y 15 etapas, el método de Newton con las derivadas estimadas numéricamente usó en promedio menos de cinco iteraciones para alcanzar la convergencia, mientras que entre 17 y 23 usó entre 9 y 24.27 iteraciones, esto sugiere que para torres con un número de etapas mayor a 23 esta estrategia puede encontrar mayores dificultades.

Tabla 6.6 Número de iteraciones promedio que usó el método de Newton con las derivadas calculadas numéricamente para alcanzar la convergencia para los diferentes números de etapas.

No. de etapas	No. de iteraciones promedio
7	3.50
9	4.00
11	3.67
13	3.80
15	4.70
17	18.25
19	9.00
21	24.27
23	19.00

Conclusiones

Se pudo desarrollar y utilizar, para experimentar, todos los programas que se planeó. La estrategia de entrenar una red MLP que estimara los valores de las variables de las etapas primera, última, de alimentación y algunos otros datos de las etapas inmediatas, combinada con la obtención de las otras variables usando interpolaciones con polinomios de primero y segundo grados, funcionó bien para generar un punto inicial para que un método casi newtoniano pudiera calcular los valores de las corrientes internas de una torre dentro de los intervalos de valores escogidos y con la limitación de fijar el plato de alimentación a la mitad de la torre.

Trabajo Futuro

Hay que probar los casos en los que la etapa de alimentación pueda estar en cualquier otro plato. Es necesario obtener los resultados para torres con un mayor número de etapas a los probados aquí. Se debe evaluar que tanto influyen los valores estimados por la red MLP y que tanto los obtenidos por la interpolación, en la convergencia hacia la solución.

Apéndice

Se muestran los resultados de una corrida para una torre de 7 etapas de equilibrio. Son los resultados correspondientes al caso 5 de la tabla 6.5.

Datos:

Valores de las variables que especifican la corriente de entrada a la torre (variables independientes) obtenidos al azar por el programa (4)

Variable independiente	Valor obtenido al azar
$Z_{hex,F}$	0.1250
$Z_{hep,F}$	0.3387
$Z_{oct,F}$	0.3151
T_F	390.40
R_d	0.6081
R_f	1.4594
N	7

Se usó la red neuronal entrenada y el programa de interpolación para obtener el punto inicial del programa que usa el método de Newton con las derivadas estimadas numéricamente. El punto inicial obtenido fue:

V 1	0.82050	L 5	0.2509
V 2	0.81971	L 6	0.2591
V 3	0.81893	L 7	0.2672
V 4	0.67342	y 1, 1	0.0166
V 5	0.67923	y 2, 1	0.3948
V 6	0.68504	y 3, 1	0.4649
V 7	0.69085	y 1, 2	0.0479
L 2	1.39114	y 2, 2	0.5116
L 3	1.38060	y 3, 2	0.4165
L 4	1.37007	y 1, 3	0.1099

y 2, 3	0.5679	y 1, 7	0.2540
y 3, 3	0.3180	y 2, 7	0.5809
y 1, 4	0.1974	y 3, 7	0.1651
y 2, 4	0.5250	T 1	426.9644
y 3, 4	0.2510	T 2	421.1931
y 1, 5	0.1863	T 3	415.4217
y 2, 5	0.5287	T 4	409.6504
y 3, 5	0.2850	T 5	408.3085
y 1, 6	0.2034	T 6	406.9666
y 2, 6	0.5425	T 7	405.6246
y 3, 6	0.2541	Q/t reherov	26738.2015

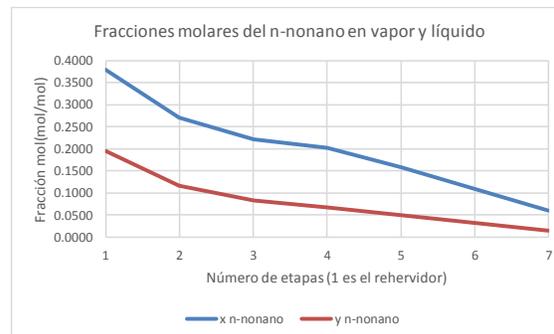
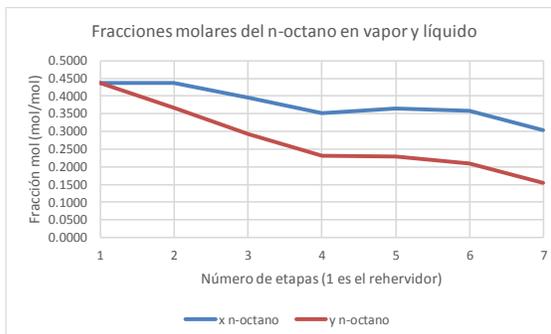
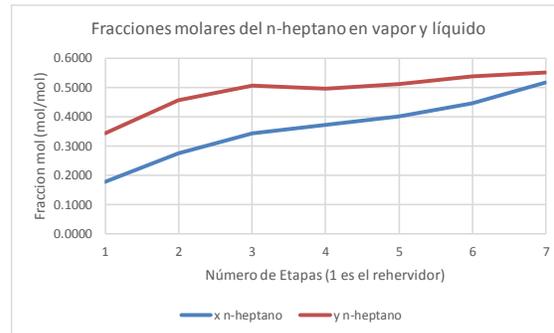
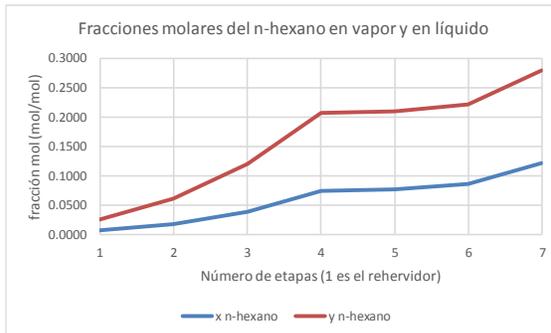
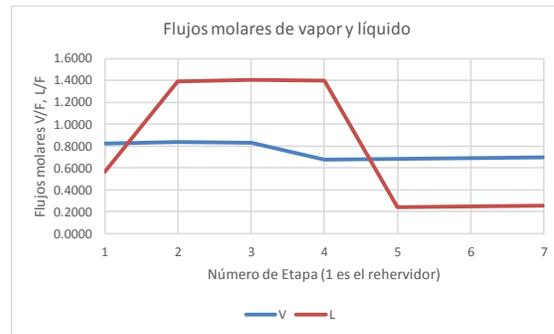
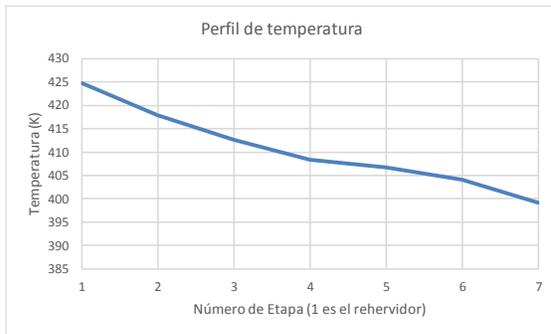
Los resultados obtenidos fueron:

No.de Etapa	V/F	L/F	x n-hexano	y n-hexano	x n-heptano	y n-heptano
1	0.8271	0.5667*	0.0066	0.0257	0.1771	0.3418
2	0.8364	1.3938	0.0179	0.0611	0.2749	0.4556
3	0.8296	1.4032	0.0391	0.1199	0.3431	0.5060
4	0.6769	1.3964	0.0739	0.2070	0.3725	0.4960
5	0.6826	0.2437	0.0774	0.2092	0.3999	0.5121
6	0.6895	0.2493	0.0864	0.2212	0.4462	0.5372
7	0.6967	0.2563	0.1219	0.2799	0.5154	0.5500

*L1 es la salida del fondo de la torre, B.

x n-octano	y n-octano	x n-nonano	y n-nonano
0.4374	0.4362	0.3789	0.1962
0.4367	0.3670	0.2705	0.1162
0.3955	0.2914	0.2223	0.0827
0.3507	0.2304	0.2029	0.0666
0.3644	0.2291	0.1584	0.0495
0.3578	0.2099	0.1095	0.0318
0.3026	0.1550	0.0601	0.0150

Graficando estos resultados:



Bibliografía

Caudill, M., Charles Butler (1992). *Understanding Neural Networks: Computer Explorations*, MIT Press, Cambridge, Massachusetts.

Eiben, A.E., J.M. Smith (2015). *Introduction to Evolutionary Computing*, Springer, Berlin.

Elliot, J. Richard, Carl T. Lira (1999). *Introductory chemical engineering thermodynamics*, Prentice-Hall, Upper Saddle River, New Jersey.

Fausett, L. (2003). *Numerical methods algorithms and applications*, Prentice Hall, Upper Saddle River, New Jersey.

Henley, E. J. y J.D. Seader (1990). *Operaciones de separación por etapas de equilibrio en ingeniería química*, Repla, México D.F.

Henseler, J, "Back Propagation", en Braspenning, P.J., F. Thuijsman y A.J.M.M. Weijters (eds.) (1995). *Artificial Neural Networks. An Introduction to ANN Theory and Practice*, Springer, Berlin Heidelberg New York, pp. 37-66.

Hutchinson, I.H. (2015). *A Student's Guide to Numerical Methods*, Cambridge University Press, Cambridge Massachusetts.

Linz, P. and R. Wang (2003). *Exploring numerical methods: an introduction to scientific computing using Matlab*, Jones and Bartlett, Boston.

Lisboa, P.G.J. (ed.), (1992). *Neural networks. Current applications*, Chapman & Hall, Suffolk, Great Britain.

McCracken, D.D. y W.S. Dorn (1979). *Métodos numéricos y programación Fortran*, Limusa, México D.F.

Mujtaba, I.M. and M.A. Hussain (2001). *Application of neural networks and other learning technologies in process engineering*, Imperial College Press, London.

Prausnitz, J. M., C.A. Eckert, R.V. Orye, J.P. O'Connell (1967). *Computer Calculations for Multicomponent Vapor-Liquid Equilibria*, Prentice-Hall, Englewood Cliffs, New Jersey.

Reklaitis, G. V. (1986). *Balances de materia y energía*, Interamericana, México D.F.

Sharma, J. N. (2007). *Numerical methods for engineers and scientists*, 2nd ed., Alpha Science International, Oxford, United Kingdom.

Shepherd, A. J.,(1997). *Second-Order Methods for Neural Networks. Fast and Reliable Training Methods for Multi-Layer Perceptrons*, Springer, London Berlin Heidelberg.

Yun, X., M. Gen (2010). *Introduction to Evolutionary Algorithms*, Springer, London, 2010.