



UNIVERSIDAD AUTÓNOMA DE SAN LUIS POTOSÍ



FACULTAD DE CIENCIAS QUÍMICAS

MAESTRÍA EN CIENCIAS EN INGENIERÍA QUÍMICA

**CONTROL DE MODELO PREDICTIVO (MPC)
NO-LINEAL PARA UNA COLUMNA PETLYUK:
COSIMULACIÓN MATLAB-ASPEN**

TESIS PARA OBTENER EL GRADO DE:
MAESTRO EN CIENCIAS EN INGENIERÍA QUÍMICA

PRESENTA:

I.Q. BERNARDO VALADEZ HERNÁNDEZ

DIRECTOR DE TESIS:

DR. RAÚL GONZÁLEZ GARCÍA

San Luis Potosí, S.L.P., febrero 2022



El Programa de Maestría en Ciencias en Ingeniería Química de la Universidad Autónoma de San Luis Potosí pertenece al Programa Nacional de Posgrados de Calidad (PNPC) del CONACYT, registro 000896, en el Nivel Consolidado.

Número de registro de la beca otorgada por CONACYT: 706330



Control de Modelo Predictivo (MPC) No-lineal para una Columna Petlyuk: Cosimulación Matlab-Aspen por Bernardo Valadez Hernández se distribuye bajo una Licencia Creative Commons Atribución-No Comercial-Sin Derivadas 4.0 Internacional.



UNIVERSIDAD AUTÓNOMA DE SAN LUIS POTOSÍ



FACULTAD DE CIENCIAS QUÍMICAS

MAESTRÍA EN CIENCIAS EN INGENIERÍA QUÍMICA

**CONTROL DE MODELO PREDICTIVO (MPC) NO-
LINEAL PARA UNA COLUMNA PETLYUK:
COSIMULACIÓN MATLAB-ASPEN**

TESIS PARA OBTENER EL GRADO DE:
MAESTRO EN CIENCIAS EN INGENIERÍA QUÍMICA

PRESENTA:

I.Q. BERNARDO VALADEZ HERNÁNDEZ

DIRECTOR DE TESIS:

DR. RAÚL GONZÁLEZ GARCÍA

SINODALES:

DR. JOSÉ FERNANDO OREJEL PAJARITO _____

DR. ERIK CÉSAR HERRERA HERNÁNDEZ _____

DR. PEDRO ANTONIO ALONSO DÁVILA _____

San Luis Potosí, S.L.P., febrero 2022



UASLP
Universidad Autónoma
de San Luis Potosí



FACULTAD DE
CIENCIAS QUÍMICAS

San Luis Potosí S.L.P. México

Febrero, 2022

Comité Académico del Posgrado en Ciencias en Ingeniería Química

Facultad de Ciencias Químicas

Presente:

Por este medio comunicamos que la tesis llevada a cabo por el alumno de Maestría IQ. **Bernardo Valadez Hernández**, titulada “CONTROL DE MODELO PREDICTIVO (MPC) NO-LINEAL PARA UNA COLUMNA PETLYUK: COSIMULACIÓN MATLAB-ASPEN”, ha sido concluida y aprobada por el comité tutorial para iniciar los trámites correspondientes para su titulación, la cual tendrá lugar el próximo día 10 de febrero del 2022 a las 11:00 horas en el auditorio chico de la facultad.

ATENTAMENTE:

Director de tesis

Sinodal externo

Dr. Raúl González García

FCQ/UASLP

Dr. José Fernando Orejel Pajarito

FCQ/UASLP

Sinodal

Sinodal

Dr. Erik César Herrera Hernández

FCQ/UASLP

Dr. Pedro Antonio Alonso Dávila

FCQ/UASLP

www.uaslp.mx

Agradecimientos

- A mi familia por haber sido mi apoyo y por creer en mí a lo largo de esta etapa en mi vida. A mis padres, por su paciencia y esfuerzo, por saber que cuento con su respaldo y por motivarme en mi formación académica. A mis hermanos por estar siempre presentes.
- Al Dr. Raúl González García por su asesoría, su apoyo y por compartirme sus conocimientos, he aprendido mucho de él en estos años.
- Al Consejo Nacional de Ciencia y Tecnología (CONACYT) por el apoyo económico otorgado para la realización del proyecto por medio de la beca con No. 706330

Resumen

En este estudio se utiliza un modelo no-lineal a base de una red neuronal artificial (RNA) en lugar de un modelo lineal, para mejorar la precisión de un controlador por modelo predictivo (MPC). Para validar el sistema de control se compara el desempeño con el de un MPC con modelo lineal. Se obtiene el diseño de dos columnas Petlyuk en Aspen Plus[®], para las mezclas: M1 (n-butano, 2-metil-butano, n-pentano) y M2 (n-butano, n-pentano, n-octano). De cada columna se extraen datos de su dinámica para entrenar la RNA. El entrenamiento se realiza utilizando el método *Stochastic One-Step* (SOS). Se utilizan ocho variables manipulables y ocho variables a controlar. La simulación se hace en Matlab[®] en conjunto con Aspen Plus Dynamics[®]. Los pesos de la función de costo del MPC son ajustados a partir de un plan experimental y una optimización. La sintonización se realiza utilizando el tiempo de estabilidad máximo, el máximo sobretiro y el cuadrado del error total como métricas de rendimiento. Para comparar el desempeño del MPC y el tipo de modelo, a la columna se le dan múltiples perturbaciones y cambios de *setpoint*. Las combinaciones de las perturbaciones están basadas en un plan experimental de 20 casos de estudio. Las RNAs entrenadas logran un ajuste adecuado de los datos dinámicos de cada columna Petlyuk. Son capaces de predecir de manera adecuada el comportamiento dinámico de las columnas. Los resultados del control muestran un mejor rendimiento del MPC no-lineal frente a el MPC lineal por una diferencia significativa. La implementación de un modelo no-lineal aumenta la robustez del sistema de control. El modelo lineal a pesar de tener un rendimiento menor funciona de una manera correcta cumpliendo con los objetivos de control. Ambos controladores con diferente modelo (lineal y no-lineal) logran estabilizar la columna Petlyuk. La metodología presentada en este trabajo se puede aplicar para controlar diferentes procesos en la industria química. La elección del modelo y su complejidad estarán determinados por el tipo de proceso y por la precisión que se requiera. Una RNA se puede utilizar como modelo no-lineal y mejorar el desempeño de un MPC.

Palabras clave: Control de procesos químicos; Control por modelo predictivo; Redes neuronales artificiales; Destilación; Columnas de destilación Petlyuk.

Abstract

In this study, a non-linear model based on an artificial neural network (ANN) is used instead of a linear model to improve the accuracy of a model predictive controller (MPC). The performance is compared with an MPC with a linear model to validate the control system. The design of two Petlyuk columns in Aspen Plus[®] is obtained for the mixtures: M1 (n-butane, 2-methyl-butane, n-pentane) and M2 (n-butane, n-pentane, n-octane). Data of its dynamics are extracted from each column to train the ANN. Training is done using the *Stochastic One-Step* (SOS) method. Eight manipulable variables and eight variables to be controlled are used. The simulation is done in Matlab[®] in conjunction with Aspen Plus Dynamics[®]. The weights of the MPC cost function are adjusted from an experimental plan and by optimization. The tuning is done using the maximum stability time, the maximum overshoot, and the total error square as performance metrics. The column is given multiple perturbations and setpoint changes to compare the performance of the MPC and the model type. The disturbance combinations are based on an experimental plan of 20 case studies. The trained ANNs achieved an adequate fit of the dynamic data of each Petlyuk column. They are capable of adequately predicting the dynamic behavior of the columns. The control results show a better performance of the non-linear MPC versus the linear MPC by a significant difference. The implementation of a nonlinear model increases the robustness of the control system. Despite having lower performance, the linear model works correctly, fulfilling the control objectives. Both controllers with different models (linear and nonlinear) stabilized the Petlyuk column. The methodology presented in this work can be applied to control different processes in the chemical industry. The process type and precision required are determining in the model choice and complexity. An ANN can be used as a nonlinear model and improve the performance of an MPC.

Keywords: Control of chemical processes; Model predictive control; Artificial neural networks; Distillation; Petlyuk distillation columns.

Contenido

Agradecimientos	i
Resumen	ii
Contenido	iv
Lista de Figuras	vii
Lista de Tablas	ix
Glosario de símbolos y acrónimos	x
1 Introducción	1
2 Conceptos y antecedentes	3
2.1 Proceso de destilación	3
2.2 Columnas térmicamente acopladas: columna Petlyuk	4
2.3 Control de las columnas térmicamente acopladas	5
2.4 Control MPC	6
2.5 Funcionamiento del MPC	8
2.6 Control con redes neuronales artificiales	9
2.7 Redes neuronales artificiales	11
2.8 Validación cruzada	13
2.9 Entrenamiento rápido (SOS-Stochastic One-Step)	13
2.10 Número de neuronas en la capa oculta	14
3 Justificación, hipótesis y objetivos	17
3.1 Justificación	17
3.2 Hipótesis	17
3.3 Objetivos	18

3.3.1	Objetivo general	18
3.3.2	Objetivos particulares	18
4	Metodología	19
4.1	Diseño de columnas Petlyuk en Aspen Plus	20
4.2	Diseño del controlador MPC lineal	21
4.3	Modelo no-lineal con RNA	26
4.3.1	Obtención de datos	26
4.3.2	Entrenamiento de la RNA	30
4.4	Diseño del controlador MPC no-lineal	32
4.5	Sintonización de los parámetros del MPC	33
5	Resultados y discusión	38
5.1	Modelo no-lineal de RNA	38
5.1.1	Resultados del entrenamiento	38
5.1.2	Predicción de la RNA	44
5.2	Sintonización de los pesos del MPC	45
5.3	Resultados de la la simulación con MPC	49
5.3.1	MPC lineal	53
5.3.2	MPC no-lineal	54
5.4	Comparación entre los modelos lineal y no-lineal	55
6	Conclusiones	72
6.1	Conclusiones generales	72
6.2	Trabajo futuro	75
	Bibliografía	76
A	Códigos de Matlab	81
A.1	Modelo de RNA	81
A.1.1	Obtención de datos	81
A.1.2	Entrenamiento de la RNA	83
A.2	Códigos MPC no-lineal	88
A.2.1	Datos MPC no-lineal	88
A.2.2	Diseño MPC no-lineal	90
A.2.3	Simulación en Simulink MPC no-lineal	92
A.3	Códigos MPC lineal	94
A.3.1	Modelo lineal y diseño de MPC	94
A.3.2	Datos MPC lineal	98
A.3.3	Simulación en Simulink MPC lineal	99

A.4	Ajuste de los pesos del MPC	101
A.4.1	Sintonización de los pesos	101
A.4.2	Optimización ponderaciones	104

Lista de Figuras

2.1	Columna Petlyuk	5
2.2	Estructura del control por modelo predictivo	7
2.3	Esquema de control por MPC	8
2.4	Concepto general del MPC	9
2.5	Red neuronal artificial	11
2.6	Red neuronal artificial de una sola capa oculta	14
2.7	Entrenamiento de dos superficies de prueba	15
2.8	Coefficiente de determinación vs número de neuronas. a) Superficie A; b) Superficie B	16
4.1	Estructura de la columna Petlyuk en Aspen Plus	24
4.2	Diagrama de bloques de Simulink del MPC lineal	27
4.3	Plan experimental en el espacio de tres variables	28
4.4	Diagrama de bloques de Simulink de la simulación a lazo abierto	29
4.5	Simulación a lazo abierto	30
4.6	Modelo de red neuronal artificial	31
4.7	Diagrama de bloques de Simulink del MPC no-lineal	34
4.8	Plan experimental para factores de tipo mezcla	37
5.1	Comportamiento de la evolución del coeficiente de determinación en la mezcla 1 para las variables Y_1 a Y_4	40
5.2	Comportamiento de la evolución del coeficiente de determinación en la mezcla 1 para las variables Y_5 a Y_8	41
5.3	Comportamiento de la evolución del coeficiente de determinación en la mezcla 2 para las variables Y_1 a Y_4	42
5.4	Comportamiento de la evolución del coeficiente de determinación en la mezcla 2 para las variables Y_5 a Y_8	43
5.5	Predicción de las variables de salida de la mezcla 1 por la RNA ante cambios aleatorios en las variables de entrada	46

5.6	Predicción de las variables de salida de la mezcla 2 por la RNA ante cambios aleatorios en las variables de entrada	47
5.7	Ponderación de las métricas del MPC lineal para la mezcla 1	50
5.8	Ponderación de las métricas del MPC lineal para la mezcla 2	50
5.9	Ponderación de las métricas del MPC no-lineal para la mezcla 1 . .	51
5.10	Ponderación de las métricas del MPC no-lineal para la mezcla 2 . .	51
5.11	Comportamiento dinámico de las variables de entrada y salida del caso 1	57
5.12	Comportamiento dinámico de las variables de entrada y salida del caso 5	58
5.13	Comportamiento dinámico de las variables de entrada y salida del caso 6	59
5.14	Comportamiento dinámico de las variables de entrada y salida del caso 12	60
5.15	Comportamiento dinámico de las variables de entrada y salida del caso 15	61
5.16	Comportamiento dinámico de las variables de entrada y salida del caso 17	62
5.17	Comparación del desempeño de los controladores con modelo lineal y no-lineal en términos del tiempo de estabilidad	66
5.18	Comparación del desempeño de los controladores con modelo lineal y no-lineal en términos del máximo sobretiro	67
5.19	Comparación del desempeño de los controladores con modelo lineal y no-lineal en términos del cuadrado del error	67

Lista de Tablas

4.1	Características del diseño de la columna de destilación para cada mezcla	22
4.2	Lista de variables manipulables y controlables	23
4.3	Código para extraer las matrices del modelo LTI	25
5.1	Resultados de la RNA entrenada por el método SOS para la mezcla 1	39
5.2	Resultados de la RNA entrenada por el método SOS para la mezcla 2	44
5.3	Ponderaciones de las métricas de rendimiento obtenidas para cada controlador	49
5.4	Pesos de la función de costo del MPC correspondientes a las variables de salida	52
5.5	Pesos de la función de costo del MPC correspondientes a los cambios en las variables de entrada	53
5.6	Plan experimental para las perturbaciones	54
5.7	Métricas de rendimiento para todos los casos del modelo lineal . . .	55
5.8	Métricas de rendimiento para todos los casos del modelo no-lineal .	56
5.9	Resultados del análisis de la Prueba t-Student de los casos de estudio para los tres criterios	68
5.10	Diferencias entre los modelos lineal y no-lineal en los tres criterios de rendimiento	69
5.11	Probabilidades obtenidas del ANOVA de los casos estudios para cada métrica	70

Glosario de símbolos y acrónimos

$\underline{\underline{A}}$	Matriz de estado del modelo LTI
$\underline{\underline{B}}$	Matriz de entrada del modelo LTI
$\underline{\underline{C}}$	Matriz de salida del modelo LTI
$\underline{\underline{D}}$	Matriz de transmisión directa del modelo LTI
E_k	Error de la respuesta de la RNA en la iteración k
J	Función de costo del MPC
J_m	Función objetivo para la optimizar las ponderaciones de las métricas de rendimiento
J_U	Término de la función de costo correspondiente a la contribución de las variables de entrada
$J_{\Delta U}$	Término de la función de costo correspondiente a la contribución del cambio en las variables de entrada
J_Y	Término de la función de costo correspondiente a la contribución de las variables de salida
n_U	Número de variables de entrada
n_Y	Número de variables de salida
r^2	Coefficiente de determinación
r_j	<i>Setpoint</i> de la variable de salida j
SSE	Suma de cuadrados del error
SST	Suma de cuadrados total
t	Tiempo [h]
\underline{T}	Vector de datos experimentales para el entrenamiento de la RNA
t_e	Tiempo de entrenamiento [h]
t_i	Tiempo de integración [h]
T_i	Dato experimental i para el entrenamiento de la RNA
t_m	Tiempo de muestreo [h]
\underline{U}	Vector de variables de entrada
U_i	Variable de entrada i

\underline{U}_k	Vector de variables de entrada al tiempo k
\underline{W}^i	Matriz de pesos de la capa i .
$W_{k,j}^i$	Peso asignado a la neurona k en la capa i con la neurona j en la capa $i + 1$
w_j^U	Peso de la función de costo correspondiente a la variable de entrada j
$w_j^{\Delta U}$	Peso de la función de costo correspondiente al cambio en la variable de entrada j
w_j^Y	Peso de la función de costo correspondiente a la variable de salida j
\underline{X}	Vector de variables de estado
$\underline{\dot{X}}$	Vector de derivadas de las variables de estado
x_{CE}	Ponderación correspondiente al cuadrado del error total
X_i	Variable de estado del sistema i
x_{OS}	Ponderación correspondiente al máximo sobretiro
x_{TE}	Ponderación correspondiente al tiempo de estabilidad
\underline{Y}	Vector de variables de salida
\underline{y}^{i+1}	Matriz de respuestas de las neuronas de la capa $i + 1$
Y_i	Variable de salida i
y_j^{i+1}	Respuesta de la neurona j en la capa $i + 1$
\underline{Y}_k	Vector de variables de salida al tiempo k
$\underline{Y}_{k,P}$	Vector de variables de salida con valores predichos al tiempo k
$\underline{Y}_{k,R}$	Vector de variables de salida con valores reales al tiempo k

Símbolos Griegos

$\underline{\delta}^i$	Matriz del error imputado a las neuronas de la capa i
η_k	Parámetro de aceleración de convergencia en la iteración k
θ_j^i	Término constante denominado <i>bias</i>

Acrónimos

ANOVA	Análisis de varianza
<i>bias</i>	sesgo
CDI	<i>Control Design Interface</i>
CE	Cuadrado de error total
CH	Horizonte de control
<i>feed</i>	Alimentación hacia adelante
<i>forward</i>	

IMC	Internal Model Control
LTI	<i>Linear time invariant</i>
LSTM	<i>Long Short-Term Memory</i>
MAIN	Columna de destilación principal
MPC	Model Predictive Control
NARX	<i>Nonlinear Auto-Regressive eXogenous</i>
OS	Sobretiro máximo absoluto
<i>overshoot</i>	Sobretiro
PH	Horizonte de predicción
PREFRAC	Prefraccionador
<i>RNA</i>	Red neuronal artificial
RNR	Red neuronal recurrente
<i>script</i>	Secuencia de comandos
<i>setpoint</i>	Valor deseado de una variable
SOS	<i>Stochastic One-Step</i>
TE	Tiempo de estabilidad máximo

Capítulo 1

Introducción

El proceso de destilación es una de las operaciones unitarias más utilizadas en la industria de los procesos químicos. Las columnas de destilación consumen la mayor parte de la energía en una planta de proceso, por lo que requieren un alto costo de operación y mantenimiento. La optimización de las condiciones de funcionamiento permite mejorar la eficiencia y reducir la cantidad de energía consumida. Con este fin, los estudios científicos se enfocan en obtener conocimientos teóricos que permitan desarrollar nuevos diseños y mejores secuencias de separación.

Las columnas de destilación totalmente acopladas han demostrado ser más eficientes que las secuencias de destilación convencionales, y se consideran una alternativa técnica y económicamente más eficiente. Las columnas Petlyuk son un tipo de columnas completamente acopladas, las cuales incluyen dos pares de corrientes de vapor y líquido acopladas térmicamente. En el caso de mezclas ternarias, la columna Petlyuk, es capaz de separar los tres componentes en un producto superior, uno inferior y uno lateral.

Las interacciones dentro de la estructura de una columna totalmente acoplada dan comportamientos complejos que se pueden caracterizar por modelos no lineales. Debido a esto los problemas de control y operación impiden la aplicación industrial a gran escala. El control predictivo por modelo (MPC - Model Predictive Control) es una opción adecuada para este tipo de columnas. Los controladores MPC son tecnologías innovadoras y eficientes que suelen diseñarse a partir de modelos lineales. A pesar de su rango limitado, el uso de un modelo lineal ofrece buenos resultados,

pero para un sistema complejo puede llegar a ser inexacto. Un MPC también se puede diseñar con modelos no lineales como las redes neuronales artificiales (RNAs), lo cual podría aumentar significativamente la calidad de estos controles. Las RNAs son algoritmos muy potentes con los que se pueden modelar comportamientos muy complejos. Son sistemas no lineales que pueden utilizarse como modelo de la planta, por lo que el control con RNAs podría aumentar los beneficios de una columna Petlyuk. En este estudio se utiliza un modelo no-lineal a base de una RNA, en lugar de un modelo lineal, para mejorar la precisión del controlador MPC. Para validar el sistema se compara el desempeño contra el de un MPC con modelo lineal. Se utiliza Aspen Plus[®] y Matlab[®] como herramientas para diseñar estrategias de control y comprobar el desempeño del MPC con distintos modelos.

Capítulo 2

Conceptos y antecedentes

2.1 Proceso de destilación

El objetivo del proceso de destilación es la separación de los componentes de una mezcla líquida por vaporización y condensación parcial, los componentes más volátiles se concentran en el destilado y los menos volátiles en el fondo. Para que exista la separación se deben formar dos fases: líquido y vapor. Por el equilibrio de fases, cada fase tendrá una composición diferente. El diseño conceptual de los equipos de destilación requiere conocer con exactitud las características del equilibrio líquido-vapor del sistema en cuestión, que permitan determinar el estado de cada fase, como son las composiciones, la temperatura y la presión. En destilación multicomponente es importante considerar la existencia de puntos azeotrópicos y regiones de destilación [30]. Las composiciones de los productos en una columna de destilación son diferentes a la composición de alimentación. El destilado, es rico en el o los componentes más volátiles y es obtenido por la parte superior de la columna. En el fondo se obtienen los componentes más pesados, y dependiendo del diseño de la columna, esta puede tener una o varias salidas laterales por donde se obtiene un componente con una volatilidad intermedia.

La destilación convencional, es una operación unitaria con mayor grado de madurez tecnológica y madurez en el uso, consume aproximadamente el 40% de la energía total del proceso y además trabajan con una eficiencia tan baja como el 5% [23]. La investigación en columnas de destilación se centra en llevar el pro-

ceso hacia un menor consumo de energía y una mayor eficiencia energética. En la literatura se proponen muchos diseños nuevos que permiten separaciones eficientes, especialmente para mezclas con tres o más componentes [19].

2.2 Columnas térmicamente acopladas: columna Petlyuk

La ineficiencia energética de las secuencias de separación convencionales puede atribuirse a la remezcla de componentes intermedios dentro de la columna. La remezcla se puede eliminar utilizando un prefraccionador, donde en una mezcla ternaria el componente de volatilidad intermedia es distribuido uniformemente antes de entrar a la columna principal [9]. Su objetivo es evitar que el componente más pesado salga por encima y evitar que el componente más ligero salga por la parte inferior. También es posible usar los flujos de líquido y vapor para generar un intercambio de energía y obtener un ahorro. Las columnas que utilizan este contacto se conocen como columnas térmicamente acopladas. Una configuración de este tipo son las columnas Petlyuk (Ver Figura 2.1). En una columna Petlyuk se utilizan corrientes de acoplamiento térmico para sustituir el condensador y el reboiler en la columna prefraccionadora. Una columna Petlyuk también se puede simplificar al compactarse en una sola columna que cuente con una pared divisoria. Ambas columnas, la columna Petlyuk y la de pared divisoria son termodinámicamente equivalentes cuando se ignora la transferencia de calor a través de la pared [9]. Chen [7] ha hecho una clasificación de las diferentes configuraciones en columnas de pared divisoria y sus equivalentes en columnas térmicamente acopladas.

Para cuatro componentes la columna térmicamente acoplada correspondiente es conocida como columna Kaibel. Se considera que es un sustituto económico y rentable de las secuencias de destilación convencionales para la destilación cuaternaria. Sin embargo, la aplicación industrial de dicha estructura todavía es dudosa debido a la mayor complejidad y las incertidumbres relacionadas con su construcción y operación [9].

Para la destilación ternaria, la columna Petlyuk ha demostrado ser más eficiente, ahorrando hasta 30% del consumo de energía [4, 11]. Una ventaja extra es

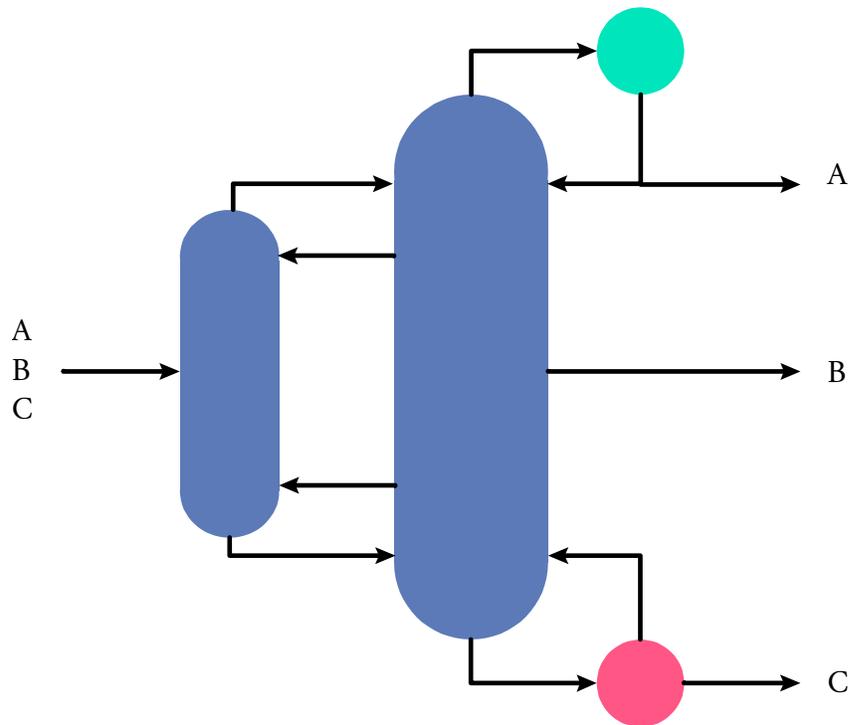


Figura 2.1: Columna Petlyuk

que, en comparación con una columna convencional, se tienen dos grados de libertad adicionales: las corrientes de líquido y vapor entre la columna principal y el prefraccionador [23]. Y al utilizar las columnas equivalentes con pared divisoria el costo se reduce aún más, ya que es como colocar múltiples columnas en una sola estructura [11]. Sin embargo, estas ventajas solo pueden lograrse mediante un diseño óptimo de la columna en estado estacionario y un control dinámico efectivo [9].

2.3 Control de las columnas térmicamente acopladas

La compleja estructura interna que pueden llegar a tener las columnas térmicamente acopladas da como resultado fuertes interacciones y comportamientos altamente no lineales. Por lo que la incertidumbre en el control y operación impiden que la industrialización se lleve a gran escala [31]. En el control de este tipo de columnas,

la forma de manipular las dos variables adicionales (los flujos de líquido y vapor entre el prefraccionador y la columna principal) pueden ser un problema. Por lo general el flujo de líquido se manipula fácilmente y por lo tanto está disponible para fines de control. El flujo de vapor es más complicado porque generalmente no es ajustable, depende de las áreas de sección transversal [21]. Aunque hay propuestas para el control del flujo de vapor, sigue siendo problemático en una columna a escala industrial [16]. Hay algunos trabajos en los que buscan una configuración alternativa más operable, por ejemplo, Cui [9] estudió una configuración con el flujo lateral solo de líquido, ya que es mucho más común en la industria y podría ser más aceptable. Sin embargo, el comportamiento dinámico y la estructura de control aún es complejo, además de que se reducen las ventajas de una columna completamente acoplada.

Muchos estudios han demostrado que se pueden lograr un buen control mediante el uso de estrategias convencionales (control proporcional/integral PI o control proporcional/integral/derivativo PID) y de control avanzado como el MPC [22, 33]. Varios estudios proponen diferentes estructuras de control basadas en control PI y PID, pero no son lo suficientemente robustas para controlar altas perturbaciones [9]. Kiss y Rewagad [33] estudiaron la estructura de control de composición para una columna Petlyuk utilizando control PI y MPC encontraron que las estructuras de control que utilizan MPC son mucho mejores que las que utilizan el control PI. Aunque el control por PID está bien establecido, tiene desventajas al tratar las interacciones no lineales. En comparación el MPC tiene muchas ventajas. Un MPC puede anticipar los valores futuros de las variables controladas y encontrar sus correcciones antes de que sucedan. Las fuertes interacciones y los comportamientos no lineales de la columna Petlyuk hacen del MPC una opción adecuada para la operación del proceso.

2.4 Control MPC

En la teoría de control, el control por modelo interno (IMC por sus siglas en inglés) es una estrategia que simula la respuesta del sistema con el fin de estimar el resultado de una perturbación. El MPC puede considerarse una forma desarrollada del IMC

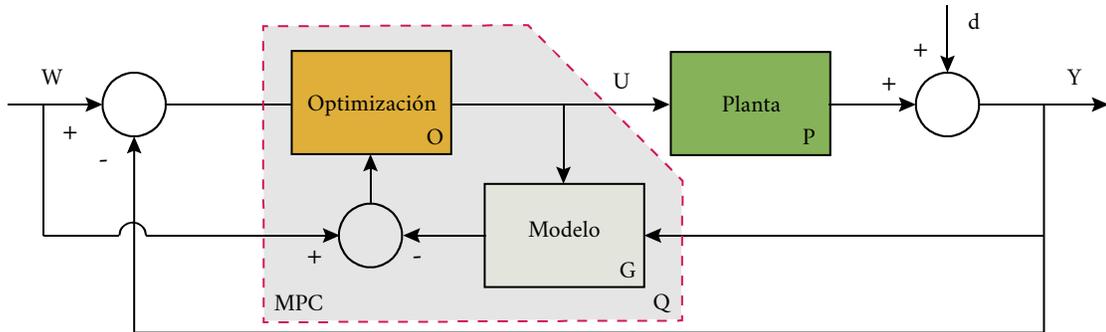


Figura 2.2: Estructura del control por modelo predictivo

[26, 29]. La Figura 2.2 muestra la estructura del control MPC: está compuesta por un controlador, Q , una Planta, P , y un modelo, G . En este caso el controlador es un MPC, cuenta con el modelo de predicción, G , e incluye un proceso de optimización, O , para definir la acción de control.

El controlador MPC es utilizado en la industria de procesos químicos y en refinerías de petróleo [27, 35]. El objetivo del MPC es medir las perturbaciones y compensarlas antes de que la variable controlada se desvíe del punto de control. Hace uso del modelo del proceso para predecir las salidas futuras de la planta y con base en ello optimizar las acciones de control futuras. El control por modelo predictivo se basa en tres operaciones: 1) la predicción de los resultados futuros del proceso, 2) optimización de las salidas y entradas del proceso, 3) ajustes de predicción.

Una desventaja es que el MPC por lo general se basan en el modelo de espacio de estado lineal, lo que podría dar como resultado errores relativamente grandes para un sistema de destilación complejo [31].

Aunque los controles MPC por modelo lineal son muy atractivos y robustos, existen categorías dentro de estos que ofrecen mayores ventajas como lo es el control predictivo no-lineal, basado en RNAs. El control predictivo por RNAs en esencia utiliza el modelo no-lineal de la planta, hecho que lo posiciona en un rango de mayor robustez comparado con el desarrollado por modelo lineal.

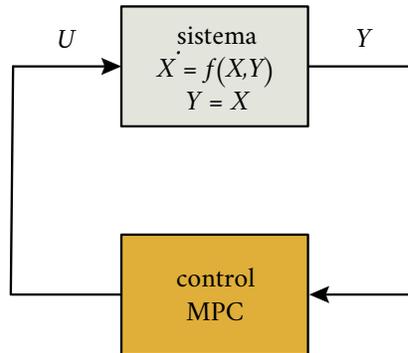


Figura 2.3: Esquema de control por MPC

2.5 Funcionamiento del MPC

El MPC esencialmente realiza un proceso de optimización basado en un modelo del sistema. Ejecuta predicciones de diferentes estrategias de control sobre el modelo. La predicción se hace en un período de tiempo conocido como horizonte de predicción (PH). Esencialmente se determina la próxima acción del controlador basada en la optimización. Una vez que se aplica la acción de control, se avanza el tiempo de predicción y se rehace la optimización para encontrar las próximas entradas de control.

En Figura 2.3 se tiene un esquema del control por MPC. Una señal de control U definida por el controlador, se aplica a un sistema para controlar la variable Y , y hay un *setpoint* al que se debe de llegar o mantenerse en él. A un tiempo k se tiene un estado X en el cual el sistema es iniciado (ver Figura 2.4). El controlador realiza una optimización interna con la que intenta encontrar la mejor entrada U sobre un lapso de tiempo, conocido como horizonte de control (CH), que haga que el sistema se acerque al objetivo. La entrada U encontrada se aplica al sistema para controlarlo. Si el sistema no llegó al objetivo el proceso se repite, y así en cada paso de tiempo. Si hay perturbaciones el controlador las compensa con el tiempo. El controlador, al utilizar un modelo, puede optimizar a valores poco realistas, y estos valores pueden estar fuera de los límites físicos del sistema. Una ventaja es que se pueden establecer límites para las entradas U para evitar ese inconveniente. La optimización es un poco demandante en recursos computacionales, porque se

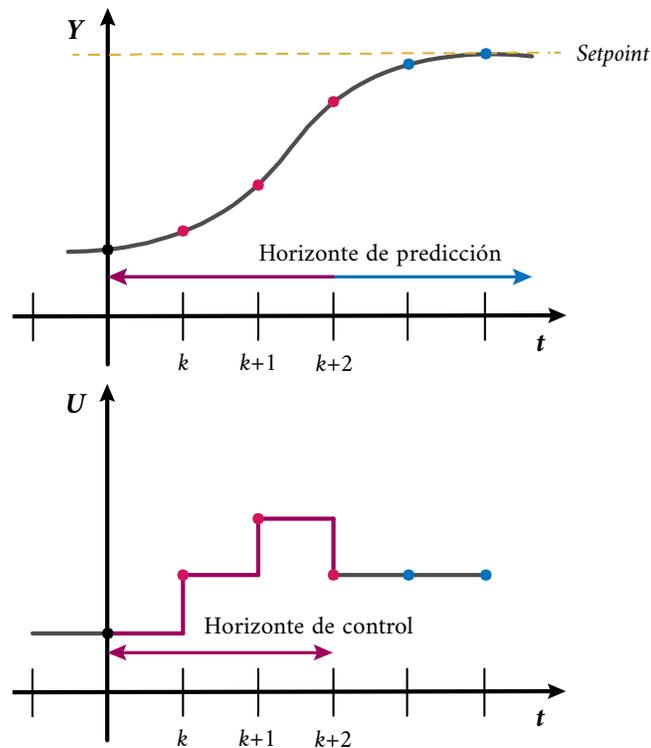


Figura 2.4: Concepto general del MPC

ejecuta en cada paso de tiempo, por eso se suele utilizar ecuaciones del sistema linealizadas, incluso si se tiene un sistema no-lineal. El modelo no-lineal se linealiza sobre un estado en particular y luego se hace una optimización lineal.

2.6 Control con redes neuronales artificiales

Hay dos formas en las que se puede utilizar una RNA en el control de procesos [32]: 1) utilizar una RNA como modelo dinámico e implementarlo en un sistema de control como el MPC. 2) la RNA realiza directamente las acciones de control, pasando por alto el modelo dinámico y la optimización, esta forma requiere una cantidad de datos excesivos y no es competitiva con la primera forma [32]. La forma 1) se suele aplicar comúnmente en el control de procesos químicos [24].

Cuando se realiza la identificación de un sistema por lo general el objetivo es encontrar un modelo lineal [32]. En un MPC el modelo lineal reduce la carga computacional de la simulación iterativa y la optimización [37]. Pero con el equipo actual (más eficiente), cada vez es más posible optimizar directamente un modelo no-lineal. Un MPC requiere un esfuerzo computacional grande. Por lo tanto, un modelo de alta fidelidad difícilmente se puede utilizar debido a la gran carga de cómputo [34]. El controlador se puede volver ineficaz para la optimización en tiempo real de los procesos químicos. Un modelo de RNA en lugar de un modelo linealizado puede aumentar la precisión del control. Se han empleado recientemente RNAs para predecir las respuestas futuras de los sistemas no lineales. Una RNA puede modelar un sistema difícil de desarrollar, por ejemplo, reacciones donde no se cuente con la cinética [37]. Un modelo de RNA puede ser más accesible para la optimización del controlador en tiempo real, además de que su diseño es más sencillo que un modelado completo de los fenómenos físicos [18]. Hay trabajos en los que se ha aplicado esta estrategia, utilizando una RNA como modelo no-lineal en un MPC. Se han desarrollado controladores para reactores químicos [1, 20, 38] y columnas de destilación convencionales [37, 40, 24]. Los resultados son favorables, pero aún se limitan a pocas variables de control o solo una.

Uno de los retos dentro de la identificación de un sistema con una RNA, es que se requiere un conjunto de datos suficientemente grande. La obtención de datos es uno de los pasos más importantes en el modelado y que requieren más tiempo. Otro punto importante es la estructura de la RNA. Las RNAs pueden tener diferentes estructuras dependiendo de la aplicación en la que se utilicen. La estructura de tipo *feed forward* se emplea ampliamente para modelar procesos dinámicos [17]. Dentro de las RNAs utilizadas también se encuentran las redes neuronales recurrentes (RNRs); de tipo: *Long Short-Term Memory* (LSTM); o de tipo: *Nonlinear Auto-Regressive eXogenous* (NARX) [17, 12]. Este tipo de RNAs pueden ajustar a modelos más rigurosos pero sus estructuras son más complejas y por lo tanto más difíciles de entrenar [37]. El método de entrenamiento también es un punto importante que considerar. La RNA debe ser capaz de generalizar. Debe realizar una predicción adecuada ante nuevos datos. En las siguientes secciones se abordará este punto de manera más detallada.

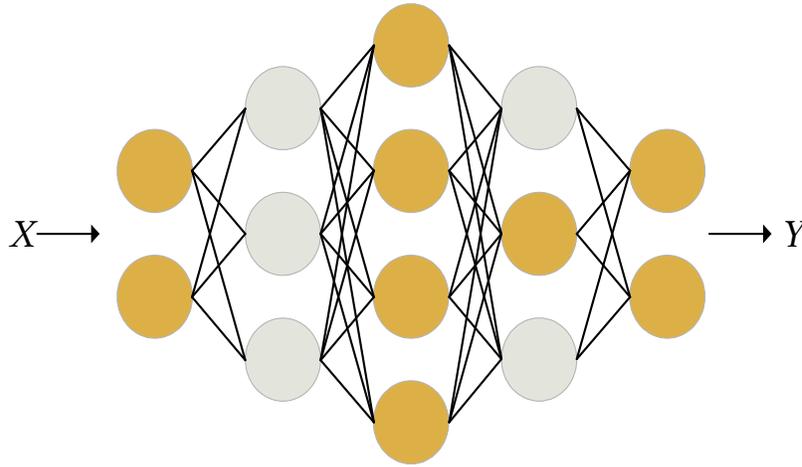


Figura 2.5: Red neuronal artificial

2.7 Redes neuronales artificiales

Una RNA es un modelo matemático no lineal que transforma un conjunto de valores de entrada en uno de salida. Tienen una amplia gama de aplicaciones, se utilizan para la modelación de sistemas o procesos a partir de datos. También en aplicaciones de reconocimiento, agrupación, clasificación y optimización [28, 39]. Están inspiradas en sistemas neuronales biológicos y tienen capacidades para aprender, almacenar y predecir datos [37, 6]. Está constituida por neuronas interconectadas arregladas en capas. Los datos ingresan por la capa de entrada, pasan a través de las capas ocultas y salen por la capa de salida. La Figura 2.5 es la representación de una RNA con 3 capas ocultas, una capa de entrada con dos variables y la capa de salida con dos variables. Es una RNA de tipo *feed forward* o alimentación hacia adelante.

Dentro de una neurona entran muchos valores, que se multiplican por un peso W asignado, los valores se suman y después se aplica una función de activación de acuerdo con la expresión:

$$y_j^{i+1} = g \left(\theta_j^i + \sum_{k=1}^{n_i} y_k^i \cdot W_{k,j}^i \right) \quad (2.1)$$

donde $W_{k,j}^i$ es el peso asignado a la neurona k en la capa i con la neurona j en la

capa $i+1$, θ_j^i es el término constante denominado *offset* o *bias*, y y_j^{i+1} es la respuesta de la neurona j en la capa $i+1$. Los coeficientes $W_{k,j}^i$ y θ_j^i se pueden agrupar en una sola matriz $\underline{\underline{W}}^i$ de cada capa i . Usando la notación matricial, la Ecuación 2.1 puede cambiarse a su forma matricial de acuerdo a:

$$\underline{\underline{y}}^{i+1} = g \left(\left[\begin{array}{c} \underline{\underline{1}} \\ \underline{\underline{y}}^i \end{array} \right]^T \cdot \underline{\underline{W}}^i \right) \quad (2.2)$$

Para obtener los pesos adecuados, que permitan modelar de forma correcta un proceso, la RNA es “entrenada”. Los pesos se obtienen por optimización reduciendo el error entre los datos experimentales T_i y la respuesta que arrojen las neuronas de la última capa, L :

$$E(\underline{\underline{W}}) = \sum_{i=1}^m (T_i - y_i^L)^2 \quad (2.3)$$

En la mayoría de los casos el entrenamiento o la optimización se realiza de forma iterativa con métodos como el descenso del gradiente o gradiente conjugado:

$$\underline{\underline{W}}_{k+1} = \underline{\underline{W}}_k + \eta_k \cdot \underline{\underline{P}}_k \quad (2.4)$$

donde η_k es un parámetro de aceleración de la convergencia, en el método del gradiente $\underline{\underline{P}}_k = -\nabla E_k$ y en el gradiente conjugado $\underline{\underline{P}}_k = -\nabla E_{k-1} + \frac{\nabla E_k \cdot \nabla E_k}{\nabla E_{k-1} \cdot \nabla E_{k-1}} \underline{\underline{P}}_{k-1}$. Para calcular el gradiente por lo general se utiliza la regla delta o *backpropagation*, que se describe a continuación. En la última capa:

$$\underline{\underline{\delta}}^{L-1} = \underline{\underline{T}} - \underline{\underline{y}}^L \quad (2.5)$$

$$\frac{\partial E}{\partial \underline{\underline{W}}^{L-1}} = - \left[\begin{array}{c} \underline{\underline{1}} \\ \underline{\underline{y}}^{L-1} \end{array} \right]^T \cdot \underline{\underline{\delta}}^{L-1} \quad (2.6)$$

en el resto de las capas:

$$\underline{\underline{\delta}}^i = g' \left(\left[\begin{array}{c} \underline{\underline{1}} \\ \underline{\underline{y}}^i \end{array} \right] \cdot \underline{\underline{W}}^i \right) \cdot \left[\underline{\underline{\delta}}^{i+1} \cdot (\underline{\underline{W}}^{i+1})^T \right] \quad (2.7)$$

$$\frac{\partial E}{\partial \underline{\underline{W}}^i} = - \left[\begin{array}{c} \underline{\underline{1}} \\ \underline{\underline{y}}^i \end{array} \right]^T \cdot \underline{\underline{\delta}}^i \quad (2.8)$$

2.8 Validación cruzada

Las RNA son aproximadores universales que pueden llegar a tener un sobre ajuste y modelar el ruido que pueden llegar a tener los datos experimentales. Para evitar esto se hace una validación cruzada donde una porción de los datos se utiliza para entrenar la RNA y el resto para validar la calidad del ajuste. Se calcula el error para ambos grupos de datos, este disminuye con la evolución del entrenamiento. El error de validación llega a tener un valor mínimo y después vuelve a aumentar, es en el punto mínimo donde se detiene el entrenamiento. En caso contrario al seguir entrenando ocurre el sobre ajuste ya que el error de entrenamiento siempre disminuye.

También se puede calcular el coeficiente de determinación y el coeficiente de determinación ajustado para medir la calidad del ajuste. Son parámetros estandarizados y aumentan con la evolución del entrenamiento llegando a tener un valor máximo cuando se calculan con datos de validación. La mejor RNA se obtiene en este punto. Cuando se calculan con datos de entrenamiento el coeficiente de determinación es asintótico a 1.

$$r^2 = 1 - \frac{SSE}{SST} \quad (2.9)$$

donde SSE es la suma de cuadrados del error y SST es la suma de cuadrados total.

2.9 Entrenamiento rápido (SOS-Stochastic One-Step)

Una RNA de una sola capa oculta con suficientes neuronas puede aproximar cualquier conjunto de datos [10]. El entrenamiento rápido, *Stochastic One-Step* (SOS), desarrollado por Cano-Rocha & Gonzalez-García [6] consiste en utilizar una RNA con una sola capa oculta. Esta RNA cuenta con solo dos matrices de pesos y en la Figura 2.6 se encuentra un ejemplo con dos variables de entrada y una de salida. El entrenamiento se basa en obtener la segunda matriz de pesos \underline{W}^2 con una regresión lineal, dejando el valor de los pesos de la primera matriz \underline{W}^1 escogidos aleatoriamente.

El error asociado a la última capa es el de mayor magnitud y disminuye conforme a cada capa previa. Es por ello que, si se optimizan los pesos de la última matriz el

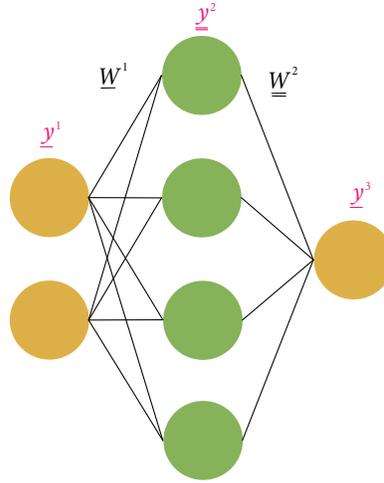


Figura 2.6: Red neuronal artificial de una sola capa oculta

error asociado a las capas previas se aproxima a cero, por lo que solo es necesario optimizar los pesos de la capa final y los pesos de las capas previas se establecen de manera aleatoria. Entrenar una RNA con el método del gradiente toma un tiempo relativamente alto, al entrenar la RNA con el método SOS se mejora la velocidad. Las Ecuaciones 2.10 y 2.11 representan la regresión lineal matricial para optimizar los valores de la matriz de pesos \underline{W}^2 .

$$\underline{A} = \begin{bmatrix} \underline{1} & \underline{y}^2 \end{bmatrix} \quad (2.10)$$

$$\underline{W}^2 = (\underline{A}^T \underline{A})^{-1} \underline{A}^T \underline{T} \quad (2.11)$$

2.10 Número de neuronas en la capa oculta

El número de neuronas óptimo en la capa oculta, se encuentra evaluando el coeficiente de determinación para diferente número de neuronas. Al ser un entrenamiento rápido se pueden evaluar diferentes estructuras y poder encontrar la mejor. La estructura con el número adecuado de neuronas es la que cuente con el valor máximo del coeficiente de determinación para el conjunto de validación. El método es similar al entrenamiento con el método del gradiente y la regla delta, aplicado al número de

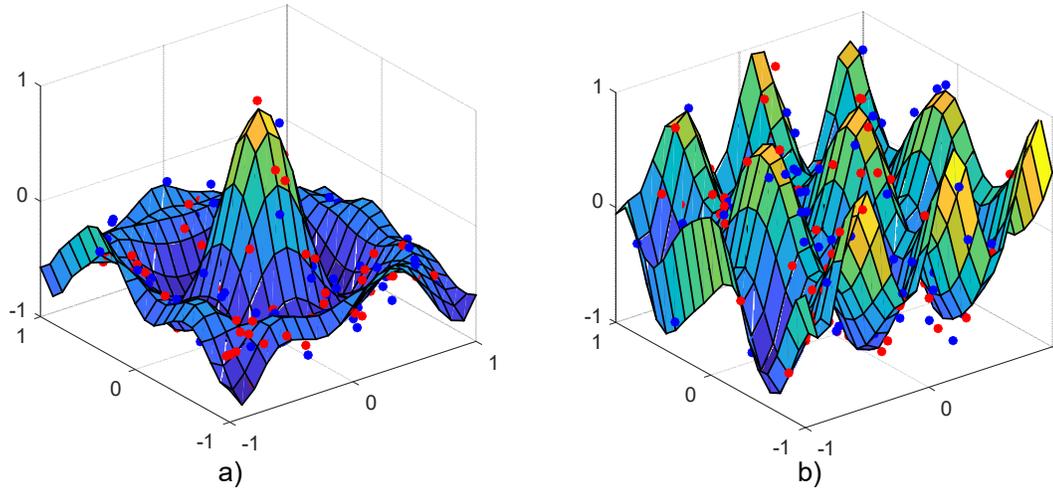


Figura 2.7: Entrenamiento de dos superficies de prueba. a) Superficie de la función gaussiana; b) Superficie de una función trigonométrica

neuronas. En las Figuras 2.7 y 2.8 se muestra un entrenamiento para dos diferentes superficies de prueba. Se utilizan datos de entrenamiento con una cantidad de ruido, representados por los puntos rojos. Los puntos azules corresponden a los datos de validación. La RNA tiene dos respuestas que corresponden a cada superficie. En la Figura 2.8 a) se muestra un máximo en el coeficiente de determinación de validación (línea azul) donde se alcanza el número de neuronas óptimo.

Una RNA con múltiples respuestas puede tener un número de neuronas óptimo para cada respuesta. Para que cada respuesta cuente con el número de neuronas adecuado, la segunda matriz se construye con los pesos óptimos de acuerdo con el número óptimo de neuronas, cada columna tendrá diferente número de valores diferentes de cero. Para que todas tengan el mismo tamaño se completa la columna con ceros, similar a la siguiente matriz:

$$\begin{bmatrix} 2 & 1 \\ 4 & 2 \\ 1 & 0 \\ 1 & 0 \\ 5 & 0 \end{bmatrix}$$

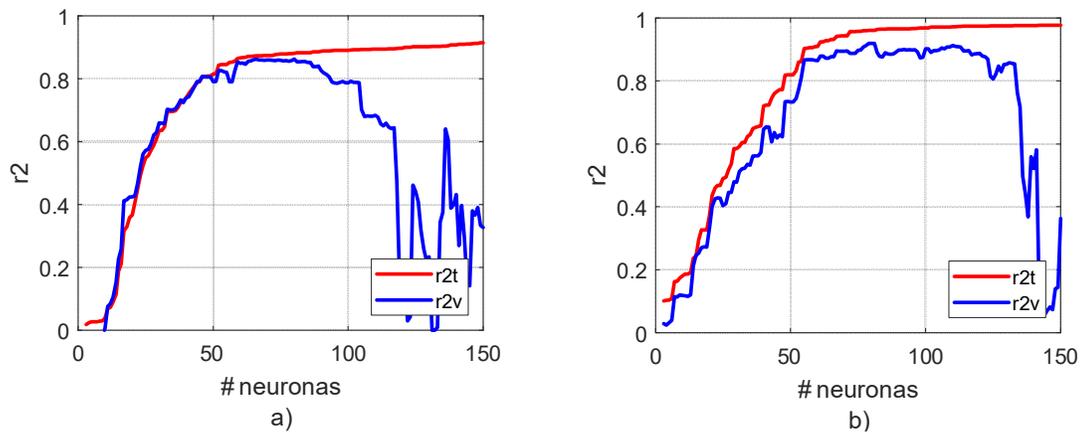


Figura 2.8: Coeficiente de determinación vs número de neuronas. a) Superficie A; b) Superficie B

Al tener el resto de los pesos con cero, se asegura de que cada respuesta tenga el número óptimo de neuronas. La primera matriz de pesos sigue teniendo valores aleatorios. En la Figura 2.8 ambas gráficas tienen un máximo en un lugar diferente por lo que cada superficie requiere un número diferente de neuronas.

Capítulo 3

Justificación, hipótesis y objetivos

3.1 Justificación

Las columnas de destilación tipo Petlyuk son energéticamente más eficientes que las columnas convencionales. Por su naturaleza son más complejas, por lo tanto, requieren un control más estricto. Los controladores MPC han demostrado ser lo suficientemente robustos para controlar procesos complejos. El uso de modelos no lineales como las RNA podría aumentar significativamente la calidad de estos controles. Las RNA son algoritmos muy potentes con los que se pueden modelar comportamientos muy complejos. Se propone utilizar un modelo no-lineal a base de RNA en vez de un modelo lineal para mejorar la precisión del controlador MPC.

3.2 Hipótesis

Un controlador por modelo predictivo a base de una RNA entrenada con el método SOS es capaz de controlar las columnas de destilación Petlyuk en casos de múltiples perturbaciones a la vez.

3.3 Objetivos

3.3.1 Objetivo general

El objetivo general es desarrollar un controlador MPC a base de RNA como modelo no-lineal y comprobar el desempeño ante múltiples perturbaciones simultáneas para una columna de destilación Petlyuk. Las mezclas por separar son:

- M1 (n-butano, 2-metil-butano, n-pentano)
- M2 (n-butano, n-pentano, n-octano)

3.3.2 Objetivos particulares

1. Diseñar en Aspen Plus[®] dos columnas Petlyuk para la separación de las mezclas ternarias M1 y M2.
2. Obtener un RNA capaz de modelar la dinámica de la columna Petlyuk para cada mezcla.
3. Obtener también un modelo lineal de cada columna para poder comparar el desempeño.
4. Diseñar un controlador MPC donde se integre el modelo no-lineal de la RNA y otro donde se utilice el modelo lineal.
5. Sintonizar los parámetros de cada MPC diseñado.
6. Estudiar y analizar diferentes escenarios de control y comparar los resultados de las variables a controlar entre el MPC con el modelo de RNA y el MPC con modelo lineal.

Capítulo 4

Metodología

El objetivo del MPC es compensar las perturbaciones en un proceso antes de que las variables controladas se desvíen de su valor. El MPC hace una predicción de lo que va a suceder en el proceso, y en base a esto realiza las acciones de control. Para realizar la predicción el controlador utiliza un modelo dinámico de la planta, por lo que previo al diseño del controlador es necesario contar con este modelo, el cual puede ser lineal y no-lineal. El diseño de un modelo no-lineal es más complejo, y en este trabajo se busca obtener este modelo utilizando una RNA.

Previo al diseño del controlador se diseñan las columnas Petlyuk en Aspen Plus[®], las columnas en Aspen funcionan como la planta “real” a controlar. El primer paso en el diseño del controlador es obtener la RNA. Es necesario obtener información suficiente del comportamiento dinámico de la planta para poder entrenar la RNA. Esta información se obtiene de las columnas en Aspen. Ya entrenada la RNA funciona como el modelo no-lineal para el controlador. El modelo se implementa en el controlador y posteriormente se realiza su sintonización. El MPC cuenta con una serie de parámetros que deben ser sintonizados para el apropiado funcionamiento del controlador. La sintonización se realiza usando técnicas de optimización.

Una vez se diseña el controlador se somete a prueba con diferentes escenarios de control. Para evaluar el desempeño del controlador no-lineal también se diseña un MPC que utilice un modelo lineal. Se utiliza el modelo lineal invariante en el tiempo o LTI (*Linear time invariant*). Se someten a prueba los dos controladores a

las mismas condiciones y se compara el desempeño de ambos.

4.1 Diseño de columnas Petlyuk en Aspen Plus

La columna para cada mezcla se diseña en Aspen Plus[®] en estado estacionario para dos mezclas diferentes. Las mezclas utilizadas en el presente trabajo son: M1 (n-butano, 2-metil-butano, n-pentano) y M2 (n-butano, n-pentano, n-octano). Los hidrocarburos siguen siendo ampliamente importantes a nivel industrial. La separación física de los componentes de una mezcla suele ser de vital importancia para poder ser utilizados en otro proceso o simplemente para venderlos purificados. Se han reportado la utilización de los componentes de las mezclas elegidas en procesos de isomerización [2]. También son compuesto precursores para diferentes productos: aditivos para combustibles, solventes, polímeros, etc. [25, 13].

Se utiliza el software Aspen Plus[®] para diseñar las dos columnas Petlyuk. El diseño se realiza en estado estacionario y después se exporta a estado dinámico. Se utiliza la metodología reportada por Cano-Rocha & Gonzalez-García [5]. Se inicia con el prediseño utilizando una configuración equivalente, que consta de un arreglo de tres columnas. Las tres columnas se diseñan por métodos cortos utilizando el modelo *DSTWU* de Aspen Plus[®]. La información que se obtiene de los métodos cortos se transfiere a la columna *RadFrac* para utilizar los métodos rigurosos y se confirme que se cumple con las especificaciones deseadas. Para realizar la integración se utiliza el método de la columna balanceada. Las columnas se acoplan en una columna principal, se elimina el reboiler de la columna superior y el condensador de la columna inferior y se igualan los flujos internos ajustando las relaciones de reflujo. A la primera columna a la cual entra la alimentación se le retira el condensador y el reboiler para funcionar como el prefraccionador. Por último, los flujos de interconexión se obtienen mediante balances de materia.

La estructura de la columna con todos sus equipos se muestra en la Figura 4.1. El diagrama cuenta con la columna principal (MAIN), el prefraccionador (PREFRAC), la corriente de alimentación (F), la corriente de destilado (D), la corriente de la salida lateral (S) y la corriente de fondos (B). Entre los equipos principales de encuentran las corrientes de interconexión: las corrientes de vapor superior (VSUP) e inferior

(VINF) y las corrientes de líquidos superior (LSUP) e inferior (LINF). También se cuenta con equipos auxiliares como bombas y válvulas. Los datos de operación y los parámetros de diseño se encuentran en la Tabla 4.1. Los datos de operación de las columnas son extraídos de los resultados del diseño en estado estacionario en Aspen Plus[®]. El modelo termodinámico utilizado es Peng-Robinson, se utiliza este debido a que es un modelo adecuado para el equilibrio líquido-vapor de hidrocarburos. Una vez que se obtienen las columnas, para poder ser controladas, se exportan a Aspen Plus Dynamics[®] para trabajar con el modelo dinámico. Hay dos formas de exportar, *flow driven* y *pressure driven*. En una simulación con *flow driven*, los flujos en las corrientes se manipulan directamente y los cambios en la presión resultan de las acumulaciones debidas a estos flujos, es útil para un primer enfoque del comportamiento dinámico. La simulación con *pressure driven* utiliza las diferencias de presión para definir los flujos de las corrientes, es más riguroso y realista pero también más complejo de configurar, se tienen que definir cambiadores de presión (tuberías, válvulas, bombas, etc.). Se exporta de acuerdo con las variables que se desean controlar y manipular, para el controlador MPC es adecuado exportar como *flow driven*, ya que se busca controlar los flujos. Cuando se exporta de Aspen Plus[®] a Aspen Plus Dynamics[®] se generan controladores por defecto, es necesario eliminarlos para evitar interferencias y que solamente el MPC controle la planta.

Las columnas de destilación contienen muchas variables y un comportamiento altamente no lineal debido a las interacciones. En este trabajo se presentan ocho variables a controlar y ocho variables manipulables. Las variables por manipular y controlar se encuentran en la Tabla 4.2. Estas variables son las más comunes a controlar y manipular en columnas de destilación.

4.2 Diseño del controlador MPC lineal

Para la obtención del modelo lineal, se necesitan datos para realizar el ajuste del modelo. Implica hacer experimentos para obtener las matrices del modelo LTI. Se tiene la ventaja de que Aspen Plus[®] cuenta con las herramientas para realizar este proceso. El modelo lineal se obtiene directamente de Aspen Plus Dynamics[®], con la herramienta *Control Design Interface* (CDI) que linealiza un modelo dinámico en

Tabla 4.1: Características del diseño de la columna de destilación para cada mezcla

	M1	M2
Flujo (kmol/h)	45	45
Fracciones mol A,B,C	0.36, 0.28, 0.36	0.325, 0.335, 0.34
Etapas de alimentación	8	10
Etapas totales de PREFRAC	17	17
Etapas de LSUP y VSUP	7	11
Etapas de VIN y LIN	38	23
Etapas de S	10	20
Relación de reflujo	15.56	3
Etapas totales	66	26
Flujo de destilado D (kmol/h)	15.56	14.05
Fracción másica A	0.9861	0.9999
Fracción másica B	0.0135	3.47×10^{-5}
Flujo en la salida S (kmol/h)	13.31	15.66
Fracción másica A	0.0497	0.02918
Fracción másica B	0.8524	0.9221
Flujo de fondos B (kmol/h)	16.12	15.29
Fracción másica A	2.38×10^{-14}	7.02×10^{-6}
Fracción másica B	0.0754	0.0199

un estado específico. Después de obtener el modelo lineal, en Matlab[®] se adecua reduciendo su orden y con el modelo reducido se crea el controlador. El control por modelo predictivo utiliza parámetros como el horizonte de predicción (PH), el horizonte de control (CH) y los pesos de la función de costo. Es necesario sintonizar estos parámetros para que el controlador pueda mantener el sistema estable con el menor número de oscilaciones.

Para linealizar la planta, es necesario correr un código (*script*) con una serie de comandos de Aspen Custom Modeler, dentro Aspen Plus Dynamics[®]. Los comandos activan la herramienta CDI, reinician las variables del sistema y permiten elegir las variables del modelo y extraer su información. Una vez que se ejecute el código se obtiene las matrices \underline{A} , \underline{B} , \underline{C} , \underline{D} del modelo LTI. La forma del modelo se representa en las Ecuaciones 4.1 y 4.2, las variables de estado \underline{X} dependen de forma lineal de las variables de entrada \underline{U} , las variables de salida \underline{Y} también dependen de

Tabla 4.2: Lista de variables manipulables y controlables

Variables por manipular	
Calor del condensador	QC
Calor del reboiler	QR
Velocidad de reflujo	RF
Flujo del destilado	DF
Flujo de la salida lateral	SF
Flujo de la salida de fondos	BF
Flujo de la interconexión líquida LINF	LINF
Flujo de la interconexión vapor VSUP	VSUP
Variables por controlar	
Presión de MAIN	P1
Nivel de líquido del condensador en MAIN	COND1
Nivel de líquido del reboiler en MAIN	SUMPL1
Fracción másica del componente A en producto D	FMA
Fracción másica del componente B en producto S	FMB
Fracción másica del componente C en producto B	FMC
Presión en PREFRAC	P2
Nivel de líquido de los fondos en PREFRAC	SUMPL2

forma lineal de las variables de estado y entrada.

$$\dot{\underline{X}} = \underline{A}\underline{X} + \underline{B}\underline{U} \quad (4.1)$$

$$\underline{Y} = \underline{C}\underline{X} + \underline{D}\underline{U} \quad (4.2)$$

Los pasos para obtener las matrices y que se ejecute todo de manera correcta son los siguientes:

1. Ejecutar *Dynamic*. Verifica que las columnas se exportaron correctamente y que funcionan de manera adecuada en Aspen Plus Dynamics[®].
2. Ejecutar *Initialization*. Se prueba que funcione el estado inicial.
3. Ejecutar *Steady state*. Se verifica que funcione el sistema en estado estacionario.
4. Quitar los controladores por defecto.
5. Ejecutar *Initialization*. Se vuelve a probar el estado inicial.
6. Ejecutar *Steady state*. Se verifica de nuevo el estado estacionario.

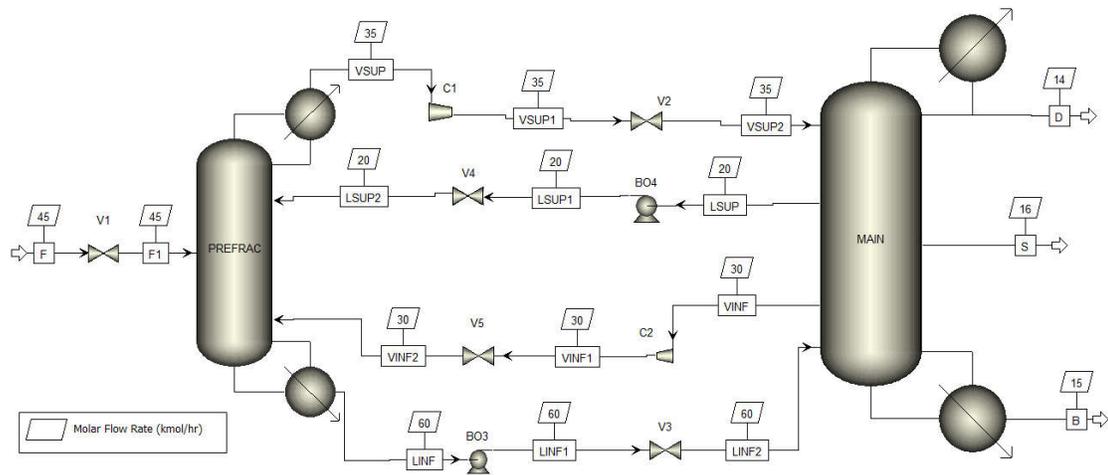


Figura 4.1: Estructura de la columna Petlyuk en Aspen Plus

7. Ejecutar el código *CDI_Calcs*. Los comandos se encuentran en la Tabla 4.3.

Lo siguiente es trabajar las matrices en Matlab[®]. A partir de los valores del estado estacionario, se escalan las señales de entrada y salida de la planta, se pasan de las variables en unidades ingenieriles a una escala porcentual con el estado estacionario en 50 %. Posteriormente se define un modelo de la planta en espacio de estados utilizando las matrices, Matlab[®] cuenta con la función propia *ss* para crear un modelo LTI. Después se reduce el orden del modelo de la planta mediante la descomposición en valores singulares (SVD por sus siglas en ingles), para esto se utiliza la función *balred*, a la función se le especifica un criterio de significancia (>0.01) y discrimina los valores que no cumplen. Para crear el controlador se utiliza la función *mpc*, con esta función se crea un objeto MPC al que se le indica el modelo a utilizar y los parámetros de control PH, CH y los pesos de la función de costo. También se le indican los límites de las variables. Una vez creado el controlador ya puede ser utilizado en una simulación en Simulink.

Para realizar la simulación es necesario conectar Aspen Plus Dynamics[®] con Matlab[®] lo cual se hace en Simulink por medio de un bloque especial de Aspen. En la Figura 4.2 se muestra el diagrama de bloques de Simulink[®] con la columna

Tabla 4.3: Código para extraer las matrices del modelo LTI

<i>CDI_Calcs</i>
SetDoc=ActiveDocument setCDI=Doc.CDI CDI.Reset CDI.AddInputVariable"blocks(""MAIN"").condenser(1).QR" CDI.AddInputVariable"blocks(""MAIN"").QrebR" CDI.AddInputVariable"blocks(""MAIN"").Reflux.FmR" CDI.AddInputVariable"streams(""D"").FmR" CDI.AddInputVariable"streams(""S"").FmR" CDI.AddInputVariable"streams(""B"").FmR" CDI.AddInputVariable"streams(""LINF"").FmR" CDI.AddInputVariable"streams(""VSUP"").FR" CDI.AddInputVariable"streams(""F"").FR" CDI.AddOutputVariable"blocks(""MAIN"").Stage(1).P" CDI.AddOutputVariable"blocks(""MAIN"").Stage(1).Level" CDI.AddOutputVariable"blocks(""MAIN"").SumpLevel" CDI.AddOutputVariable"streams(""D"").Zmn(""C1"")" CDI.AddOutputVariable"streams(""S"").Zmn(""C2"")" CDI.AddOutputVariable"streams(""B"").Zmn(""C3"")" CDI.AddOutputVariable"blocks(""PREFRAC"").Stage(1).P" CDI.AddOutputVariable"blocks(""PREFRAC"").SumpLevel" CDI.Calculate

Petlyuk controlada por el MPC. La planta es el bloque *AMSimulation*, la columna Petlyuk en Aspen Plus Dynamics[®]. Este bloque proporciona el enlace de comunicación entre Simulink[®] y Aspen Plus Dynamics[®]. El bloque está desarrollado por Aspen Technology, Inc y el bloque MPC contiene el algoritmo de control desde donde se conectan las variables de control, las variables manipuladas y los *setpoints*. El bloque *AMSimulation* es propio de Simulink[®] y tienen una programación interna ya establecida. El resto de los bloques son programados para cumplir con una función específica. En el bloque *pctNL* se encuentran los valores de las perturbaciones que son aplicadas a la planta; el bloque *Y_nom_pct* contiene los valores de los *setpoints* que se conectan al bloque del *MPC*; el bloque *U_nom_pct* contiene los valores de las variables de entrada. El bloque *FX* alimenta las fracciones másicas a Aspen. Los bloques *Pct2Engr* y *Engr2Pct* se encargan de transformar las variables

de valores porcentuales a valores ingenieriles y viceversa, necesarios para que entren los valores correctos a Aspen. El resto de los bloques $Upct_CL_MPC$, $Upctl_CL$, $Umat_ing$, $Ying$ y $Ypct_NL$ se encargan de la recolección y registro de datos que genera la simulación. Se registran los valores de las variables de entrada \underline{U} y las variables de salida \underline{Y} durante el tiempo que dura la simulación..

4.3 Modelo no-lineal con RNA

Para obtener el modelo es necesario contar con datos que representen la dinámica de las columnas. Los datos se utilizan para entrenar la RNA para que sea capaz de predecir de manera similar la dinámica de las columnas.

4.3.1 Obtención de datos

La RNA requiere muchos datos para ser entrenada. En trabajos similares donde se utilizan modelos a base de RNA, [14, 37, 36] a la planta se le dan valores aleatorios de las variables de entrada durante un tiempo y se registra el comportamiento dinámico. Debido a que este sistema tiene múltiples variables de entrada es adecuado utilizar un diseño de experimentos para obtener los datos de la dinámica del sistema. El diseño de experimentos por sí solo no es suficiente ya que la RNA requiere de una cantidad considerable de datos, por esto se agregan además casos con valores aleatorios de las variables de entrada. Matlab[®] cuenta con dos funciones para generar diseños de superficie, *bbdesign* para un diseño Box-Behnken y *ccdesign* para un diseño central compuesto. Se decidió utilizar los dos tipos de diseño para hacer uno más completo. En la Figura 4.3 se ejemplifica como es el plan experimental para tres de las variables.

Las entradas son las variables manipulables y la salida es una serie de tiempo de las variables a controlar. Los datos se obtienen simulando la planta como lazo abierto. El diagrama de Simulink[®] a lazo abierto se encuentra en la Figura 4.4. Para la toma de datos es necesario definir el tiempo de muestreo t_m . Aspen puede realizar la simulación a diferentes intervalos, para obtener una buena cantidad de datos el t_m es de 0.005 horas. Y el tiempo de simulación total es de 2 horas. La dinámica de la planta es compleja y no lineal por lo que algunos valores de entrada

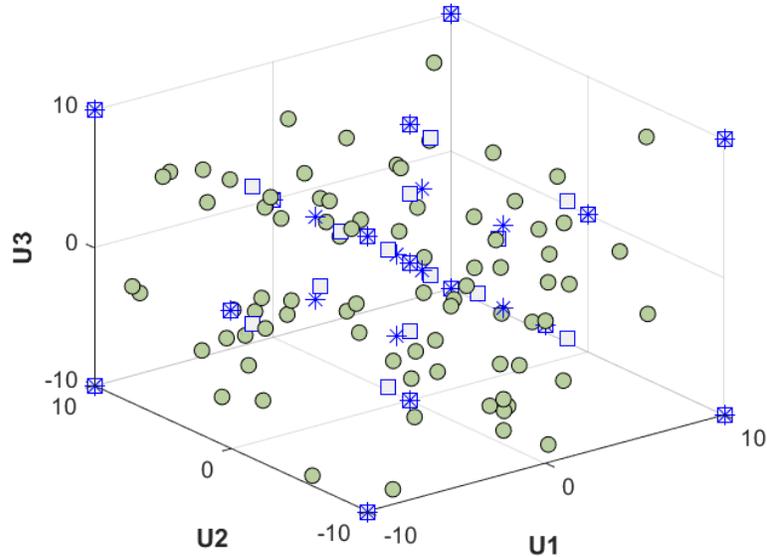


Figura 4.3: Plan experimental en el espacio de tres variables

pueden hacer que la planta pierda completamente la estabilidad. En estos casos no se cumplirá el tiempo de simulación establecido. La presión en la columna y en sus corrientes es la variable crítica que puede hacer que Aspen no tenga una solución. Para compensar esto en el diagrama de Simulink[®] se agregan algunos bloques que detengan la simulación en caso de que, la presión en algunas corrientes llegue a un valor crítico. El diagrama cuenta con el bloque *Entrada* donde se encuentran los valores de las variables de entrada correspondientes a cada caso del plan experimental. Los bloques *Pct2Engr* y *Engr2Pct* se encargan de transformar las variables de valores porcentuales a valores ingenieriles y viceversa. Y el resto de los bloques *Upctl_OL*, *Umat_ing*, *Ying_OL* y *Ypct_OL* se encargan de recolectar y registrar los datos de la simulación. En esta simulación a lazo abierto no se utiliza el controlador MPC por lo que no aparece en el diagrama, en la Figura 4.4 solo se cuenta con el bloque correspondiente a la columna Petlyuk.

El modelo debe tener la siguiente forma donde las derivadas de las variables de estado \underline{X} dependen de las variables de entrada \underline{U} y de si mismas, y las variables de

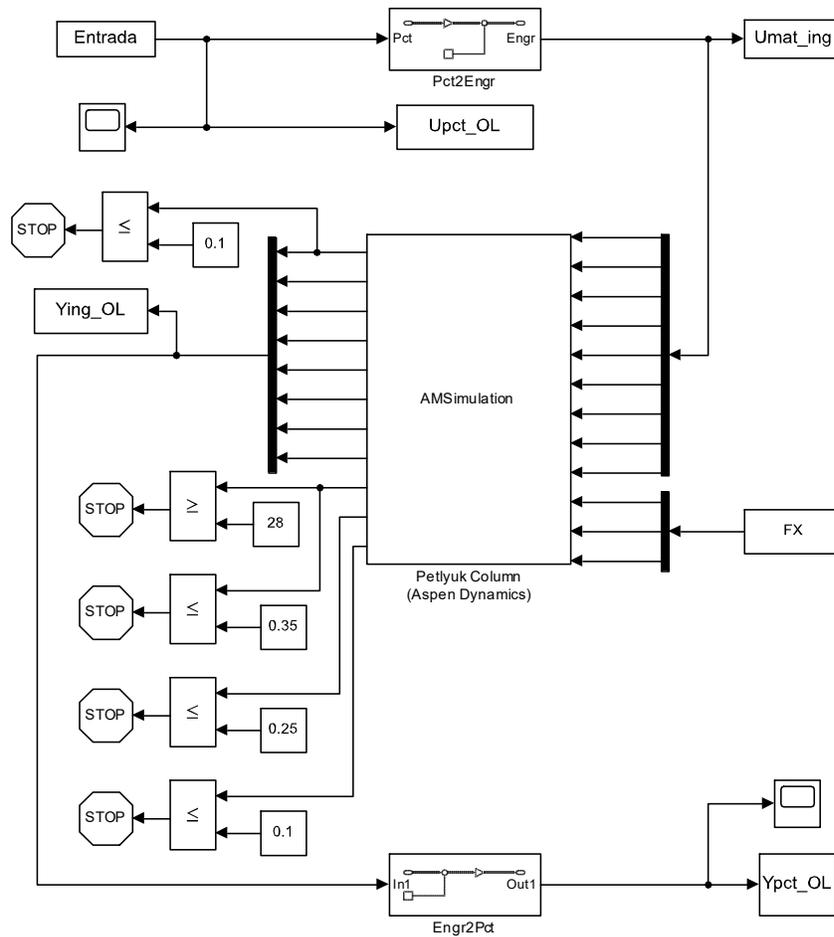


Figura 4.4: Diagrama de bloques de Simulink de la simulación a lazo abierto

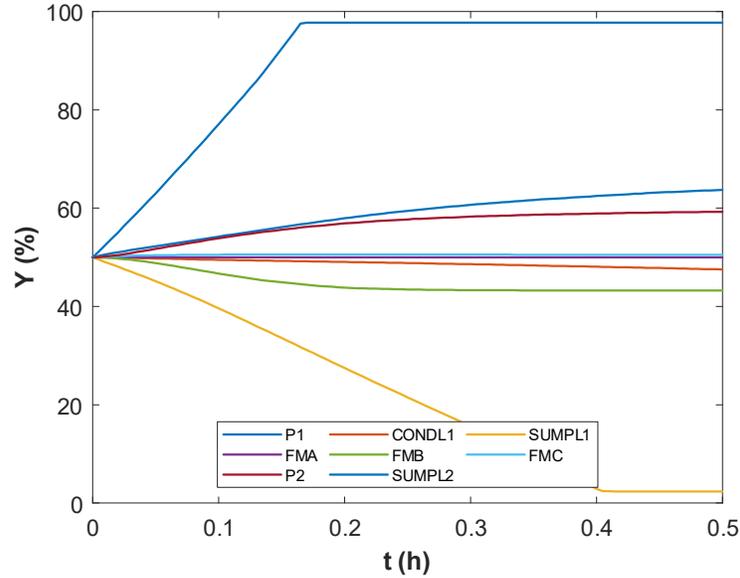


Figura 4.5: Simulación a lazo abierto

salida \underline{Y} de las variables de estado y entrada.

$$\dot{\underline{X}} = f(\underline{X}, \underline{U}) \quad (4.3)$$

$$\underline{Y} = g(\underline{X}, \underline{U}) = \underline{X} \quad (4.4)$$

En este caso las variables de estado son las mismas que de salida. La Figura 4.5 muestra un ejemplo del resultado que se puede obtener en una corrida, el comportamiento depende de las variables de entrada. La idea es que las variables de salida se alejen del valor en estado estacionario y así obtener su dinámica cuando se cambian las variables de entrada.

4.3.2 Entrenamiento de la RNA

A partir de los datos de simulación es necesario generar los datos de entrenamiento. Los datos utilizados para entrenar la RNA son en una escala porcentual. Para entrenar la RNA se toma un tiempo de entrenamiento t_e , que puede ser igual o un múltiplo del t_m . Se toman los datos cada 0.01 horas es decir dos veces el t_m . El

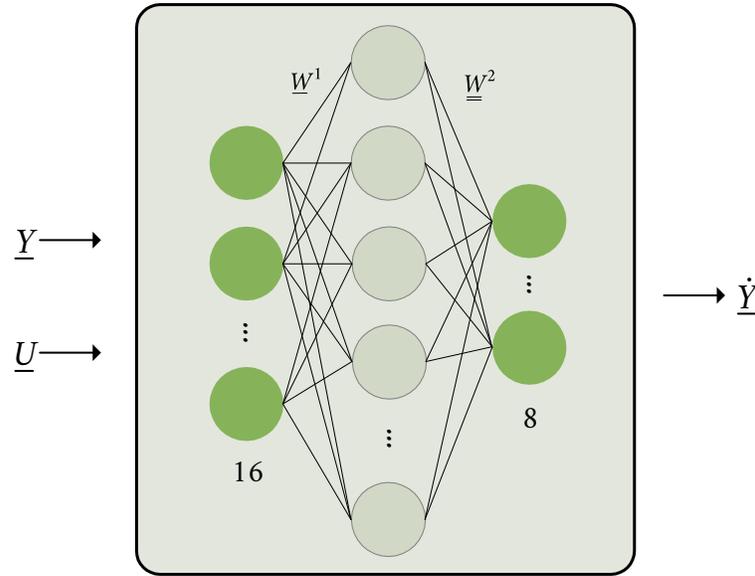


Figura 4.6: Modelo de red neuronal artificial

modelo debe calcular la derivada que tendrá la planta a un determinado tiempo. Por lo que la RNA debe de ser capaz de calcularla a partir de las variables de entrada y de las variables de salida en un tiempo anterior. Los datos se arreglan para obtener la derivada centra a un determinado tiempo.

$$\frac{dY_i}{dt} = \frac{Y_{i+1} - Y_{i-1}}{2t_e} \quad (4.5)$$

Debido a que los datos arrojados por Aspen no tienen error es necesario agregarle una pequeña desviación que funcione como error experimental. A cada dato, se le da un valor de error aleatorio con el 10% como máximo.

La Figura 4.6 ejemplifica el modelo de RNA, las entradas de la red son \underline{Y} y \underline{U} y las salidas son las derivadas de cada variable de estado $\dot{\underline{Y}}$. Son 16 entradas y 8 salidas. El entrenamiento es por el método SOS, con una RNA con una sola capa oculta. Los pesos de la capa de salida \underline{W}^2 se obtienen por regresión lineal, dejando los pesos de la capa oculta \underline{W}^1 escogidos aleatoriamente.

4.4 Diseño del controlador MPC no-lineal

Matlab[®] cuenta con una función y un toolbox para crear controladores MPC con un modelo lineal, y para modelos no lineales solo se utiliza la función *nlmpc*. Con esta función se crea un objeto MPC que tiene que ser adecuado con el modelo no-lineal. Se debe indicar el número de estados y las variables de entrada y de salida. Una vez creado el objeto se define una función de estado que en este caso es el modelo de RNA y una función para las variables de salida (Ecuaciones 4.3 y 4.4).

El controlador puede funcionar de forma continua o discontinua y dependiendo de la manera que se elige trabajar, es la forma que deben tener las funciones de estado y salida. Como el modelo de RNA calcula las derivadas, el controlador funciona de forma continua.

Una vez que se crea el objeto se establecen los parámetros como el PH y el CH. Por default Matlab[®] asigna unos pesos a las funciones de costo, pero para que el controlador funcione correctamente es necesario sintonizar estos pesos.

Hay dos formas en las que se puede correr el controlador: *adaptive* y *timevarying*. Las dos formas pueden reducir el esfuerzo computacional. La primera forma *adaptive* utiliza el modelo no-lineal y lo linealiza en el punto en que se encuentra la simulación para simplificar la predicción a lo largo del horizonte. Hay algunos trabajos que han utilizado esta configuración del controlador [3, 1]. La segunda *timevarying* calcula diferentes modelos lineales a lo largo del horizonte de predicción. Ambas formas de correr el controlador buscan reducir el esfuerzo computacional y funcionan más rápido que de forma completamente no-lineal. Un paso importante es validar el modelo, para esto se usa la función *validateFcns*, lo que hace es validar las funciones en el estado nominal y examina el objeto por posibles problemas.

El diagrama de Simulink para el controlador se encuentra en la Figura 4.7. A diferencia del controlador lineal, el bloque del MPC (bloque *Nonlinear MPC*) es más complejo y cuenta con algunas opciones extra de configuración. El bloque por defecto necesita del *setpoint (ref)*, las variables de estado (x), las variables con perturbaciones medidas (md) y los valores de las variables manipulables en un tiempo anterior ($last_mv$). Para simplificar los cálculos el bloque permite agregar valores iniciales sugeridos de las variables manipulables y de estado (entradas $mv.init$, $x.init$ y $e.init$) los valores sugeridos son los valores nominales, el valor de 50 %. Al igual

que el diagrama para el MPC Lineal, se cuenta con un bloque para las perturbaciones $pctNL$, las variables de entrada U_nom_pct , los *setpoints* Y_nom_pct , las fracciones másicas FX y los bloques para transformar de porcentajes a valores ingenieriles $Pct2Engr$ y $Engr2Pct$. De igual manera los bloques $Upct_CL_MPC$, $Upctl_CL$, $Umat_ing$, $Ying$ y $Ypct_NL$ se encargan de recolectar los datos. Todos los bloques a excepción del bloque *Nonlinear MPC* y *AMSimulation* cuentan con una programación propia.

4.5 Sintonización de los parámetros del MPC

El controlador por modelo predictivo resuelve un problema de optimización con el cual obtiene los valores de las variables manipulables o de entrada \underline{U} . Se utiliza una función de costo estándar (Ecuación 4.6) para resolver el problema de optimización. La función de costo es la suma de tres términos centrados en un aspecto del rendimiento del controlador. Cada término incluye unos pesos de ajuste correspondientes a las variables involucradas en el controlador. Estos pesos son los que deben ser ajustados para garantizar un funcionamiento correcto del controlador.

$$J = J_Y + J_U + J_{\Delta U} \quad (4.6)$$

En la función de costo el primer término (Ecuación 4.7) representa la suma del error entre las variables de salida Y_j y el *setpoint* r_j establecido a lo largo del horizonte de predicción.

$$J_Y = \sum_{j=1}^{n_Y} \sum_{i=1}^{PH} \{w_j^Y [r_j(k+i) - Y_j(k+i)]\}^2 \quad (4.7)$$

El segundo término (Ecuación 4.8) representa la contribución de las variables manipulables U_j . En algunos casos las variables manipulables también deben de cumplir con algún valor objetivo, por lo que en este término se considera esto y la ecuación es la suma el error con respecto a ese valor objetivo. Por lo general las variables manipulables no tienen un valor el cual alcanzar y por defecto los pesos son cero.

$$J_U = \sum_{j=1}^{n_U} \sum_{i=0}^{PH-1} \{w_j^U [U_j(k+i) - U_{j,t}(k+i)]\}^2 \quad (4.8)$$

El tercer término (Ecuación 4.9) corresponde al cambio en las variables manipulables. Dependiendo del valor de los pesos para cada variable, su movimiento puede ser más suave o rápido.

$$J_{\Delta U} = \sum_{j=1}^{n_U} \sum_{i=0}^{PH-1} \{w_j^{\Delta U} [U_j(k+i) - U_j(k+i-1)]\}^2 \quad (4.9)$$

En el caso del controlador MPC lineal la sintonización de los pesos se puede realizar utilizando la función *sensitivity* y ajustando los pesos minimizando una función de costo o utilizando el toolbox de diseño de MPC incluido en Matlab[®]. Para el control MPC no-lineal no se cuenta con una función o un toolbox que ayuden a sintonizar los pesos por lo que se requiere una metodología diferente.

Para sintonizar los pesos w_j^Y , w_j^U , y $w_j^{\Delta U}$, se propone variar los pesos mediante un plan experimental mediante el cual se pueda diseñar una función objetivo (Ecuación 4.10), que pueda ser optimizada.

$$J = f(w_j^Y, w_j^{\Delta U}) \quad (4.10)$$

Se utiliza un diseño de superficie box-benken para ajustar los 16 pesos de la función de costo (Ecuación 4.6), 8 para w_j^Y y 8 para $w_j^{\Delta U}$. En cada experimento se pone a funcionar el controlador con los pesos correspondientes a esa corrida y para que se realicen las acciones de control se le dan múltiples perturbaciones con valores aleatorios en las 8 variables manipulables. Al ser perturbaciones aleatorias se busca generalizar el efecto de los pesos. Para los pesos w_j^Y , los valores se encuentran entre 0.1 y 10; y para $w_j^{\Delta U}$ entre 0.01 y 1. El diseño de superficie arroja un total de 268 combinaciones de los pesos, adecuadas para un modelo cuadrático de acuerdo con la Ecuación 4.11. Por regresión utilizando los resultados del diseño de superficies se encuentran sus factores y se optimiza para encontrar los mejores pesos. La optimización se puede hacer en Matlab[®] con una función como *fmincon*. La respuesta en cada caso del diseño de superficies es una métrica de rendimiento.

$$\begin{aligned} J(w_j^Y, w_j^{\Delta U}) = & \sum_{j=1}^{n_Y} B_{w_j^Y} w_j^Y + \sum_{j=1}^{n_{\Delta U}} B_{w_j^{\Delta U}} w_j^{\Delta U} + \sum_{j=1}^{n_Y} \sum_{i=1}^{n_Y} B_{w_j^Y, w_i^Y} w_j^Y w_i^Y \\ & + \sum_{j=1}^{n_U} \sum_{i=1}^{n_U} B_{w_j^{\Delta U}, w_i^{\Delta U}} w_j^{\Delta U} w_i^{\Delta U} + \sum_{j=1}^{n_Y} \sum_{i=1}^{n_U} B_{w_j^Y, w_i^{\Delta U}} w_j^Y w_i^{\Delta U} \end{aligned} \quad (4.11)$$

Para evaluar de forma numérica el rendimiento del controlador y poder encontrar la mejor combinación de pesos, se calculan tres métricas de rendimiento: el cuadrado de error total (CE) que es la suma del cuadrado del error de cada variable de salida respecto al 50 %; el tiempo de estabilidad máximo (TE), que es el tiempo máximo en el que tardan en estabilizarse todas las variables de salida; y el sobretiro máximo absoluto u *overshoot* (OS), que es la máxima variación que puede tener una variable respecto del *setpoint* o el 50 %. Se busca que los valores de estas tres variables sean los mínimos posibles. La respuesta de la función objetivo (Ecuación 4.10) es una de las tres métricas. Por ejemplo, en Matlab[®] para realizar la sintonización de los pesos, el toolbox de diseño de un MPC lineal, minimiza solamente el CE. Para decidir cual métrica o combinación de métricas utilizar, se propone una función general (Ecuación 4.12) donde se combinen las tres métricas con una ponderación adecuada. Con un diseño de mezclas se puede encontrar la ponderación para cada métrica (x_{TE}, x_{OS}, x_{CE}). Para cada caso del diseño de mezclas se realiza una optimización, la diferencia entre casos es la ponderación utilizada en las métricas de rendimiento. Los pesos obtenidos se prueban de nuevo dando múltiples perturbaciones a las variables de entrada. De cada caso del diseño de mezclas se calcula una métrica general ponderando por igual las tres métricas, de acuerdo con la Ecuación 4.13. Los valores de las métricas están escalados para que tengan la misma magnitud y puedan ser sumadas. El diseño de mezclas es un Simplex Lattice (Ver Figura 4.8) para ajustar un modelo cúbico (Ecuación 4.14). Con los resultados se puede ajustar este modelo y obtener cual es la mejor relación entre las métricas.

$$J_m = x_{TE}TE + x_{OS}OS + x_{CE}CE \quad (4.12)$$

$$J_m = \frac{1}{3}TE + \frac{1}{3}OS + \frac{1}{3}CE \quad (4.13)$$

$$J_m = \sum_{i=1}^3 B_i x_i + B_{12}x_1x_2 + B_{13}x_1x_3 + B_{23}x_2x_3 + B_{123}x_1x_2x_3 \quad (4.14)$$

$$+ B_{12,12}x_1x_2(x_1 - x_2) + B_{13,13}x_1x_3(x_1 - x_3) + B_{23,23}x_2x_3(x_2 - x_3)$$

Una vez encontradas las relaciones adecuadas, se vuelve a optimizar la Ecuación 4.11 y se calculan los pesos definitivos. Los pesos de ambos controladores con modelo lineal y no-lineal son sintonizados utilizando esta metodología.

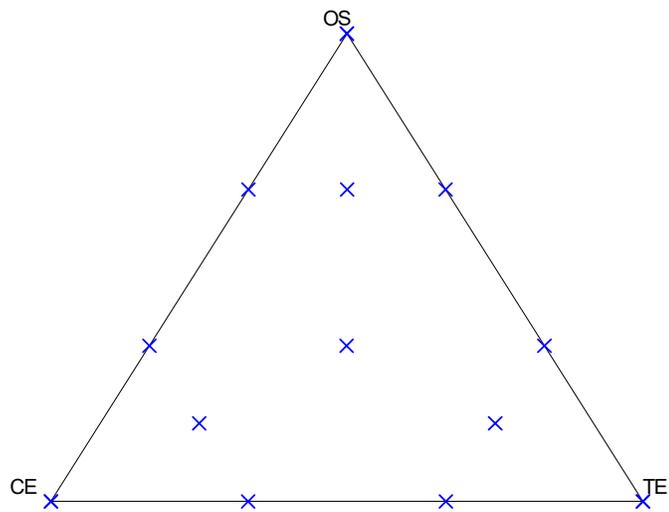


Figura 4.8: Plan experimental para factores de tipo mezcla

Capítulo 5

Resultados y discusión

5.1 Modelo no-lineal de RNA

5.1.1 Resultados del entrenamiento

Para cada mezcla se entrena una RNA. A los datos de la dinámica se le agrega error aleatorio con un 10 % como máximo. De los datos utilizadas 80% es para realizar el entrenamiento y 20 % para la validación. Para medir el grado de ajuste se mide el coeficiente de determinación de los datos de entrenamiento y validación. La Tabla 5.1 y 5.2 muestra los resultados del entrenamiento para ambas mezclas. En la Figura 5.1 y 5.2 se encuentra los resultados de un entrenamiento para la mezcla 1, están graficados los coeficientes de determinación para cada variable de salida. La RNA entrenada con los datos de la mezcla 1 alcanza un coeficiente de determinación alrededor de 0.9 para la mayoría de las variables. Por lo que se puede considerar un buen ajuste. Solo la P1 y SUMPL1 tiene un coeficiente menor.

El método SOS permite encontrar el número de neuronas adecuado para cada variable. Para elegir el número de neuronas óptimo se utiliza el coeficiente de determinación de los datos de validación. En las gráficas de las Figuras 5.1 y 5.2 se observa como este coeficiente (línea continua de color azul) sube rápidamente al aumentar el número de neuronas en la capa oculta. Después de las primeras 1000 neuronas el coeficiente sube de forma más lenta hasta llegar a su valor máximo. El coeficiente de determinación de entrenamiento (línea continua roja) sigue subiendo

Tabla 5.1: Resultados de la RNA entrenada por el método SOS para la mezcla 1

Variables por controlar	Número de neuronas	r^2
Y_1 P1	5300	0.823
Y_2 CONDL1	4900	0.918
Y_3 SUMPL1	4900	0.878
Y_4 FMA	4500	0.947
Y_5 FMB	4500	0.923
Y_6 FMC	4500	0.907
Y_7 P2	3500	0.941
Y_8 SUMPL2	3900	0.921

conforme se aumenta el número de neuronas. Aunque se sigue mejorando el ajuste después del punto máximo en el coeficiente de validación la RNA está capturando también el error en los datos, y cayendo en un sobreajuste.

El total de neuronas requeridas se encuentra alrededor de las 4500 neuronas en la capa oculta. El número requerido más bajo es de 3500 neuronas para la presión en el prefraccionador (P2) y el más alto es de 5300 neuronas para la presión en la columna principal (P1). El número de neuronas para cada variable se encuentra en la Tabla 5.1.

En la mezcla 2 ocurre algo similar. El mayor aumento en el coeficiente de determinación de validación ocurre en las primeras 1000 neuronas, después continúa aumentando de forma más lenta hasta llegar a su máximo (Ver Figura 5.3). Después de este punto continúa con el sobreajuste. En la Tabla 5.2 se encuentran el número de neuronas óptimo y su coeficiente de determinación para cada variable de salida. En la mezcla 2 se requiere un número menor de neuronas. Todas las variables requieren entre 3100 y 3700 neuronas, a excepción de FMA que requiere 4900. El coeficiente de determinación de validación para todas las variables tiene valores mayores de 0.85. Solo la fracción másica del componente B (FMB) tiene un coeficiente de 0.613. El siguiente paso es verificar la capacidad de predicción de las RNA.

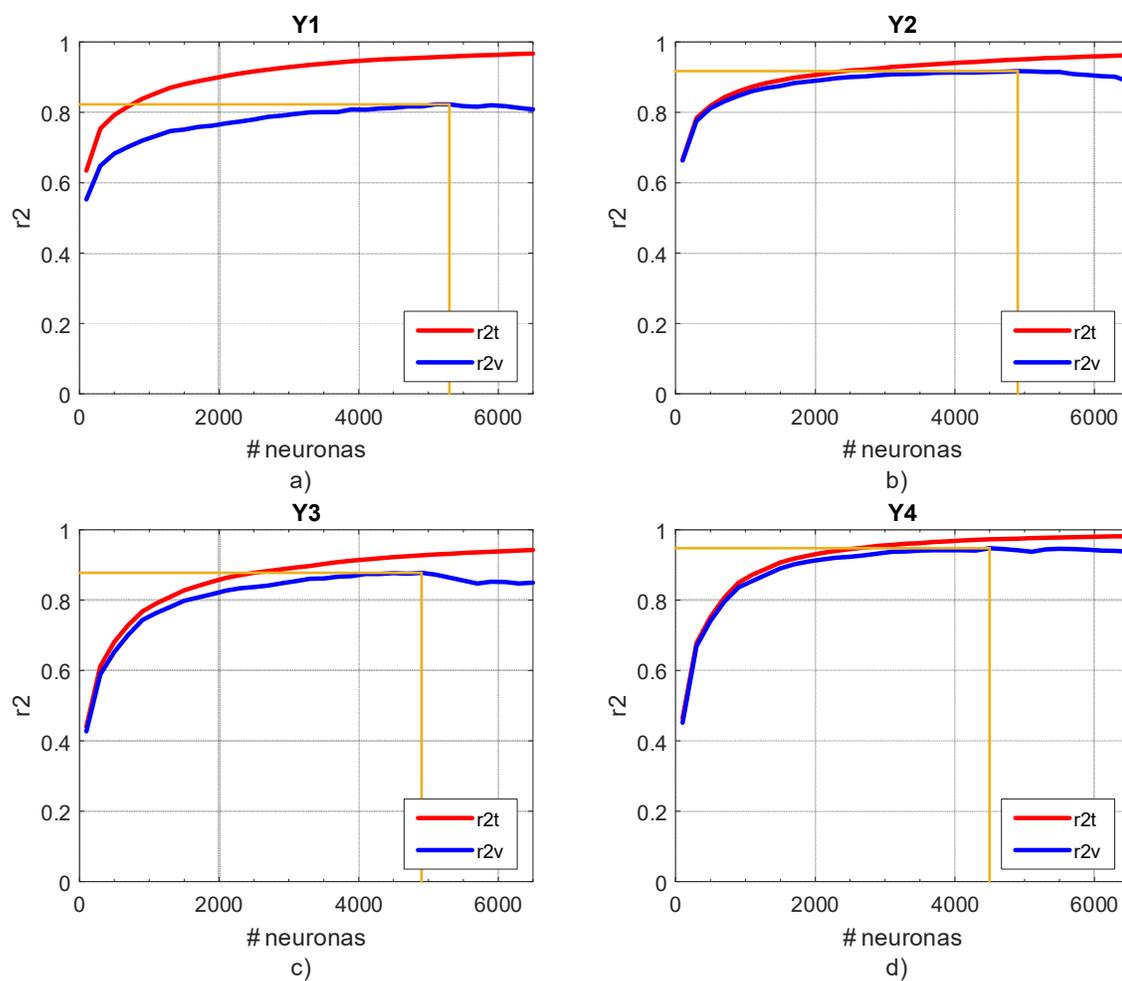


Figura 5.1: Comportamiento de la evolución del coeficiente de determinación en la mezcla 1 para las variables de salida Y_1 a Y_4 . a) P1; b) CONDL1; c) SUMPL1; d) FMA

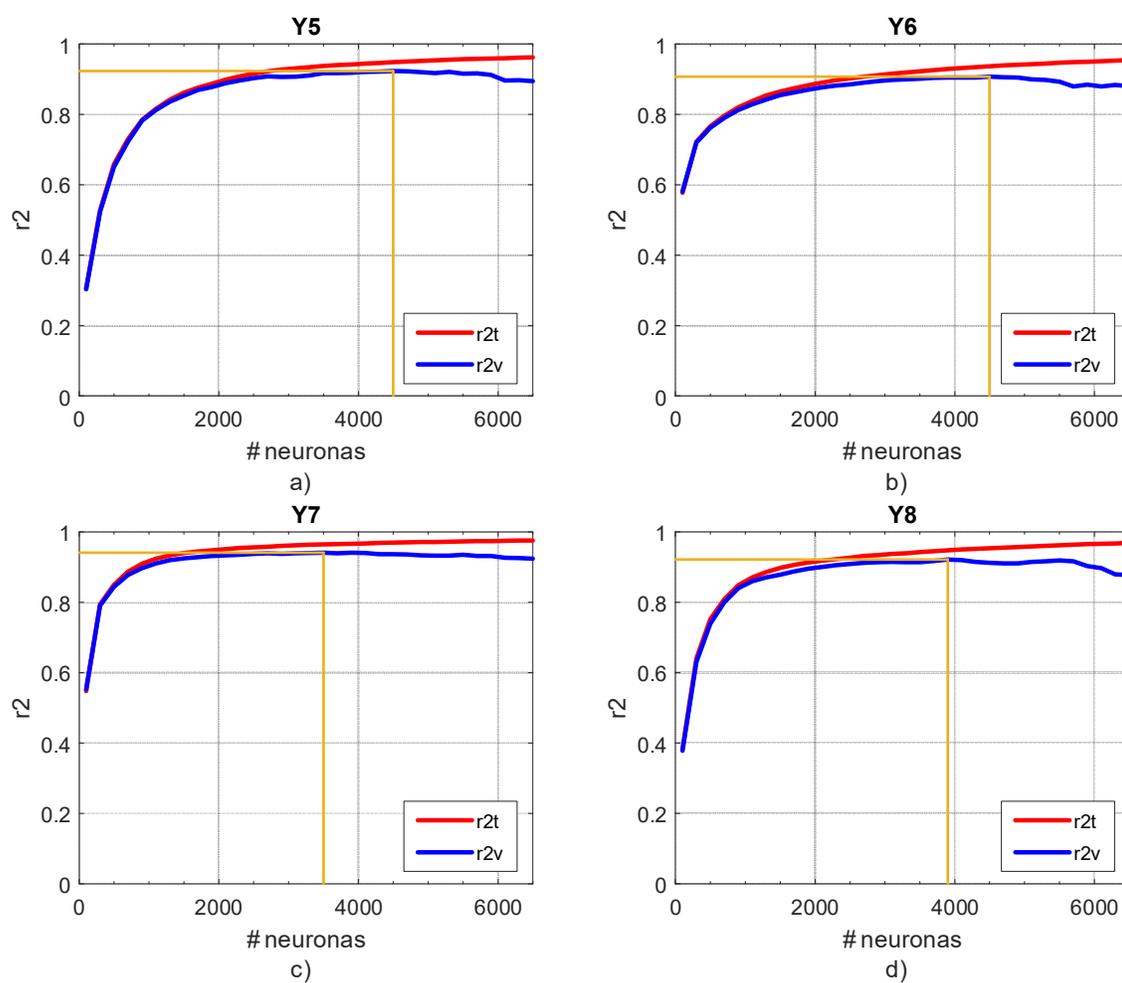


Figura 5.2: Comportamiento de la evolución del coeficiente de determinación en la mezcla 1 para las variables de salida Y_5 a Y_8 . a) FMB; b) FMC; c) P2; d) SUMPL2

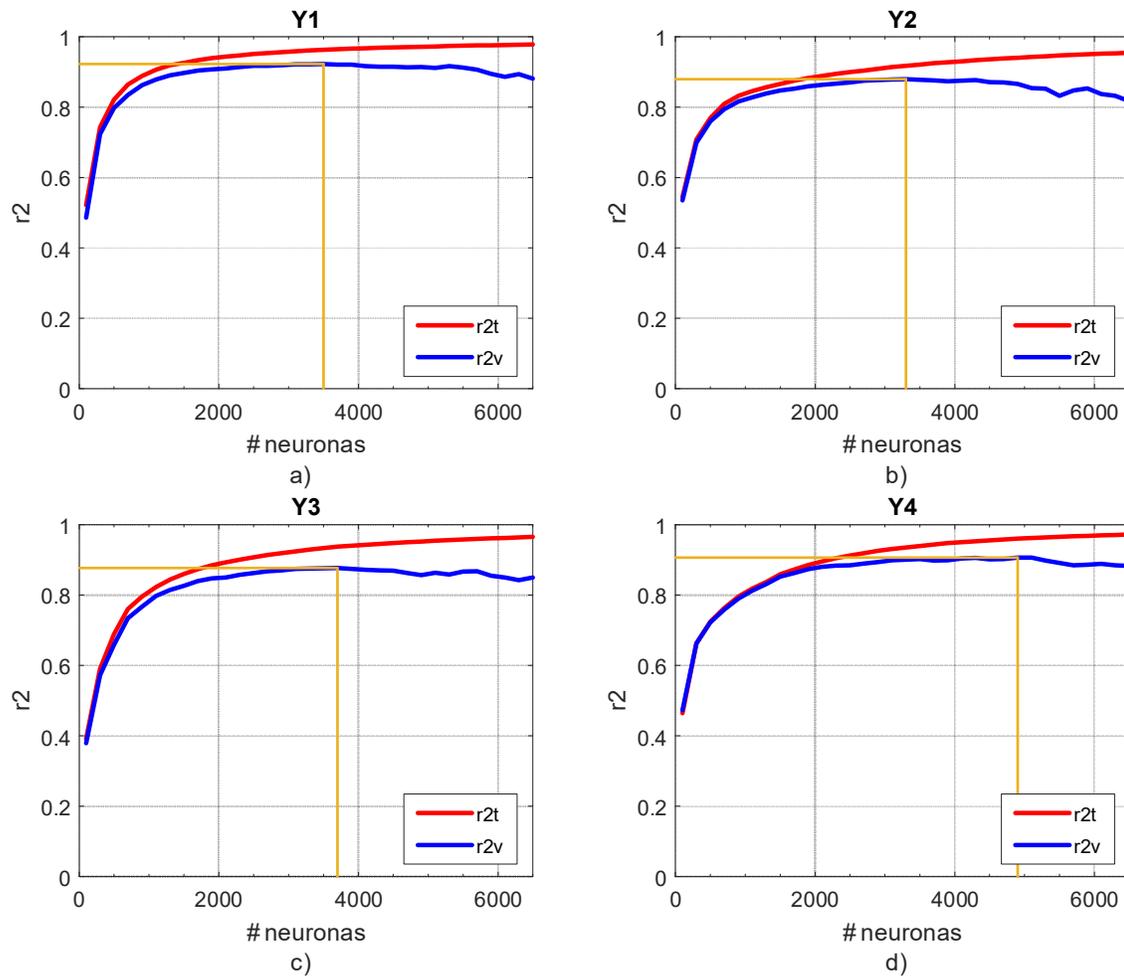


Figura 5.3: Comportamiento de la evolución del coeficiente de determinación en la mezcla 2 para las variables de salida Y_1 a Y_4 . a) P1; b) CONDL1; c) SUMPL1; d) FMA

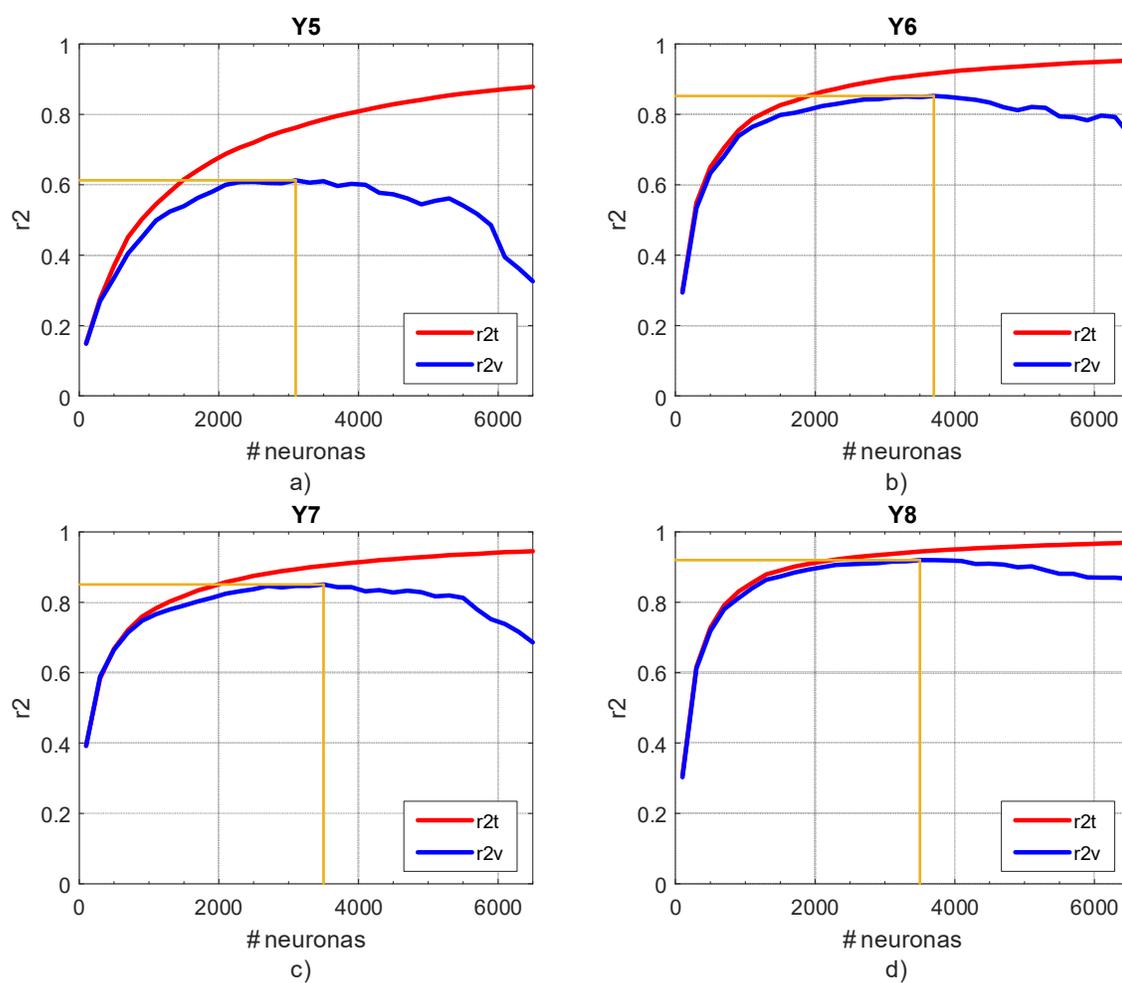


Figura 5.4: Comportamiento de la evolución del coeficiente de determinación en la mezcla 2 para las variables de salida Y_5 a Y_8 . a) FMB; b) FMC; c) P2; d) SUMPL2

Tabla 5.2: Resultados de la RNA entrenada por el método SOS para la mezcla 2

Variables por controlar	Número de neuronas	r^2	
Y_1	P1	3500	0.922
Y_2	CONDL1	3300	0.880
Y_3	SUMPL1	3700	0.877
Y_4	FMA	4900	0.906
Y_5	FMB	3100	0.613
Y_6	FMC	3700	0.853
Y_7	P2	3500	0.851
Y_8	SUMPL2	3500	0.920

5.1.2 Predicción de la RNA

Con las RNA entrenadas, es posible verificar su función como modelo dinámico. La RNA proporciona la derivada en un punto por lo que se puede hacer una integración durante un tiempo. La integración se hace de manera discreta utilizando un tiempo de integración t_i . Para evaluar la predicción de la RNA, se prueban diferentes tiempos de integración y diferentes horizontes de predicción a corto y mediano plazo. Es decir, se realiza la predicción dos o más veces el t_m . La integración inicia con las variables de salida en su valor en estado estacionario. Para que el comportamiento de las columnas cambie, las variables de entrada se inician con un valor diferente al del estado estacionario. A la RNA se le dan los valores de \underline{Y}_k y \underline{U}_k y se predicen los valores siguientes varios tiempos adelante. La Ecuación 5.1 representa la forma en que se calculan las variables de salida en cada paso y como se realiza la integración.

$$\underline{Y}_{k+1} = 2t_i \frac{d\underline{Y}_k}{dt} + \underline{Y}_{k-1} = 2t_i RNA(\underline{U}_k, \underline{Y}_k) + \underline{Y}_{k-1} \quad (5.1)$$

Para la predicción a corto plazo la predicción se realiza utilizando los valores reales de las variables. Estos valores se obtienen previamente de una simulación a lazo cerrado utilizando las mismas variables de entrada \underline{U}_k . Solo se calcula el siguiente paso $\underline{Y}_{k+1,P}$ en cada punto $\underline{Y}_{k,R}$.

$$\underline{Y}_{k+1,P} = 2t_i RNA(\underline{U}_k, \underline{Y}_{k,R}) + \underline{Y}_{k-1,R} \quad (5.2)$$

En la Figura 5.5 y Figura 5.6 se encuentra la predicción de la RNA. Se presenta la predicción para tres tiempos de integración diferentes: 0.005 h, 0.01 h y 0.02 h.

La Figura 5.5 (a, c y e) y 5.2 (a, c y e) muestran la respuesta dinámica a corto plazo predicha por la integración de la RNA ante una perturbación. Los cruces representan la predicción y las líneas continuas el comportamiento dinámico real.

En la predicción a largo plazo a la RNA se le dan los valores predichos y así consecuentemente durante un horizonte de predicción. Esta es la forma en la que el controlador MPC realiza la predicción de manera interna. Se espera que a lo largo del horizonte el error en la predicción se va acumulando.

$$\underline{Y}_{k+1,P} = 2t_i RNA(\underline{U}_k, \underline{Y}_{k,P}) + \underline{Y}_{k-1,P} \quad (5.3)$$

La Figura 5.5 (b, d y f) y 5.2 (b, d y f) muestra la predicción a largo plazo durante un horizonte predicción de 0.5 h.

En la predicción a corto plazo para los tres tiempos de integración, la predicción es correcta. De acuerdo con los resultados en la Figura 5.5 (a, c y e) y 5.2 (a, c y e) la RNA es capaz de predecir a corto plazo de manera efectiva. El siguiente punto predicho por lo general siempre se encuentra sobre la línea de los valores reales. A largo plazo la RNA tiene más dificultades, conforme aumenta el tiempo la respuesta se vuelve incorrecta, el error se acumula hasta que la predicción es completamente errónea. Comparando con la respuesta dinámica real, la RNA puede predecir de una forma correcta hasta alrededor de 0.2 y 0.3 horas. Lo que corresponde a 40 veces el t_m de 0.005 h. Con este dato se puede elegir el horizonte de predicción para el controlador de 0.2 h. La predicción cambia ligeramente con el tiempo de integración, con un t_i de 0.02 h la desestabilización ocurre en tiempos similares que con un t_i de 0.01 o 0.005 h, se puede ver en las tres gráficas b, d y f de la Figura 5.5 y 5.2. En general la RNA está haciendo una buena predicción, aunque a largo plazo la predicción se vuelve es más errónea, para los objetivos del controlador el tiempo durante el que se predice de forma correcta es adecuado. Ya que, con cada ajuste de control, el MPC vuelve a realiza la predicción y se ha demostrado que la predicción a corto plazo se realiza de forma correcta.

5.2 Sintonización de los pesos del MPC

Después de diseñar el controlador se continúa con su sintonización, esto para que el controlador funcione de manera óptima. La sintonización se realiza ajustando los

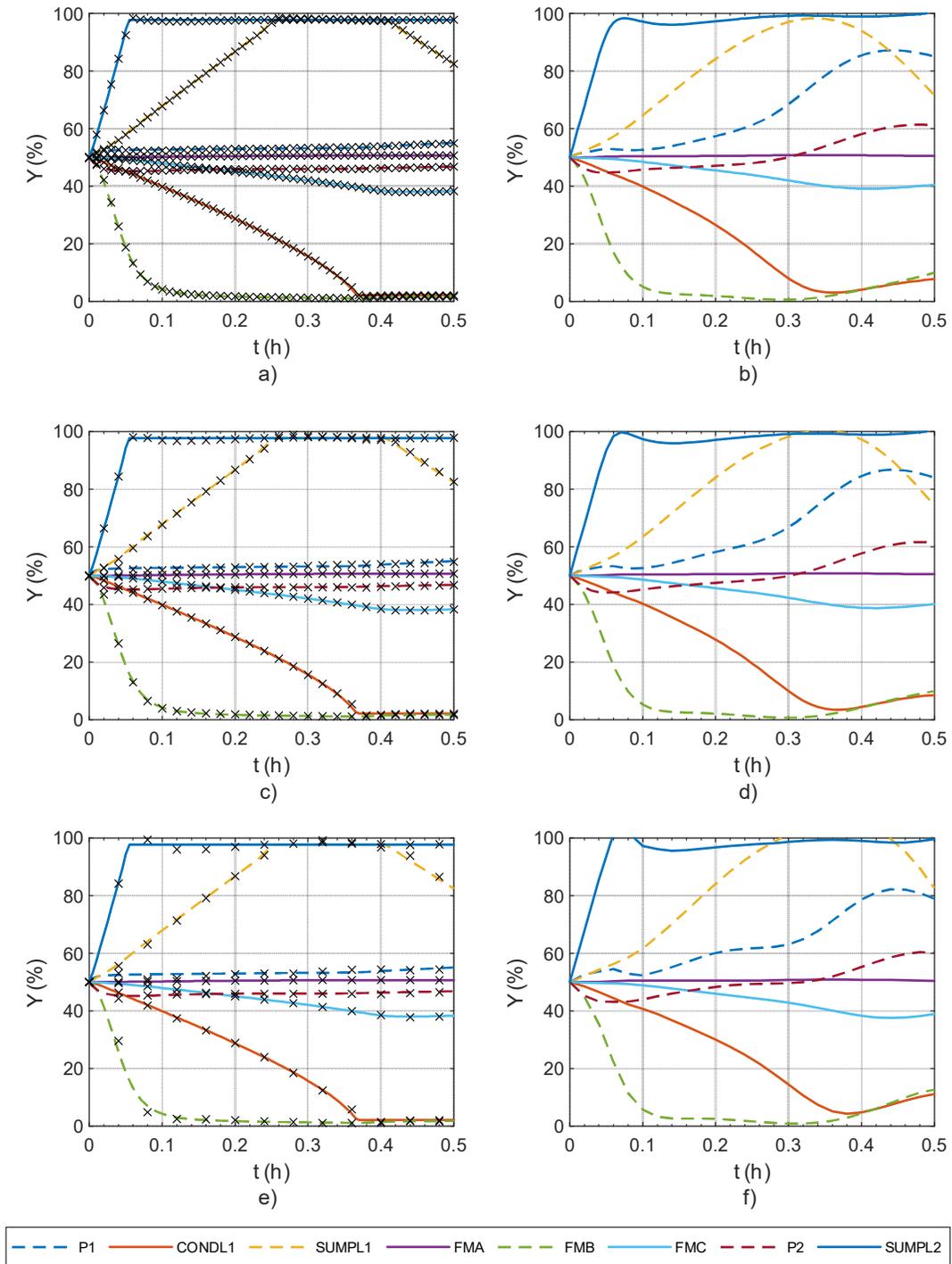


Figura 5.5: Predicción de las variables de salida de la mezcla 1 por la RNA ante cambios aleatorios en las variables de entrada. a) Corto plazo $t_i=0.005$ h; b) Largo plazo $t_i=0.005$ h; c) Corto plazo $t_i=0.01$ h; d) Largo plazo $t_i=0.01$ h; e) Corto plazo $t_i=0.02$ h; f) Largo plazo $t_i=0.02$ h

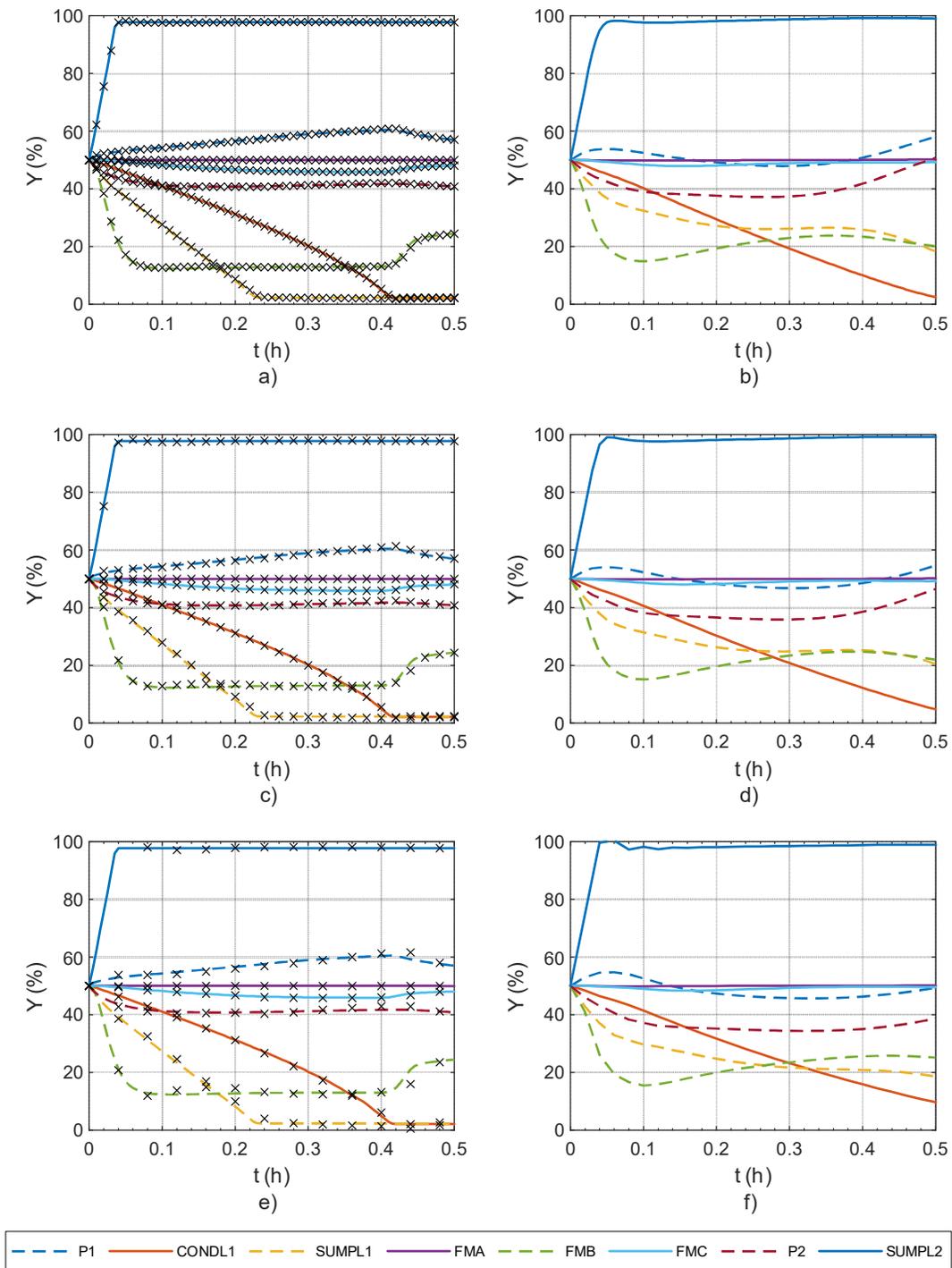


Figura 5.6: Predicción de las variables de salida de la mezcla 2 por la RNA ante cambios aleatorios en las variables de entrada. a) Corto plazo $t_i=0.005$ h; b) Largo plazo $t_i=0.005$ h; c) Corto plazo $t_i=0.01$ h; d) Largo plazo $t_i=0.01$ h; e) Corto plazo $t_i=0.02$ h; f) Largo plazo $t_i=0.02$ h

pesos w_j^Y y $w_j^{\Delta U}$ de la función de costo (Ecuación 4.6) que utiliza el controlador. Los pesos w_j^Y y $w_j^{\Delta U}$ son ajustados de acuerdo con la metodología mencionada anteriormente. Para cada controlador se sintonizan los pesos utilizando el plan experimental de superficie, el criterio para sintonizar los pesos es utilizando una combinación de las métricas de rendimiento, la mejor relación se encuentra utilizando un diseño de mezclas. Este procedimiento puede ser entendido como una suboptimización para ajustar los pesos, ya que los pesos pueden ser encontrados utilizando una sola de las métricas, por ejemplo, minimizando el cuadrado del error. Para cada caso se evalúa el control del MPC, a las variables de entrada se les da un cambio de forma aleatoria con un máximo de 10%. Con las perturbaciones establecidas, la planta se desestabiliza y el controlador debe regresar las variables de salida a su *setpoint*, el estado estacionario. El desempeño está determinado por los valores que tienen los pesos. Los resultados se encuentran en las Figuras 5.7 a 5.10, donde se grafica la respuesta de la Ecuación 4.13. El asterisco rojo está donde se encuentra el mínimo de la función ajustada y el cual representa las relaciones adecuadas entre las métricas.

La Tabla 5.3 tiene los resultados para los cuatro controladores. En todos los casos la mejor relación incluye al cuadrado del error. Para el MPC lineal de la mezcla 1 las ponderaciones están entre el CE y el OS. En la Figura 5.7 se ve como los mejores casos están cercanos al CE. Para el MPC lineal de la mezcla 2 la mejor relación es con el CE y el OS. En la Figura 5.9 se observa que los casos con mejor rendimiento están al centro de la figura, en las esquinas utilizando solo el OS o el TE se tienen rendimientos menores. En el MPC no-lineal de la mezcla 1, los mejores casos se encuentran en la región cercana a el TE, para este controlador la mejor relación es entre CE y el TE. Y para el MPC no-lineal de la mezcla 2, la relación es entre el CE y OS. En la Figura 5.10 los casos con mejor rendimiento se encuentran cercanos al OS. Para cada controlador el comportamiento obtenido al utilizar diferentes combinaciones de las métricas es muy diferente, cada controlador tiene su propio comportamiento y se puede ver en las superficies que generan. La ponderación óptima que se obtiene es diferente entre ellos. De los resultados obtenidos, se puede observar que para sintonizar los pesos se puede utilizar el CE en combinación con una de las dos métricas, en tres de los cuatro controladores es

Tabla 5.3: Ponderaciones de las métricas de rendimiento obtenidas para cada controlador

MPC	Mezcla	xTE	xOS	xCE
Lineal	1	0	0.15	0.84
Lineal	2	0	0.65	0.35
No-lineal	1	0.011	0.595	0.393
No-lineal	2	0.702	0	0.298

con el OS.

Los pesos obtenidos para cada controlador se encuentran en la Tabla 5.4 y 5.5. Estos son los pesos que se utilizan de forma definitiva en cada controlador. Aunque las variables a controlar y manipular son las mismas para los 4 controladores, los valores para los mismos pesos son muy diferentes entre sí. Los valores por default para las variables de salida son de 1, los valores obtenidos se mantienen alrededor de 1 pero son muy diferentes. Lo mismo ocurre para los pesos de los cambios en las variables de entrada, el valor por default es de 0.1 y de nuevo los valores obtenidos son muy diferentes. Siguiendo la idea de las reglas heurísticas, en las que, al aumentar el valor del peso de una variable de salida, el controlador le da más importancia. Con el ajuste de los valores de los pesos se puede entender que algunas variables de la columna Petlyuk requieren de mayor atención y son más importantes para controlar de forma adecuada. De igual forma en las variables de entrada, las variables con valores más grandes en sus pesos son las variables que tendrán un movimiento menos brusco. El ajuste de los pesos permite tener un equilibrio entre la importancia de las variables a controlar y la suavidad con lo que se mueven las variables de entrada.

5.3 Resultados de la la simulación con MPC

Con los pesos optimizados para los controladores con el modelo lineal y el no-lineal de RNA lo siguiente es verificar su funcionamiento. Se espera que el rendimiento del MPC no-lineal sea mejor que el MPC lineal. Para poder comparar el desempeño de cada controlador, cada columna es sometida a diversas perturbaciones. El controlador cuenta con ocho variables de entrada que pueden ser perturbadas. De estas

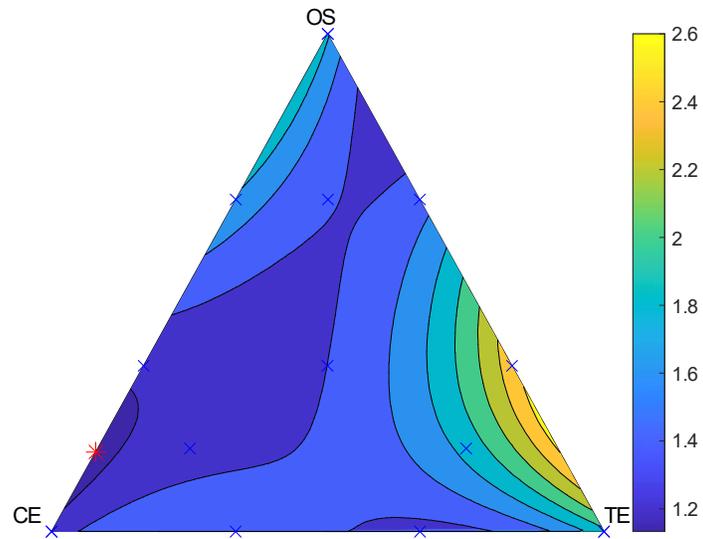


Figura 5.7: Ponderación de las métricas del MPC lineal para la mezcla 1

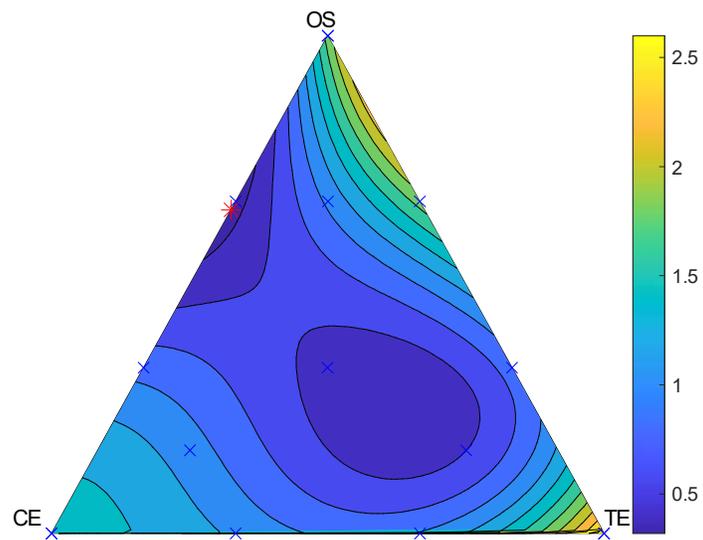


Figura 5.8: Ponderación de las métricas del MPC lineal para la mezcla 2

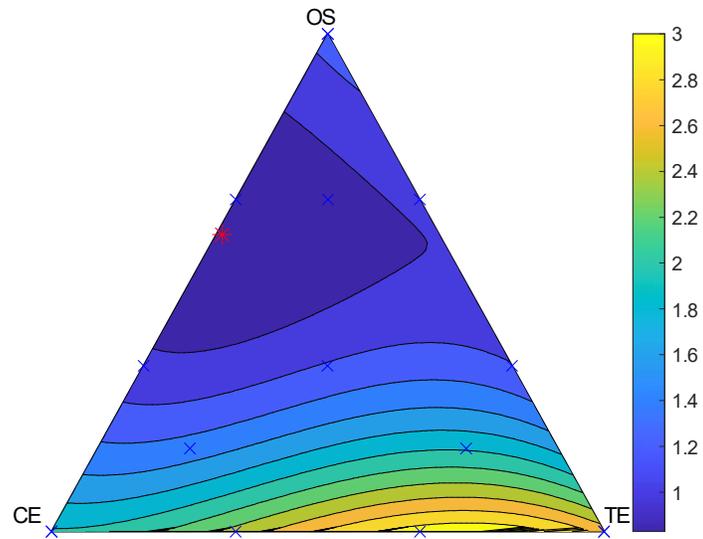


Figura 5.9: Ponderación de las métricas del MPC no-lineal para la mezcla 1

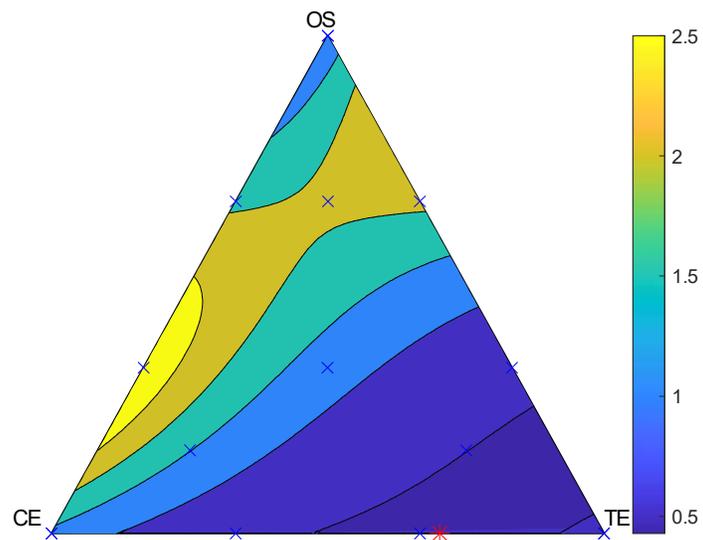


Figura 5.10: Ponderación de las métricas del MPC no-lineal para la mezcla 2

Tabla 5.4: Pesos de la función de costo del MPC correspondientes a las variables de salida

MPC	Lineal	Lineal	No-lineal	No-lineal
Mezcla	1	2	1	2
$w_1^Y, w_j^{\Delta U}$	1.701	2.139	1.843	1.276
w_2^Y	1.642	1.863	1.253	1.743
w_3^Y	2.041	2.098	2.004	1.282
w_4^Y	1.186	2.394	1.405	1.724
w_5^Y	0.86	1.079	0.594	0.359
w_6^Y	1.102	1.89	2.089	2.371
w_7^Y	1.582	0.242	1.212	1.028
w_8^Y	0.94	3.898	1.657	2.351
Promedio	1.339	1.956	1.451	1.571

ocho variables se propone perturbar cinco de ellas. Las perturbaciones se aplican a el flujo de vapor en el reboiler (QC), a el flujo de líquido de refrigeración en el condensador (QR), a la relación de reflujo (RF), a el flujo en la corriente inferior de líquido entre el prefraccionador y la columna principal (LINF) y al flujo en la corriente superior de vapor entre el prefraccionador y la columna principal (VSUP). En un proceso real en algún punto se requiere que las variables de salida cambien y mantengan en algún otro valor, por motivos de seguridad, de calidad o por diversos motivos. Por lo que también se propone hacer un cambio de *setpoint* en las presiones de la columna, en el prefraccionador (P2) y en la columna principal (P1). En total se cambian ocho variables de la columna Petlyuk. Se propone un plan experimental con un diseño de superficie IV-optimal para un modelo lineal, el objetivo es comparar ambos modelos: lineal y no-lineal. En total el plan experimental cuenta con 20 casos (Ver Tabla 5.6).

De cada caso del plan experimental se tienen gráficas que representan el comportamiento dinámico de la planta, como es que fueron cambiando las variables de salida ante las perturbaciones y como el MPC realiza las acciones de control. Se tiene una gráfica de las variables de salida y una para las variables de entrada. Para evaluar de forma numérica el rendimiento de cada controlador, se calcula el cuadrado del error de cada variable de salida respecto al 50 % y se suman para obtener el cuadrado del error total (CE). También se obtiene el tiempo de estabili-

Tabla 5.5: Pesos de la función de costo del MPC correspondientes a los cambios en las variables de entrada

MPC	Lineal	Lineal	No-lineal	No-lineal
Mezcla	1	2	1	2
$w_1^{\Delta U}$	0.397	0.166	0.033	0.063
$w_2^{\Delta U}$	0.082	0.025	0.072	0.071
$w_3^{\Delta U}$	0.051	0.048	0.282	0.124
$w_4^{\Delta U}$	0.028	0.061	0.197	0.364
$w_5^{\Delta U}$	0.022	0.061	0.434	0.083
$w_6^{\Delta U}$	0.089	0.203	0.074	0.238
$w_7^{\Delta U}$	0.023	0.396	0.037	0.076
$w_8^{\Delta U}$	0.038	0.045	0.166	0.101
Promedio	0.192	0.334	0.255	0.347

dad máximo (TE). Y también se mide el sobre tiro máximo absoluto (OS), que es la máxima variación que puede tener una variable respecto del *setpoint*.

5.3.1 MPC lineal

En todos los casos del plan experimental el MPC con el modelo lineal logra estabilizar la columna y cumplir con los cambios de *setpoint*. Dependiendo de las perturbaciones al controlador le tomo diferente tiempo llegar a un estado estable. En general el controlador funciona de manera adecuada. En la Tabla 5.7 se encuentran los resultados del plan experimental para este controlador. En general los valores del TE se encuentran por debajo de 1 h, a excepción del caso número 4 y el número 16 en los que el TE fue mayor. El OS de cada caso tiene valores variados, el menor es de una diferencia 5.3 % en el caso 17 y el mayor es de 24.9 % en el caso 12. De igual forma con el CE las respuestas son variadas, hay algunos casos en los que el controlador tiene un comportamiento muy bueno como lo es en el caso 13 en el que el CE es 106, pero también hay casos como el número 4 y 20 en el que el CE esta alrededor de 3000. Hay varios casos en los que el CE es mayor de 1000, en estos casos las perturbaciones tienen un mayor efecto en desestabilizar la columna.

Tabla 5.6: Plan experimental para las perturbaciones

Std	P1	P2	QC	QR	RF	LINF	VSUP	Mezcla
1	10	10	-10	10	-10	-10	-10	1
2	-10	-10	10	-10	10	-10	-10	1
3	-10	-10	-10	-10	-10	10	-10	1
4	-10	10	10	10	-10	10	-10	1
5	10	-10	-10	10	10	10	-10	1
6	-10	-10	-10	-10	-10	-10	10	1
7	10	10	-10	-10	10	-10	10	1
8	-10	-10	10	10	10	-10	10	1
9	10	10	10	10	-10	10	10	1
10	10	10	10	-10	10	10	10	1
11	10	-10	-10	-10	-10	-10	-10	2
12	10	10	10	10	-10	-10	-10	2
13	10	10	10	-10	10	-10	-10	2
14	-10	10	-10	-10	10	10	-10	2
15	-10	-10	10	10	10	10	-10	2
16	-10	-10	10	10	-10	-10	10	2
17	-10	10	-10	10	10	-10	10	2
18	-10	10	-10	-10	-10	10	10	2
19	10	-10	10	-10	-10	10	10	2
20	10	-10	-10	10	10	10	10	2

5.3.2 MPC no-lineal

El modelo a base de una RNA es capaz de predecir a corto plazo de forma aceptable y funciona adecuadamente para realizar la sintonización del MPC. En este punto se evalúa el controlador utilizando las perturbaciones y cambios de *setpoint* del plan experimental. Los resultados muestran que el controlador es capaz de corregir las perturbaciones y controlar la planta. Los resultados del plan experimental se encuentran en la Tabla 5.8. En todos los casos el TE fue menor de 1 hora. El mayor TE fue de 0.9 h en el caso número 8 y el menor fue de 0.09 h en el caso 15. El OS tiene valores desde 4.5 % en el caso 6 hasta 18.1 el caso 20. El CE en la mayoría de los casos se encuentra por debajo de 1000 y con ello se pueden identificar cuales son los casos en los que a el controlador se le dificulta más estabilizar la columna. Estos casos son el número 5, 12 y 20, los cuales tienen los mayores valores del CE.

Tabla 5.7: Métricas de rendimiento para todos los casos del modelo lineal

Std	TE (h)	OS (%)	CE
1	0.31	14.5	972.7
2	0.2	11.8	811.6
3	0.5	5.8	333.4
4	1.01	10.2	2966.2
5	0.58	10.8	1020.1
6	0.2	6.4	349.1
7	0.18	17.2	944.3
8	0.17	5.8	426.1
9	0.96	12.2	1970.3
10	0.15	11.7	539.5
11	0.99	13.3	676.8
12	0.46	24.9	3004.1
13	0.2	15.5	106.3
14	0.5	10.8	1214.1
15	0.42	8	509.8
16	1.01	9.6	1140.2
17	0.72	8.5	306.4
18	0.26	12.9	606.3
19	0.78	16.4	1207.1
20	0.61	16.2	3442.4

De acuerdo con los resultados del plan experimental, en la mayoría de los casos el TE, el OS y el CE tienen valores menores que en el MPC lineal. En la siguiente sección se abordará mejor las diferencias del control entre ambos modelos.

5.4 Comparación entre los modelos lineal y no-lineal

El control MPC utilizando ambos modelos funciona de manera adecuada. El objetivo del control es estabilizar y establecer las variables en los *setpoint*. En esta sección se analizan las principales diferencias del control utilizando ambos modelos. Se ha mostrado que una RNA se puede utilizar como modelo dinámico de

Tabla 5.8: Métricas de rendimiento para todos los casos del modelo no-lineal

Std	TE (h)	OS (%)	CE
1	0.15	9.9	662.9
2	0.2	11.7	816.7
3	0.28	5.8	209.3
4	0.6	6.5	514
5	0.66	16.7	1047.2
6	0.11	4.5	169.4
7	0.18	12.9	659
8	0.9	7.2	449.7
9	0.19	11.8	576.9
10	0.15	11.7	634.7
11	0.25	13.3	725.1
12	0.46	14	1832.2
13	0.16	13.5	479.3
14	0.39	6.9	437.3
15	0.09	6.5	195.5
16	0.87	6.6	533.7
17	0.11	6.8	203.4
18	0.22	12.3	711.4
19	0.37	13.2	936.7
20	0.37	18.1	2049.3

una columna Petlyuk. Implementar este modelo no-lineal en un MPC es posible y funciona de forma adecuada para controlar una columna. En las siguientes Figuras desde la 5.11 hasta 5.16 se encuentra graficado el comportamiento dinámico de las columnas Petlyuk ante perturbaciones, para los casos 1, 5, 6, 12, 15 y 17 del plan experimental. Solo se presentan 6 casos del plan experimental ya que se requiere una cantidad grande de espacio para presentar todos los casos. Los casos elegidos están seleccionados en base a los cambios de *setpoint* y por mezcla. Por cada mezcla es un caso donde los cambios de *setpoint* sean de +10 %, un caso donde se tengan cambios de *setpoint* diferentes y un caso donde sean de -10 %. Esta graficado el comportamiento del MPC no-lineal junto al del MPC lineal. De esta forma se pueden comparar de forma gráfica ambos controladores.

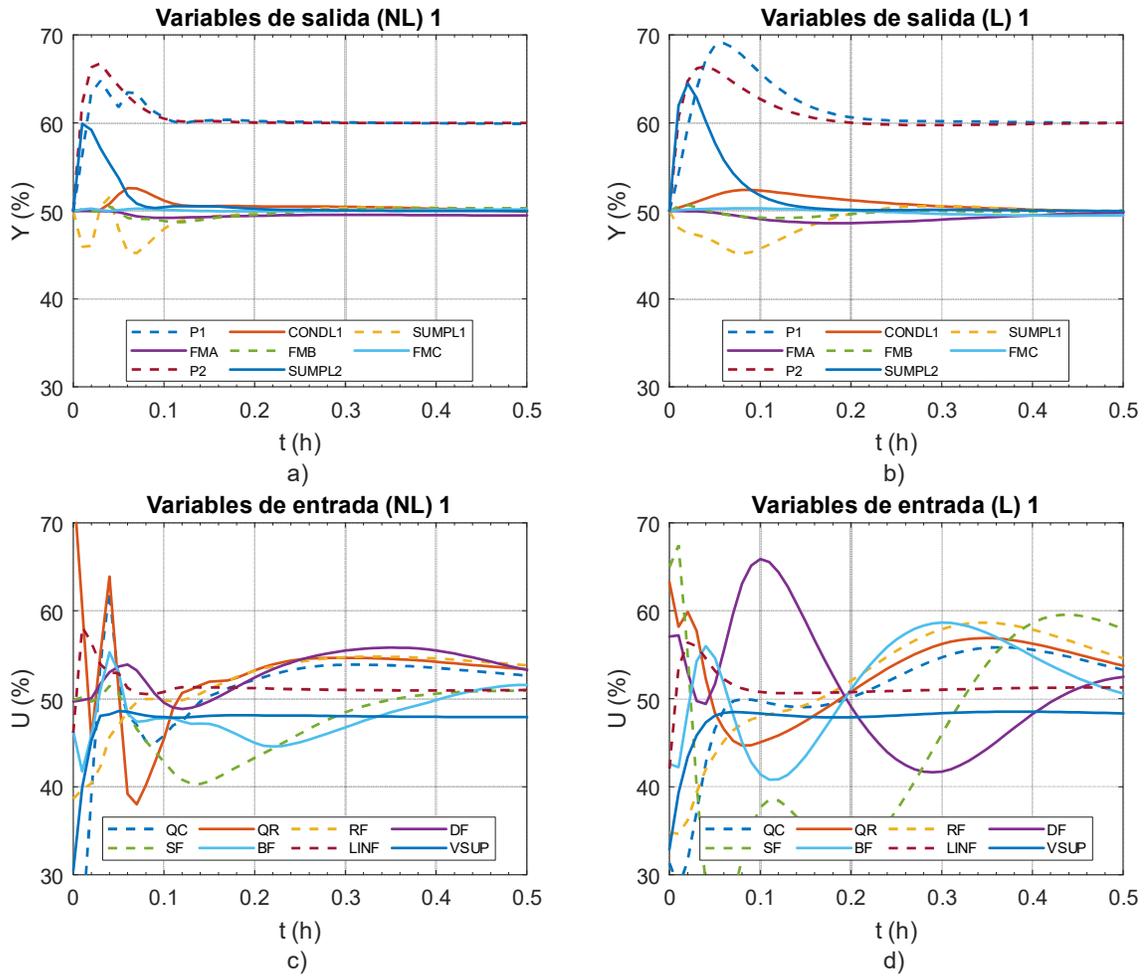


Figura 5.11: Comportamiento dinámico de las variables de entrada y salida del caso 1. a) Variables de salida (MPC no-lineal); b) Variables de salida (MPC lineal); c) Variables de entrada (MPC no-lineal); d) Variables de entrada (MPC lineal)

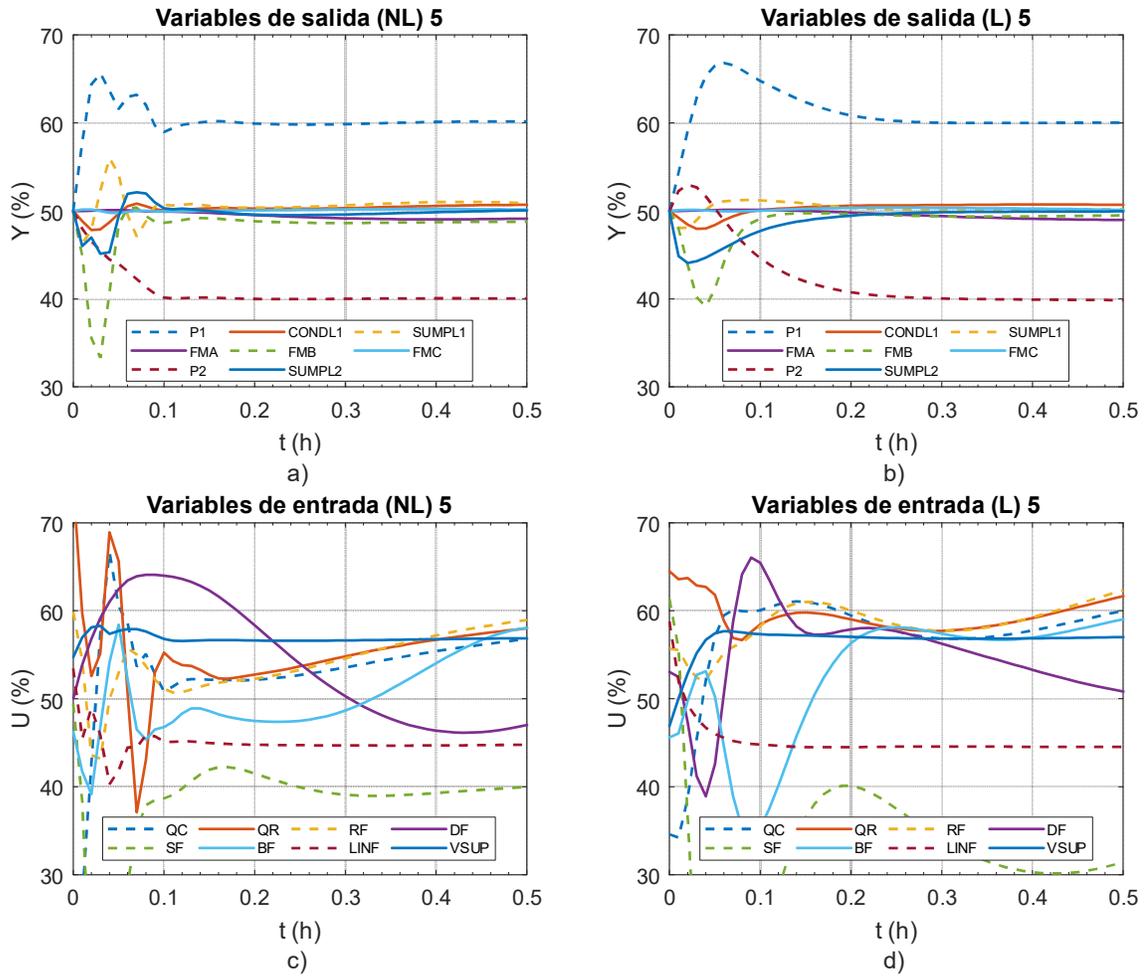


Figura 5.12: Comportamiento dinámico de las variables de entrada y salida del caso 5. a) Variables de salida (MPC no-lineal); b) Variables de salida (MPC lineal); c) Variables de entrada (MPC no-lineal); d) Variables de entrada (MPC lineal)

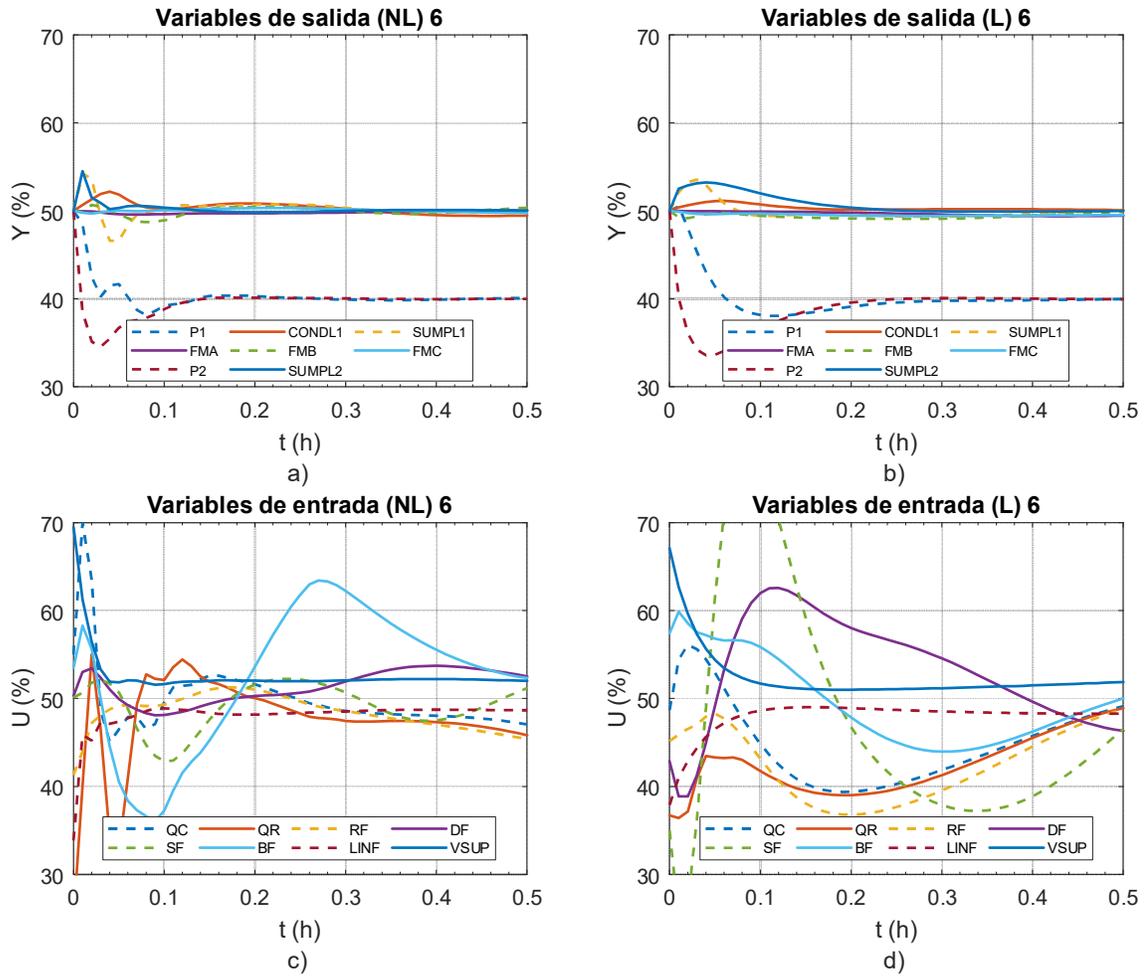


Figura 5.13: Comportamiento dinámico de las variables de entrada y salida del caso 6. a) Variables de salida (MPC no-lineal); b) Variables de salida (MPC lineal); c) Variables de entrada (MPC no-lineal); d) Variables de entrada (MPC lineal)

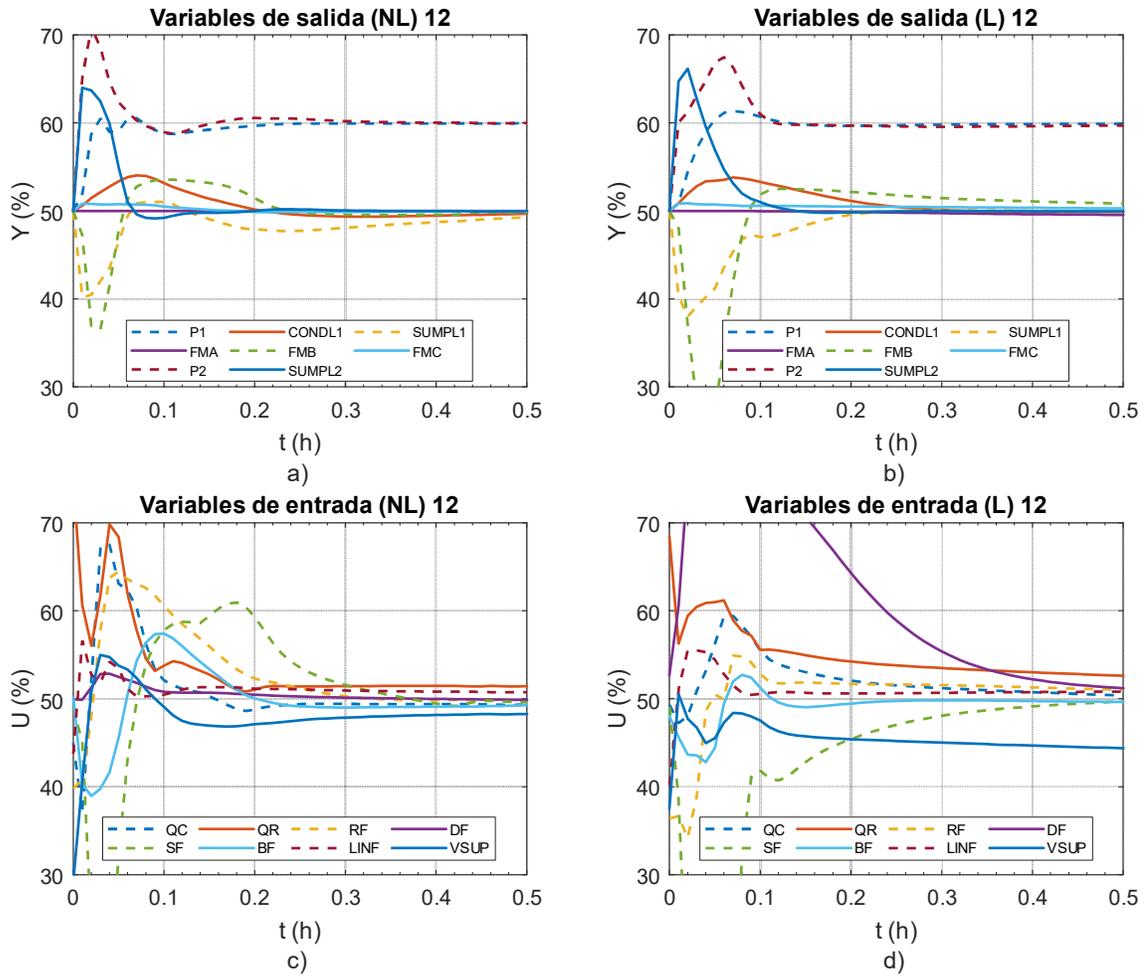


Figura 5.14: Comportamiento dinámico de las variables de entrada y salida del caso 12. a) Variables de salida (MPC no-lineal); b) Variables de salida (MPC lineal); c) Variables de entrada (MPC no-lineal); d) Variables de entrada (MPC lineal)

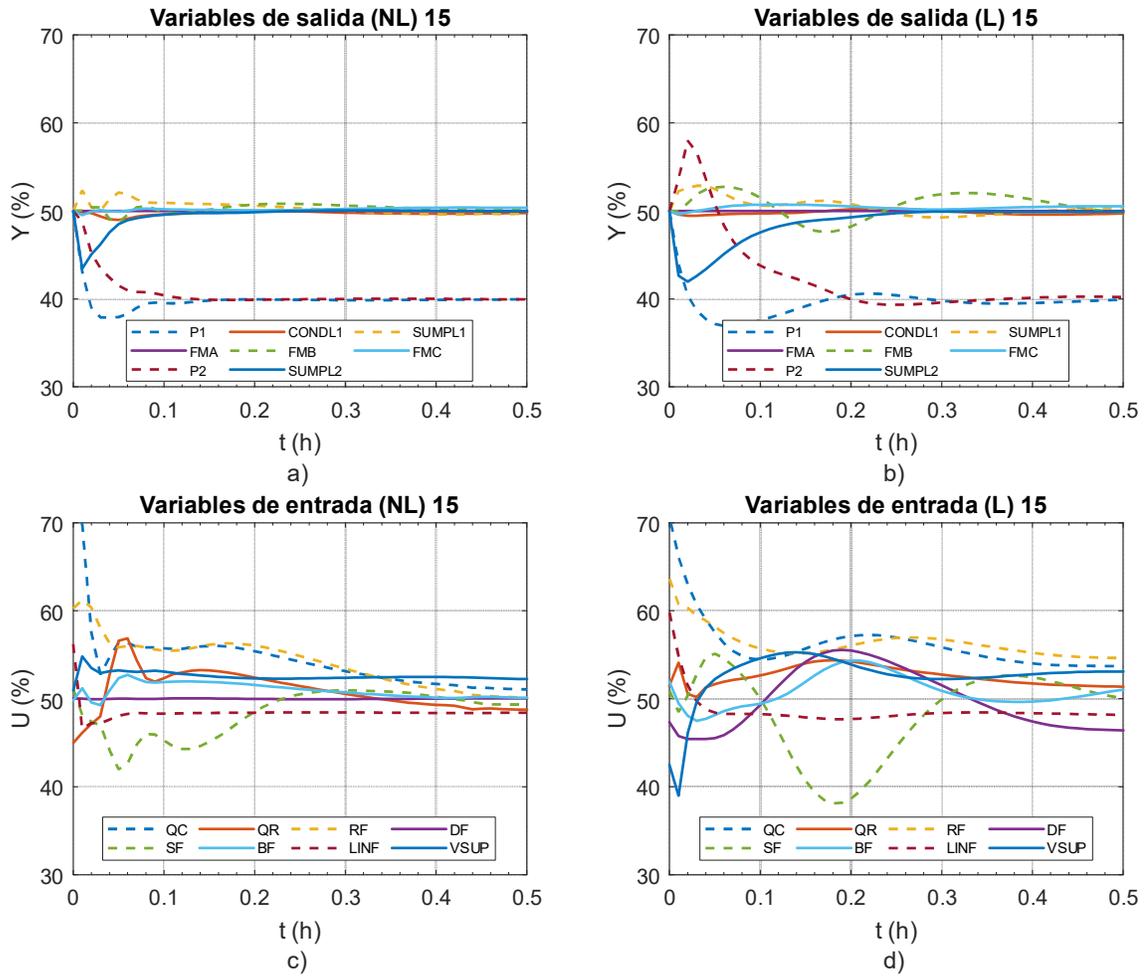


Figura 5.15: Comportamiento dinámico de las variables de entrada y salida del caso 15. a) Variables de salida (MPC no-lineal); b) Variables de salida (MPC lineal); c) Variables de entrada (MPC no-lineal); d) Variables de entrada (MPC lineal)

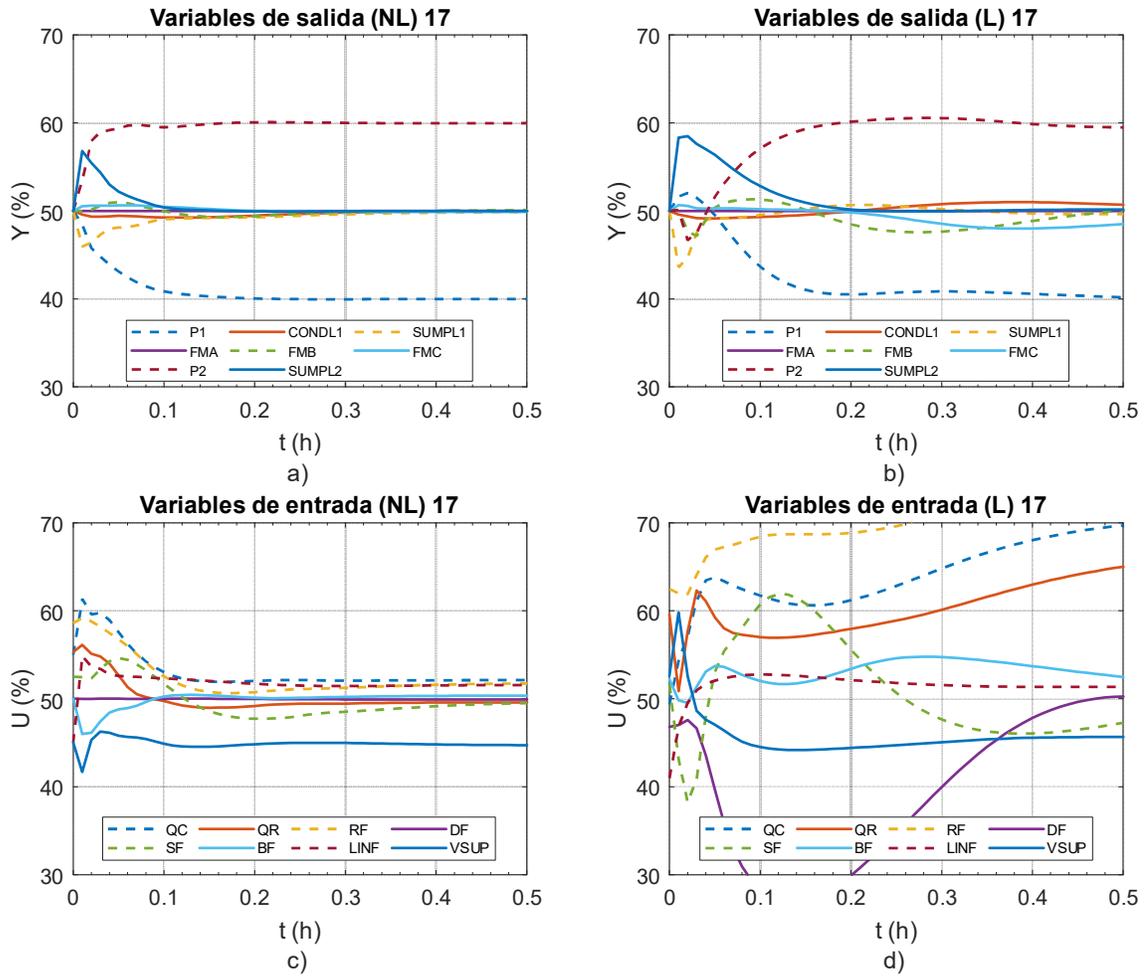


Figura 5.16: Comportamiento dinámico de las variables de entrada y salida del caso 17. a) Variables de salida (MPC no-lineal); b) Variables de salida (MPC lineal); c) Variables de entrada (MPC no-lineal); d) Variables de entrada (MPC lineal)

En el caso número 1 (Figura 5.11), ambos controladores tienen un buen desempeño y logran estabilizar la columna. En el MPC lineal los sobretiros son un poco mayores, y se observa como las variables de salida están más tiempo alejadas del *setpoint*. Esto percute en el valor del cuadrado del error total. La principal diferencia entre ambos controladores es en las acciones de control que toman. En el MPC no-lineal las variables de entrada las cuales han sido perturbadas intentan regresar a su valor en estado estable del 50 %. Las acciones de control que toma el MPC no-lineal se alejan en menor medida del 50 % en comparación con el MPC lineal. El comportamiento que toman las variables de entrada en ambos controladores es similar, principalmente en las variables QC, RF, LINF y VSUP. Las variables correspondientes a los flujos DF, SF y BF tienen comportamientos diferentes. En el MPC no-lineal la variable QR tiene mayores oscilaciones que el MPC lineal, esto en los primeros minutos, después su comportamiento se vuelve similar al del MPC lineal.

En el caso número 5 (Figura 5.12), el desempeño de ambos controladores es bueno, pero de acuerdo con las métricas de rendimiento el desempeño del MPC lineal es mejor que el del MPC no-lineal. En el MPC no-lineal el CE es mayor, al igual que TE, también el OS es más grande en la variable FMB. Aunque el rendimiento es mejor en el MPC lineal, se puede observar que en el MPC no-lineal algunas variables se estabilizan más rápido. También los cambios de *setpoint* en P1 y P2 se alcanzan más rápido en el MPC no-lineal. El rendimiento general de algunas variables es mejor en el MPC no-lineal. De manera similar al caso 1 las variables de entrada QC, RF, LINF y VSUP tienen comportamientos similares entre ambos controladores. La variable QR de nuevo presenta mayores oscilaciones al inicio de la simulación, después se sigue un comportamiento parecido al del MPC lineal. Y las principales diferencias se encuentran en como ajustan las corrientes de flujo DF, SF y BF.

En el caso número 6 (Figura 5.13), el desempeño general es mejor en el MPC no-lineal. Los valores del TE, el OS y el CE son menores en el MPC no-lineal. En la Figura 5.13 a) se observa claramente que las variables de salida son controladas de mejor manera. Los sobretiros son mayores en el MPC lineal y el mayor OS se encuentra en la variable P2. En el MPC lineal se puede observar que las variables

de entrada se alejan más del 50 %. El MPC lineal tiene más oscilaciones en las variables de entrada que el MPC no-lineal. En este caso los comportamientos de las variables son muy diferentes entre ambos controladores a excepción de LINF y VSUP.

Los casos 1, 5 y 6 corresponden a la mezcla 1. En general el controlador MPC no-lineal para estos tres casos realiza las acciones de control manteniendo las variables de entrada más cercanas al estado estable. La variable QR tiene mayores oscilaciones en el MPC no-lineal que en el MPC lineal. Ambos controladores realizan acciones de control similares en las variables LINF y VSUP, en el resto de las variables las acciones son diferentes, y solo en algunos casos tienen comportamientos parecidos. Cada controlador llega a soluciones muy distintas en la optimización interna.

En la Figura 5.14 se encuentra el caso 12, corresponde a la mezcla 2. Ambos controladores estabilizan la columna y en general el desempeño es bueno. En este caso, se tienen sobretiros considerables en varias variables de salida: SUMPL1, FMB, P2 y SUMPL2. El máximo OS lo tiene el MPC lineal en FMB. El desempeño general es mejor en el MPC no-lineal. En las variables de entrada, se puede observar como el MPC no-lineal realiza las acciones de control en menor tiempo y después se mantienen más cercanas a su valor estable. En el MPC lineal, las variables de entrada tienen movimientos más amplios y tardan más en acercarse a sus valores estables.

En el caso número 15 (Figura 5.15) el comportamiento dinámico es mejor en el MPC no-lineal, las variables de salida no parecen perturbarse mucho. Alcanzan la estabilidad en un menor tiempo. En el MPC lineal las variables de salida están más perturbadas y los cambios de *setpoint* se alcanzan en menor tiempo, en especial se puede observar en la variable P2. También los sobretiros son mayores. En este caso las variables de entrada tienen comportamientos similares, se mantienen cerca del estado estable. Las variables de flujo DF, SF y BF tienen mayores oscilaciones en el MPC lineal.

En el caso número 17 (Figura 5.16) el mejor desempeño es el del MPC no-lineal. El comportamiento dinámico tanto en las variables de salida como las de entrada se ve claramente mejor en el MPC no-lineal. Las variables de salida se mantienen más cerca del *setpoint* y son controladas en un menor tiempo. Al hacer la comparación

entre la Figura 5.16 c) y 5.16 d), claramente el MPC lineal tiene mayores dificultades para realizar el control, las variables de entrada se alejan más del 50 %. En cambio, en el MPC no-lineal se corrigen las perturbaciones y las variables de entrada tratan de regresar a los valores estables.

Para estos tres casos, correspondientes a la mezcla 2, las acciones de control son más suaves en comparaciones con los casos presentados para la mezcla 1. Las variables de salida son controladas de forma adecuada con ambos controladores. En algunos casos los sobretiros son considerables y esto depende de las perturbaciones que se le aplican a la columna.

La controlabilidad de la columna depende también de la mezcla se va a separar. Y esto se puede observar en el comportamiento dinámico que tienen las variables de entrada de cada mezcla. En algunos casos, algunas variables de entrada pueden presentar oscilaciones considerables.

Una característica de los controladores MPC es que se pueden restringir que tanto se puede mover una variable de entrada con respecto a su valor anterior. En este trabajo se enfoca en la obtención del modelo no-lineal y en la sintonización de los pesos de la función de costo del MPC. Una sintonización más completa puede incluir los parámetros correspondientes a los máximos cambios permitidos en las variables de entrada, así como también los rangos en los que se mueven las variables. De forma general estos límites y cuánto puede cambiar una variable, están dados por la naturaleza del sistema a controlar, pero también pueden ser optimizados para mejorar el rendimiento del controlador.

En la Figura 5.17 esta graficado el TE, muestra la comparación de todos los resultados del plan experimental para ambos modelos. En la mayoría de los casos el TE es menor en el MPC no-lineal. En los casos 9, 11 y 17 la diferencia entre los valores es considerable, las barras en el grafico son relativamente mucho más altas en el modelo lineal. En dos casos, el 5 y 8, el MPC lineal tiene un TE menor. En el caso 8 el TE es mucho mayor en el modelo no-lineal. Y en los casos 2, 7, 10 y 12 se puede considerar que los valores en el TE son iguales en ambos controladores.

En la Figura 5.18 está graficado el OS para todos los casos del plan experimental. En esta métrica es donde se tienen resultados parecidos entre ambos controladores. En el caso 12 es donde se tiene la diferencia más alta. En 13 casos el OS fue mayor

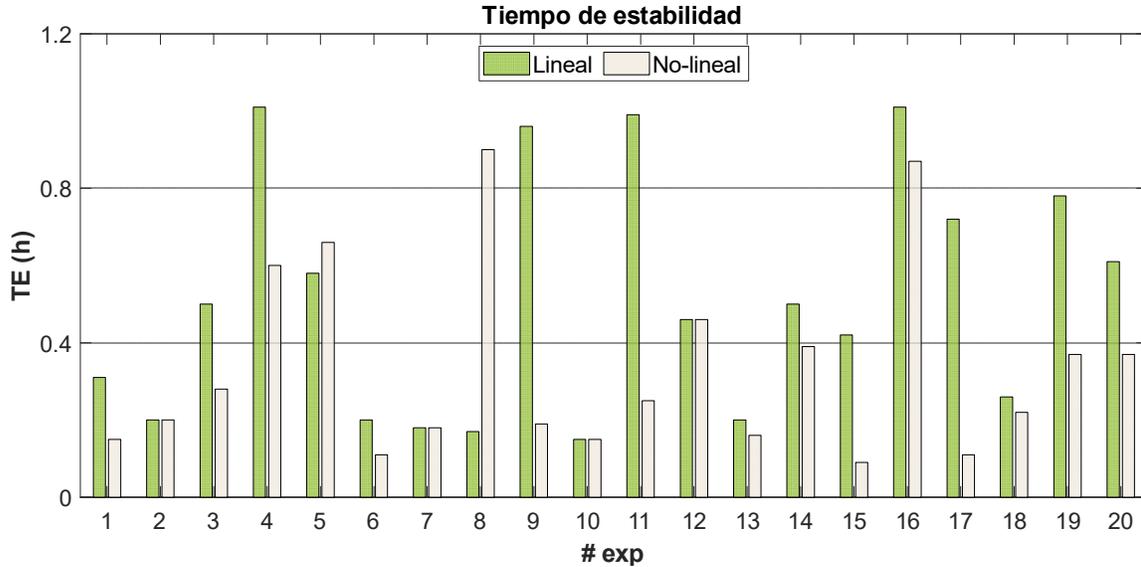


Figura 5.17: Comparación del desempeño de los controladores con modelo lineal y no-lineal en términos del tiempo de estabilidad

en el MPC lineal. En los casos 2, 3, 10 y 11 se puede considerar que tiene el mismo OS. En el resto de los casos, 3 en total, (5, 8 y 20) el MPC no-lineal tiene un OS mayor.

El CE de cada caso del plan experimental esta graficado en la Figura 5.19. El cuadrado del error total puede considerarse una métrica que generaliza más el rendimiento del MPC, aunque no significa que las otras dos métricas tengan menos importancia. En 13 de los casos el CE es menor en el MPC no-lineal. Y en 4 de estos (4, 9, 12 y 20) la diferencia entre los valores es considerable. En los casos 2, 5, 8, 10, 11, 13 y 18, el CE es menor en el MPC lineal. Los casos 2, 5, 8 y 11 tienen valores muy similares, la diferencia entre el MPC lineal y el MPC no-lineal es pequeña.

En la mayoría de los casos el desempeño es similar en los tres criterios, pero hay algunos casos en los que el desempeño mostrado en una métrica difiere del mostrado en otra. Por ejemplo, en los casos 13 y 18, de acuerdo los valores del CE, el MPC lineal tiene un mejor desempeño, pero en el TE y el OS, el desempeño del MPC no-lineal es mejor. También en el caso número 8 las combinaciones en las perturbaciones

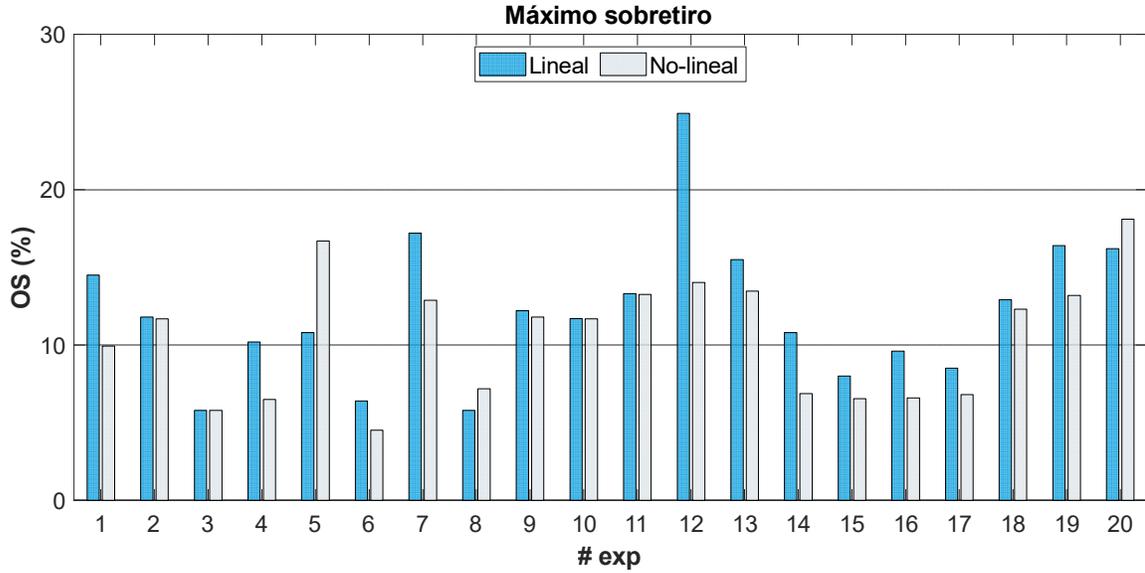


Figura 5.18: Comparación del desempeño de los controladores con modelo lineal y no-lineal en términos del máximo sobretiro

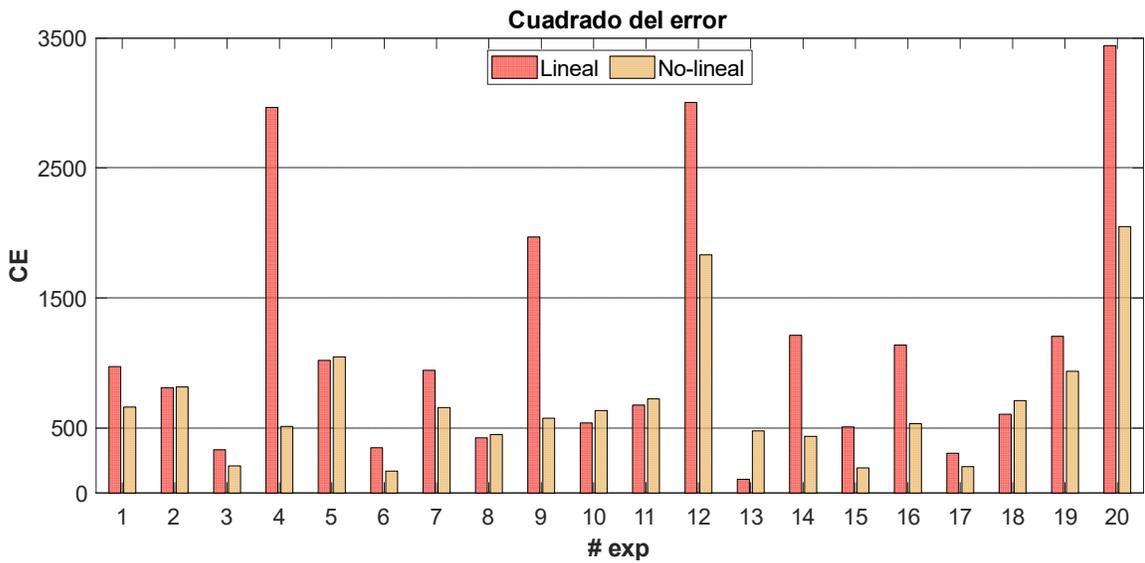


Figura 5.19: Comparación del desempeño de los controladores con modelo lineal y no-lineal en términos del cuadrado del error

Tabla 5.9: Resultados del análisis de la Prueba t-Student de los casos de estudio para los tres criterios

	TE (h)	OS (%)	CE
Media - L	0.51	12.13	1127.34
Media - NL	0.34	10.49	692.19
Varianza - L	0.10	21.02	936960.12
Varianza - NL	0.06	15.17	241087.96
Estadístico T	2.377	2.227	2.824
Probabilidad	0.014	0.019	0.005

provocan que el TE sea mayor en el MPC no-lineal, pero el desempeño general es similar al del MPC lineal tanto en el OS como en el CE. En algunos casos el MPC logra controlar la planta en un tiempo menor al reportado por el TE, al hacer la medición se toma la variable con el valor mayor y la mayoría de las variables pueden estar estabilizándose en un tiempo promedio menor. En estos casos el valor del CE y el OS puede representar mejor el desempeño del MPC. En los casos donde se aumenta la presión (+10 %) o se realiza una disminución y aumento (-10 % y +10 %) en P1 y P2 es donde se disminuye el desempeño, los valores de los criterios son altos, por ejemplo, en el caso 4, 5, 12, y 20.

Las gráficas de las Figuras 5.17 a 5.19 presentan de forma visual las diferencias entre ambos modelos, y en la mayoría de los casos el controlador con el modelo de RNA tiene un mejor desempeño. Para verificar que el modelo de RNA funciona de una mejor manera se puede utilizar una prueba t-Student para medias de dos muestras emparejadas. Con la prueba t-Student se verifica si la diferencia entre ambos modelos es significativa. La hipótesis nula es que las medias de los resultados son iguales entre los modelos y la hipótesis alternativa es que el modelo lineal tiene valores más grandes que el modelo no-lineal. Esta prueba se realiza para los tres criterios de rendimiento. Los resultados de la prueba t-Student se encuentran en la Tabla 5.9, se muestra la media de los resultados de los 20 casos de estudio, la varianza, el estadístico T y la probabilidad (significancia) obtenida.

Los valores promedio en los resultados del CE, el OS y el TE son menores con el modelo no-lineal. Los valores de la varianza también son menores en el modelo no-lineal. Los valores en los resultados de los tres criterios están más dispersados

Tabla 5.10: Diferencias entre los modelos lineal y no-lineal en los tres criterios de rendimiento

Std	TE [L-NL] (h)	OS [L-NL] (%)	CE [L-NL]
1	0.16	4.6	309.7
2	0	0.08	-5.03
3	0.22	0	124
4	0.41	3.68	2452.3
5	-0.08	-5.83	-27.1
6	0.09	1.89	179.6
7	0	4.34	285.3
8	-0.73	-1.4	-23.7
9	0.77	0.44	1393.4
10	0	0.03	-95.2
11	0.74	0.07	-48.3
12	0	10.89	1171.9
13	0.04	2	-373
14	0.11	3.93	776.8
15	0.33	1.51	314.3
16	0.14	3.04	606.5
17	0.61	1.69	103
18	0.04	0.56	-105
19	0.41	3.19	270.4
20	0.24	-1.9	1393.1

en el modelo lineal. Para el TE la probabilidad es de 0.014, para el OS es de 0.019 y para el CE es de 0.005. Las tres probabilidades tienen valores menores de 0.05 por lo que hay una diferencia significativa entre los resultados de las tres métricas de ambos modelos, lineal y no-lineal. La probabilidad más baja se encuentra en el CE. Es en el CE donde se tiene una mayor diferencia entre ambos modelos y se confirma con el valor bajo de la probabilidad. La probabilidad más alta es en el OS, la diferencia entre ambos modelos con este criterio es menor y se puede observar en la Figura 5.18 donde algunos valores son parecidos entre sí. El MPC con cada modelo tiene un desempeño diferente. La diferencia significativa en los tres criterios confirma que el modelo de RNA tiene un mejor desempeño.

La diferencia entre los modelos puede estar dada por el efecto de alguno de los fac-

Tabla 5.11: Probabilidades obtenidas del ANOVA de los casos estudios para cada métrica

	TE	OS	CE
Modelo	0.702	0.089	0.307
P1	0.572	0.768	0.753
P2	0.721	0.02	0.442
QC	0.616	0.247	0.899
QR	0.825	0.851	0.039
RF	0.148	0.065	0.163
LINF	0.395	0.091	0.171
VSUP	0.824	0.455	0.972
Mezcla	0.275	0.169	0.884

tores utilizados en las perturbaciones. Se realiza un análisis de varianza (ANOVA) para determinar los factores que influyen en la diferencia de la controlabilidad. El ANOVA se realiza con los resultados del plan experimental, con la diferencia entre ambos modelos y para cada criterio utilizado. La Tabla 5.10 muestra las diferencias entre los modelos en cada caso del plan experimental. En la Tabla 5.11 se encuentran las probabilidades (significancias) obtenidas del ANOVA.

La variable P2 en el OS tienen una probabilidad menor a 0.05. El OS esta influenciado por la P2 y posiblemente por el RF y la LINF, las siguientes variables con los valores de probabilidad más pequeños. En el CE es la variable QR la que tiene una probabilidad menor a 0.05. Estas dos variables, P2 y QR, son significativas, tienen un efecto en la diferencia entre los modelos. El resto de las variables no presenta probabilidades menores a 0.05, estas perturbaciones no tienen un efecto en la diferencia. La mezcla no tiene un efecto, su probabilidad (significancia) correspondiente no se encuentra por debajo del valor 0.05 en las tres métricas. No hay una diferencia significativa entre las dos mezclas usadas.

De acuerdo con los resultados el modelo no-lineal basado en un RNA mejora el rendimiento del MPC. Aun así, el desempeño del MPC con un modelo lineal funciona correctamente y tiene buenos resultados de control. La elección del controlador puede estar determinada por el tipo de proceso. En este caso de estudio, utilizando una columna Petlyuk, en algunos resultados el desempeño del MPC lineal es similar o mejor al del MPC no-lineal. En otros procesos químicos donde la no linealidad

tiene un mayor efecto, utilizar un modelo como el desarrollado en este trabajo puede mejorar en una mayor proporción el desempeño del control. Se demuestra que una RNA puede funcionar de forma efectiva como un modelo no-lineal. La metodología utilizada es adecuada para obtener un modelo con la robustez suficiente para ser implementada en un MPC.

Capítulo 6

Conclusiones

6.1 Conclusiones generales

En este trabajo se desarrolló un sistema de control MPC con un modelo no-lineal de RNA, utilizando una columna de destilación Petlyuk como caso de estudio. Se diseñaron dos columnas Petlyuk en Aspen Plus[®] para separar las mezclas M1 (n-butano, 2-metil-butano, n-pentano) y M2 (n-butano, n-pentano, n-octano). Las columnas fueron exportadas a Aspen Plus Dynamics[®] para trabajar con el modelo dinámico y ser controladas. Se presentaron ocho variables a controlar y ocho manipulables. Las cuales son las más comunes de controlar y manipular en columnas de destilación.

Posteriormente se entrenó una RNA para cada mezcla, la cual funciona como modelo no-lineal. Para el entrenamiento de la RNA se extrajeron datos de la dinámica de cada columna desde Aspen Plus Dynamics[®]. Los cuales se obtuvieron utilizando un diseño de superficie de respuesta y se aumentó con puntos con valores escogidos aleatoriamente. El entrenamiento se realizó con el método SOS. Cada variable obtuvo el número de neuronas adecuado, en la mezcla 1 se requirieron alrededor de 3000 a 5000 neuronas y para la mezcla 2 alrededor de 3000 y 4000. El entrenamiento de la RNA logró modelar correctamente los datos dinámicos. La mayoría de los coeficientes de determinación de cada variable tienen un valor entre 0.8 y 0.9, por lo que el ajuste es relativamente bueno. La RNA mostró una predicción adecuada para poder ser utilizada como modelo no-lineal. La RNA alcanzó un ho-

rizonte de predicción alrededor de 0.2 y 0.3 h. La precisión de la predicción de la RNA disminuye a largo plazo, utilizar una estructura diferente de la RNA podría aumentar la precisión, se puede considerar esta opción para un trabajo futuro.

Para comparar el desempeño del MPC con modelo de RNA, se desarrolló un MPC con un modelo lineal LTI. Las matrices del modelo se obtuvieron con la herramienta CDI de Aspen Plus Dynamics[®]. Posteriormente se redujo el orden del modelo mediante la descomposición en valores singulares.

En Matlab[®] se implementaron ambos modelos no-lineal y lineal en un MPC. La simulación del controlador se realizó en Simulink[®] en conjunto con Aspen Plus Dynamics[®]. Los pesos de la función de costo de cada MPC se sintonizaron mediante un plan experimental y optimizando una función objetivo. En cada caso del plan experimental se aplicaron perturbaciones con valores aleatorios a la columna. En la sintonización se utilizaron las métricas de rendimiento: el tiempo de estabilidad, el máximo sobretiro y el cuadrado del error total. Se combinaron las tres métricas en una ponderación adecuada. Los resultados mostraron que, utilizando el cuadrado del error total en conjunto con otra de las métricas, se pueden sintonizar de manera adecuada los pesos del MPC.

Para analizar el funcionamiento de cada MPC y comparar los modelos, a la columna Petlyuk, se le aplicaron múltiples perturbaciones y cambios de *setpoint*. Las combinaciones de estas perturbaciones se basaron en un plan experimental de 20 casos de estudio. Este plan se aplicó a cada MPC con modelo lineal y no-lineal. De cada caso de estudio se obtuvo el comportamiento dinámico de las variables a controlar y las acciones de control que aplicó el MPC en las variables manipulables. Para cuantificar el rendimiento se mide el tiempo de estabilidad, el máximo sobretiro y se calcula el cuadrado del error total.

Los resultados muestran un mejor rendimiento del MPC no-lineal frente a el MPC lineal. En 14 casos del plan experimental el TE fue menor en el modelo no-lineal y en 4 fueron similares entre ambos modelos. En 13 de los casos, el OS fue menor en el modelo no-lineal y en 3 casos se obtuvieron valores similares. Para el CE en 13 casos se obtuvieron valores menores en el modelo no-lineal y en 4 casos valores similares. La implementación del modelo de RNA aumentó la robustez del sistema de control. Para verificar que la diferencia entre ambos modelos fue

significativa y que el modelo no-lineal funcionó de mejor manera, se utilizó una prueba t-Student para medias de dos muestras emparejadas. Las probabilidades obtenidas de la prueba fueron: 0.014 para el TE, 0.019 para el OS y 0.005 para el CE. Estas fueron menores a 0.05 por lo que el modelo no-lineal tuvo un mejor rendimiento por una diferencia significativa. Otro estudio realizado fue el ANOVA con las diferencias entre los modelos de las respuestas del plan experimental. La variable P2 en el OS y la variable QR en el CE tuvieron un efecto en la diferencia entre los modelos. Sus valores de probabilidad fueron significativos. La mayoría de las variables utilizadas como perturbaciones no presentaron probabilidades menores a 0.05. El ANOVA también demostró que no hay una diferencia significativa entre las dos mezclas usadas. El modelo lineal a pesar de tener un rendimiento menor funcionó de una manera correcta cumpliendo con los objetivos del control. Ambos controladores con diferente modelo (lineal y no-lineal) lograron estabilizar la columna Petlyuk y cumplir con los cambios de *setpoint*. Los controladores cuentan con una robustez considerable ya que lograron controlar ocho variables a la vez, mientras se perturbaban otras 5 y se realizaban 2 cambios de *setpoint*.

Se demuestra que una RNA puede funcionar de manera correcta como un modelo dinámico de un proceso químico. La capacidad de predicción que puede obtener utilizando el método de entrenamiento SOS le permitió ser implementada en un sistema de control MPC. Se comprobó que los controladores MPC son adecuados para equipos como una columna de destilación Petlyuk. La metodología expuesta en este trabajo puede ser aplicada para controlar diferentes procesos en la industria química que contengan muchas variables de proceso y comportamientos no-lineales. La elección del tipo modelo y su complejidad en un MPC pueden estar determinados por el tipo de proceso y por la precisión que se requiera. Utilizar un modelo no-lineal en un MPC puede llegar a mejorar la robustez del sistema de control y garantizar que los productos del proceso cumplan con las especificaciones de calidad además de obtener un ahorro en energía y costos.

6.2 Trabajo futuro

- Desarrollar estrategias MPC para diferentes procesos químicos con diferentes grados de no linealidad, como por ejemplo para: reactores químicos o columnas de destilación reactivas.
 - Estudiar diferentes arquitecturas de redes neuronales artificiales utilizadas en control. Por ejemplo: redes neuronales recurrentes, redes neuronales de tipo Long short-term memory (LSTM) o redes transformers.
 - Realizar un análisis más completo utilizando factores energéticos y económicos.
 - Desarrollar e implementar un esquema de control MPC en un sistema real y verificar su aplicabilidad.

Bibliografía

- [1] N. ABDULLAH, M. A. MOHD RAZALI, M. H. O. AHMED, M. Z. NUAWI, M. M. MUSTAFA, Z. MOHD NOPIAH, A. MOHAMED, AND A. B. MOHAMAD. Neural network adaptive real-time optimizing control of industrial processes. *Chemical Engineering Research Bulletin* **19**, 129 (2017).
- [2] R. AHMAD, J. MELSHEIMER, F. C. JENTOFT, AND R. SCHLÖGL. Isomerization of n-butane and of n-pentane in the presence of sulfated zirconia: Formation of surface deposits investigated by in situ UV-vis diffuse reflectance spectroscopy. *Journal of Catalysis* **218**(2), 365–374 (2003).
- [3] V. A. AKPAN AND G. D. HASSAPIS. Nonlinear model identification and adaptive model predictive control using neural networks. *ISA Transactions* **50**(2), 177–194 (2011).
- [4] N. ASPRION AND G. KAIBEL. Dividing wall columns: Fundamentals and recent advances. *Chemical Engineering and Processing: Process Intensification* **49**(2), 139–146 (2010).
- [5] H. CANO-ROCHA AND R. GONZALEZ-GARCIA. “Estudio paramétrico de las relaciones de reflujo en columnas de destilación tipo petlyuk”. Tesis de maestría en ciencias en ingeniería química, Universidad Autónoma de San Luis Potosí (2017).
- [6] H. CANO-ROCHA AND R. GONZALEZ-GARCIA. Stochastic One-Step Training for Feedforward Artificial Neural Networks. *Neural Processing Letters* **52**(3), 2021–2041 (2020).

-
- [7] Z. CHEN AND R. AGRAWAL. Classification and comparison of dividing walls for distillation columns. *Processes* **8**(6) (2020).
- [8] J. CHINPRASIT AND C. PANJAPORNPON. Model predictive control of vinyl chloride monomer process by Aspen Plus Dynamics and MATLAB/Simulink co-simulation approach. *IOP Conference Series: Materials Science and Engineering* **778**(1) (2020).
- [9] C. CUI, Q. ZHANG, X. ZHANG, AND J. SUN. Eliminating the vapor split in dividing wall columns through controllable double liquid-only side-stream distillation configuration. *Separation and Purification Technology* **242**(March), 116837 (2020).
- [10] G. CYBENKO. Approximation by superpositions of a sigmoidal function. *Mathematics of Control, Signals, and Systems* **2**(4), 303–314 (dec 1989).
- [11] I. DEJANOVIĆ, L. MATIJAŠEVIĆ, AND Ž. OLUJIĆ. Dividing wall column—A breakthrough towards sustainable distilling. *Chemical Engineering and Processing: Process Intensification* **49**(6), 559–580 (2010).
- [12] M. FAKHROESLAM, R. BOZORGMEHRY BOOZARJOMEHRY, A. M. SAHLODIN, G. SIN, AND S. S. MANSOURI. Dynamic Simulation of Natural Gas Transmission Pipeline Systems through Autoregressive Neural Networks. *Industrial and Engineering Chemistry Research* **60**(27), 9851–9859 (2021).
- [13] S. GAIL, R. F. CRACKNELL, D. CORRIGAN, A. FESTA, V. SHANKAR, B. POULET, G. LOVETT, R. D. BÜTTGEN, K. A. HEUFER, R. MARICONTI, M. CUCCHI, AND F. MORTELLARO. Evaluating a novel gasoline surrogate containing isopentane using a rapid compression machine and an engine. *Proceedings of the Combustion Institute* **38**(4), 5643–5653 (2021).
- [14] X. GE, B. WANG, X. YANG, Y. PAN, B. LIU, AND B. LIU. Fault detection and diagnosis for reactive distillation based on convolutional neural network. *Computers and Chemical Engineering* **145**(xxxx), 107172 (2021).

- [15] D. E. GÓNGORA-PANTOJA AND R. GONZALEZ-GARCIA. “Diseño de control por modelo predictivo en columnas Petlyuk de mezclas ternarias mediante cosimulación Matlab-Aspen”. Tesis de maestría en ciencias en ingeniería química, Universidad Autónoma de San Luis Potosí (2018).
- [16] G. R. HARVIANTO, K. H. KIM, K. J. KANG, AND M. LEE. Optimal operation of a dividing wall column using an enhanced active vapor distributor. *Chemical Engineering Research and Design* **144**, 512–519 (2019).
- [17] E. HEIDARI, A. DAEICHIAN, M. A. SOBATI, AND S. MOVAHEDIRAD. Prediction of the droplet spreading dynamics on a solid substrate at irregular sampling intervals: Nonlinear Auto-Regressive eXogenous Artificial Neural Network approach (NARX-ANN). *Chemical Engineering Research and Design* **156**, 263–272 (2020).
- [18] K. JEON, S. YANG, D. KANG, J. NA, AND W. B. LEE. Development of surrogate model using CFD and deep neural networks to optimize gas detector layout. *Korean Journal of Chemical Engineering* **36**(3), 325–332 (2019).
- [19] Z. JIANG AND R. AGRAWAL. Process intensification in multicomponent distillation: A review of recent advancements. *Chemical Engineering Research and Design* **147**, 122–145 (2019).
- [20] K. KIŠ AND M. KLAUČO. Neural network based explicit MPC for chemical reactor control. *Acta Chimica Slovaca* **12**(2), 218–223 (2019).
- [21] A. A. KISS AND C. S. BILDEA. A control perspective on process intensification in dividing-wall columns. *Chemical Engineering and Processing: Process Intensification* **50**(3), 281–292 (2011).
- [22] A. A. KISS AND R. R. REWAGAD. Energy efficient control of a BTX dividing-wall column. *Computers and Chemical Engineering* **35**(12), 2896–2904 (2011).
- [23] B. KONG, Q. ZHANG, C. CUI, AND J. SUN. Optimal design and effective control of Kaibel column with liquid-only transfer streams for quaternary distillation. *Separation and Purification Technology* **250**(April), 117261 (2020).

- [24] H. KWON, K. C. OH, Y. CHOI, Y. G. CHUNG, AND J. KIM. Development and application of machine learning-based prediction model for distillation column. *International Journal of Intelligent Systems* **36**(5), 1970–1997 (2021).
- [25] G. LI, J. CHEN, B. SONG, S. WANG, R. PAN, L. PEI, S. LIU, AND Q. Y. YANG. Separation of toluene from benzene derivatives and extraction of toluene from water based on a flexible naphthalene diimide coordination network. *Separation and Purification Technology* **256**(September 2020), 117781 (2021).
- [26] P. S. MALI, B. K. JAGTAP, AND K. P. PAWAR. Nonlinear process identification and model predictive control using neural network. *International Journal of Engineering and Technology* **4**(5), 341–348 (2012).
- [27] M. NIKOLAOU. Model Predictive Controllers : a Critical. **26**, 131–204 (2001).
- [28] L. M. OCHOA-ESTOPIER, M. JOBSON, AND R. SMITH. Operational optimization of crude oil distillation systems using artificial neural networks. *Computers and Chemical Engineering* **59**, 178–185 (2013).
- [29] M. POLIGNANO. “Analogies between Internal Model Control and Predictive Control algorithms”. Master thesis in chemical engineering, University of Pisa (2015).
- [30] T. PÖPKEN AND J. GMEHLING. Simple Method for Determining the Location of Distillation Region Boundaries in Quaternary Systems. *Industrial and Engineering Chemistry Research* **43**(3), 777–783 (2004).
- [31] X. QIAN, K. HUANG, S. JIA, H. CHEN, Y. YUAN, L. ZHANG, AND S. WANG. Composition/temperature cascade control for a Kaibel dividing-wall distillation column by combining PI controllers and model predictive control integrated with soft sensor. *Computers and Chemical Engineering* **126**, 292–303 (2019).
- [32] J. B. RAWLINGS AND C. T. MARAVELIAS. Bringing new technologies and approaches to the operation and control of chemical process systems. *AIChE Journal* **65**(6), 1–18 (2019).

-
- [33] R. R. REWAGAD AND A. A. KISS. Dynamic optimization of a dividing-wall column using model predictive control. *Chemical Engineering Science* **68**(1), 132–142 (2012).
- [34] J. RÚA AND L. O. NORD. Optimal control of flexible natural gas combined cycles with stress monitoring: Linear vs nonlinear model predictive control. *Applied Energy* **265**(March), 114820 (2020).
- [35] A. S. SARAPKA, M. A. MARTINS, AND D. ODLOAK. Stable distributed MPC with zone control and input targets. *Computers and Chemical Engineering* **155**, 107507 (2021).
- [36] N. SHARMA AND K. SINGH. Model predictive control and neural network predictive control of TAME reactive distillation column. *Chemical Engineering and Processing: Process Intensification* **59**, 9–21 (2012).
- [37] Y. SHIN, R. SMITH, AND S. HWANG. Development of model predictive control system using an artificial neural network: A case study with a distillation column. *Journal of Cleaner Production* **277**, 124124 (2020).
- [38] Z. WU, D. RINCON, AND P. D. CHRISTOFIDES. Process structure-based recurrent neural network modeling for model predictive control of nonlinear processes. *Journal of Process Control* **89**, 74–84 (2020).
- [39] X. YANG, J. GUAN, L. DING, Z. YOU, V. C. LEE, M. R. MOHD HASAN, AND X. CHENG. Research and applications of artificial neural network in pavement engineering: A state-of-the-art review. *Journal of Traffic and Transportation Engineering (English Edition)* **8**(6), 1000–1021 (2021).
- [40] Z. ZHANG, Z. WU, D. RINCON, AND P. D. CHRISTOFIDES. Real-time optimization and control of nonlinear processes using machine learning. *Mathematics* **7**(10), 1–25 (2019).

Apéndice A

Códigos de Matlab

A.1 Modelo de RNA

A.1.1 Obtención de datos

MPCsimulacion_OL_c.m

```
clc; clear
[y,Fs] = audioread('Fin2.mp3');%Alarma
load('de_ol_D7_v3.mat')% Cargar diseño de experimentos
for i=1:length(de_ol)%Numero de Experimento
disp(i)
ex=de_ol(i,:); %Experimento
mez=1; %Mezcla
addpath(['M',sprintf('%G',mez)])%Folder de la mezcla
[U_nom,Y_nom,U_nom_pct,FX] = OLsimulacion_datos(mez); %Datos variables
nt=200;
Ent=[repmat(ex,nt,1);repmat([0,0,0,0,0,0,0,0,ex(9)], . . .
round(nt/2),1);repmat([0,0,0,0,0,0,0,0],round(nt/2),1)];
Entrada=Ent+50;
disp(Ent(1,:))
Entrada = timeseries(Entrada);
Entrada.TimeInfo.Increment=0.005;
```

```
% Ejecutar simulink
T=length(Ent)*0.005;%Tiempo de simulación
model=['M',sprintf('%G',mez),'_OL']; %Open loop
load_system(model);
%open_system(model);
sim(model);
%close_system(model);
a=size(Ypct_OL);
if a(1)<length(Ent) & a(1)-5>5
    if a(1)<nt
        nt=a(1)-5;
    else
        nt=nt-5;
    end
    Ent = [repmat(ex,nt,1);repmat([0 0 0 0 0 0 0 ex(9)], . . .
    round(nt/2),1);repmat([0 0 0 0 0 0 0 0],round(nt/2),1)];
    disp('A')
    Entrada=Ent+50;
    Entrada = timeseries(Entrada);
    Entrada.TimeInfo.Increment=0.005;
    T=length(Ent)*0.005;
    sim(model);
end
%Crea los datos para la red Neuronal
u=Upct_OL;
x=Ypct_OL;
%Datos de entrada a la Red
X=[x u];
%Resultados
Resu.Umat_ing=Umat_ing; Resu.Upct_OL=Upct_OL;
Resu.Ying_OL=Ying_OL; Resu.Ypct_OL=Ypct_OL;
%Guarda los Resultados
```

```

cd('ResultadosOL')
if i==1
    XT{i}=X;
else
    load('X_n.mat')
    XT{i}=X;
end
save('X_n.mat','XT');
cd ..
%Sale del folder de la mezcla
rmpath(['M',sprintf('%G',mez)])
sound(y,Fs);
clearvars -except de de_ol y Fs XT V3 M1 i j VD7M4
end

```

A.1.2 Entrenamiento de la RNA

generadata.m

```

% Datos para la red
clear;clc
addpath('ResultadosOL')
load('X_n.mat')
%entrada
%x=XT(:,1:8)';
dtm=0.005; %Tiempo de muestreo 0.005
nh=1; %Numero de tiempos
dt=dtm*nh; %Tiempo de entrenamiento de la Red
VN=0;VE=0;X=zeros(1,17);Y=X(:,1:8);XN=X;YN=Y;XE=X;YE=Y;
tic
for i=1:length(XT)
    %disp(i)
    lx=length(XT{i}(:,1));

```

```
vdp=1:nh:lx;
x=XT{i}(vdp,:);y=x(:,1:8);
if length(x(:,1))<4
continue
end
%Derivada inicial adelante
dyi=y(2,:)-y(1,:);
dyidt=dyi/dt;
%Derivada centrada
m=1;a=0;
for j=2:length(y(:,1))-1
der=y(j+1,:)-y(j-1,:);
%——
%Dinamica natural
if a~ =1
if y(j,8) > 90 ...
& abs(der(8)) < 0.01 || y(j,8) < 10 ...
& abs(der(8)) || y(j,3) > 90 ...
& abs(der(3)) < 0.01 || y(j,3) < 10 ...
& abs(der(3))
a=1;cj=j-1;
VN=[VN;i];
%break
end
end
%——
dyc(m,:)=der;
m=m+1;
end
dycdt=dyc/(2*dt);
%Derivada Final
dyf=y(end,:)-y(end-1,:);
```

```

dyfdt=dyf/dt;
dydt=[dyidt;dycdt;dyfdt];
if a==0
cj=m;
if lx>200
%disp(m)
VE=[VE;i];m=m+1;
XE=[XE;x];YE=[YE;dydt];%Solo Estables
end
end
clearvars -except x dydt XT X Y XN YN XE YE i dt nh m VN VE cj j
X=[X;x];Y=[Y;dydt]; %Todos
XN=[XN;x(1:cj,:)];YN=[YN;dydt(1:cj,:)];%Parte Natural y Estables
end
X=X(2:end,:);Y=Y(2:end,:);XN=XN(2:end,:);YN=YN(2:end,:);
XE=XE(2:end,:);YE=YE(2:end,:);
toc

```

entrenarna.m

```

clear;clc;
addpath('Funciones_NN')
load('DataRedM4D7_T_N_E_2te_2.mat') %Datos Dinámica de la columna
Y=[Y];
X=[X];
I = randperm(numel(X(:,1)))'; %Permuta aleatoriamente
pd=1; %Porcentaje de datos
X_Tr = X(I(1:pd*numel(X(:,1))),:);Y_Tr = Y(I(1:pd*numel(X(:,1))),:);%Datos
%—
%Agregar error a los datos
pce=0.1; %porcentaje de error max
li=1-pce;ls=1+pce;
Er = li + (ls-li)*rand(length(Y_Tr),8);

```

```

Y_err=Y_Tr.*Er;
%Carpetas con funciones
Ei=[17 100 8];% Estructura inicial de la RN
nt=6500;In=200;
ptr=.8; %Porcentaje para entrenamiento
a=1;% Función de Activación
%(1) G = @(V) 2 ./ (1 + exp(-2*V)) - 1;(2) G = @(V) tanh(V)
[nn] = crnn(Ei,a);% Crea la red
[nn] = nndat(nn,X_Tr,Y_err,ptr);% Agrega Datos a la red
clearvars -except nn nt In
%[nnt] = ENTr(nn,nt,In); % Entrena la red
%——
%Entrena la red y guarda todo en la estructura: nnt
Ei=nn.Ei;ni=Ei(2);
Wr=cell(3,1);sset=zeros(3,Ei(3));ssev=sset;
r2ajt=sset;r2ajv=sset;r2t=sset;r2v=sset;
cc=1;
for n=ni:In:nt
    disp(n)
    if n==ni
        %Entrenamiento rápido
        % Calcula ultima W
        U = ones(numel(nn.D.Xe.t(:,1)),1);
        Y1= nn.G([U nn.D.Xe.t]*nn.W{1,1});
        A=[U Y1];
        clearvars -except nn A Wr sset ssev ...
        r2ajt r2ajv r2t r2v cc n ni In Ei
        Wu=(A'*A)\(A'*nn.D.Ye.t);
        nn.W{2}=Wu;
        clear A
        %nn.W = ENTrf(nn);
        Wr{cc}=nn.W;%nn.W=W;

```

```

else
%Agrega neuronas, agrega un peso
nn.W{1}(:,end+1:end+In)=( 12*(log10(Ei(1)) +1) ...
/Ei(1) )*(rand(Ei(1)+1,In)-0.5);
%Calcula W por RL
U = ones(numel(nn.D.Xe.t(:,1)),1);
Y1= nn.G([U nn.D.Xe.t]*nn.W{1,1});
A=[U Y1];
clearvars -except nn A Wr sset ssev ...
r2ajt r2ajv r2t r2v cc n ni In Ei
Wu=(A'*A)\(A'*nn.D.Ye.t);
nn.W{2}=Wu;
clear A
%nn.W = ENTrf(nn);
Wr{cc}=nn.W;%nn.W=W;
%nn.W = ENTrf(nn);Wr{cc}=nn.W;
end
%Calcula las r
[r2ajt(cc,:), r2t(cc,:),sset(cc,:)] = ...
R2aj(nn.D.Xe.t,nn.D.Ye.t,nn.W,nn.G,nn.D.ne,n);
[r2ajv(cc,:), r2v(cc,:),ssev(cc,:)] = ...
R2aj(nn.D.Xe.v,nn.D.Ye.v,nn.W,nn.G,nn.D.ne,n);
cc=cc+1;
end
%Encuentra la r2 mas grande
[r2v_op,psm] = max(r2v);
Ep=Ei;Ep(:,2)=Ep(2)+ (max(psm)-1)*In;
disp('Estructura:')
disp(Ep)
%Encuentra la W1 correspondiente
Ws{1}=Wr{max(psm)}{1};
%Hace W2-matrix amputada

```

```

for j=1:numel(psm)
    Ws{2}(1:length(Wr{psm(j)}{2}(:,j)),j)=Wr{psm(j)}{2}(:,j);
end
%Guarda los datos
nnt=nm;nnt.E=Ep;nnt.W=Ws;
nnt.Pf.nn_op=(psm-1)*In+ni;nnt.Pf.In=In;
nnt.Pf.Ps_op=psm;%Posición
nnt.Pf.r2.ajt=r2ajt;nnt.Pf.r2.t=r2t;
nnt.Pf.r2.ajv=r2ajv;nnt.Pf.r2.v=r2v;
nnt.Pf.r2.ajt_op=r2ajt(psm);nnt.Pf.r2.t_op=r2t(psm);
nnt.Pf.r2.ajv_op=r2ajv(psm);nnt.Pf.r2.v_op=r2v_op;
nnt.Pf.sse.t=sset;nnt.Pf.sse.v=ssev;
nnt.Pf.sse.t_op=sset(psm);nnt.Pf.sse.v_op=ssev(psm);
%Función RNA
nnt.NNf=@(X) mapminmax('reverse',(NNf(mapminmax('apply',X',nnt.D.Xe.PSx)',...
    nnt.W,nnt.G))',nnt.D.Ye.PSy)';
disp('R2v:')
disp(nnt.Pf.r2.v_op)
disp('# Neuronas por salida:')
disp(nnt.Pf.nn_op)
%Graficas
plotNN_r2(nnt) %R2

```

A.2 Códigos MPC no-lineal

A.2.1 Datos MPC no-lineal

MPCsimulacion_datos_P.m

```

function [U_nom,Y_nom,U_nom_pct,Y_nom_pct,MPCobj,FX] = ...
MPCsimulacion_datos_P(ex,NNn)
Mez=ex(end);
%Valores de las variables de entrada (manipulables) en estado estacionario

```

```

QC=[-80154.82101275578 -202651.3161430501];
QR=[99083.54550443194 196376.2313035084];
RF=[2451.417426634 7356.446504171];
DF=[816.7529697429853 907.0582908279695];
SF=[1142.195069532 949.1895558707999];
BF=[1726.505232725092 1163.279297292634];
LINF=[5037.160049963329 4941.698031246296];
VSUP=[35.05204357487258 46.92887663519267];
FF=45;
%Valores de las variables de salida (a controlar) en estado estacionario
P1=[2 1.999999999999995];
CONDL1=[0.4211675474880008 0.5736687760263445];
SUMPL1=[0.66599869999997451 1.001658985009278];
FMA=[0.9999653238065017 0.9860774430988239];
FMB=[0.9221477428167862 0.8524049566987729];
FMC=[0.9800888698685173 0.9245575974627911];
P2=[1.999999999999999 1.99999999999983];
SUMPL2=[0.5533799607500902 0.6176166632461076];
U_nom = [QC(Mez), QR(Mez), RF(Mez), DF(Mez), SF(Mez), BF(Mez),...
LINF(Mez), VSUP(Mez), FF];%Variables de entrada
Y_nom = [P1(Mez), CONDL1(Mez), SUMPL1(Mez), FMA(Mez), FMB(Mez),...
FMC(Mez), P2(Mez), SUMPL2(Mez)];%Variables de Salida Setpoints
%Step 2: Scale the plant signals
U_span = 2*U_nom;
Y_span = 2*Y_nom;
%Convert nominal condition from engineering units to percentages
U_nom_pct = (U_nom - zeros(1,length(U_nom)))*100./U_span;
Y_nom_pct = (Y_nom - zeros(1,length(Y_nom)))*100./Y_span;
%Diseño del MPC
MPCobj=MPCdesign(U_nom_pct,Y_nom_pct,NNn,Mez);%Controlador MPC
%Cambio de Setpoint
Y_nom_pct=Y_nom_pct+ex(3:10);

```

```

% Composiciones en la alimentación en fracción mol
FA=ex(20); %
FB=[0.335 0.28];
FC=1-FA-FB(Mez);
FX=[FA FB(Mez) FC];
end

```

A.2.2 Diseño MPC no-lineal

MPCdesign.m

```

function [MPCobj] = MPCdesign(U_nom,Y_nom,NNn,mez)
%MPC Obtiene el controlador MPC No Lineal
% U_nom 8 variables manipulables
% Y_nom 8 variables respuesta (setpoint)
%Design Model Predictive Controller
%Step 1: Create a nonlinear MPC controller object.
MPCobj = nlmpc(8, 8,'MV', 1:8, 'MD', 9);
%Step 2:
Ts = 0.01; % sample time
PH = 0.1/Ts; % prediction horizon
CH = 0.05/Ts; % control horizon
MPCobj.Ts = Ts;
% To reduce computational effort, use a short prediction horizon
MPCobj.PredictionHorizon = PH;
MPCobj.ControlHorizon = CH;
MPCobj.Model.StateFcn = @(x,u) StateFcn(x,u,NNn);
MPCobj.Model.OutputFcn = 'OutputFcn';
%—————
%Step 3: Specify weights for manipulated variables and controlled outputs
WMVR(2,:)= [0.0627403076682624,0.0710866106194889,0.123548283557411,...
0.363890573993434,0.0831242321866194,0.237954022132465,...
0.0757358306858776,0.101231138241584];

```

```
W_OV(2,:)= [1.27628107993338,1.74310870337756,1.28209974360870,...
1.72431387307407,0.359261893818625,2.37107372068142,...
1.02753333622552,2.35106469077907];
W_MVR(1,:)= [0.0329495087050633,0.0723457880127479,0.281892143220738,...
0.196721981044569,0.433851948000093,0.0744312706291346,...
0.0365210880137646,0.165683297025774];
W_OV(1,:)= [1.84274324730665,1.25296075090027,2.00437116246218,...
1.40479045886786,0.594367968278443,2.08853384837188,...
1.21173504770994,1.65676502213158];
W_MVR=W_MVR(mez,:);
W_OV=W_OV(mez,:);
MPCObj.Weights.ManipulatedVariablesRate = W_MVR;
MPCObj.Weights.OutputVariables = W_OV;
%Cosimulate MPC Controller and Nonlinear Plant
MV = MPCObj.MV;
OV = MPCObj.OV;
S = MPCObj.S;
% Physical bounds
for i = 1:8
MV(i).Min = 15;
MV(i).Max = 80;
end
for i = [1:3 7 8]
MV(i).RateMin = -15;
MV(i).RateMax = 15;
end
for i = 4:6
MV(i).RateMin = -5;
MV(i).RateMax = 5;
end
MPCObj.MV = MV;
for i = 1:8
```

```

OV(i).Min = 0;
OV(i).Max = 100;
end
for i = [2 3 8]
OV(i).Min = 25;
OV(i).Max = 75;
end
MPCobj.OV = OV;
for i = 1:8
S(i).Min = 0;
S(i).Max = 100;
end
MPCobj.S = S;
%—Optimization
MPCobj.Optimization.RunAsLinearMPC='timevarying';
% It is best practice to test your prediction model and any other custom
% functions before using them in a simulation. To do so, use the
% |validateFcns| command. In this case, use the initial operating point as
% the nominal condition for testing, setting the unmeasured disturbance
% state to |0|.
x0 = Y_nom;
u0 = U_nom;
validateFcns(MPCobj,x0,u0(1:8),u0(9));
end

```

A.2.3 Simulación en Simulink MPC no-lineal

MPCsimulacion_NL_c_P.m

```

clc; clear
[y,Fs] = audioread('Fin.mp3');%Alarma
load('dep6.mat'); % Cargar diseño de experimentos
addpath('Funciones P')%Carpeta con funciones

```

```

addpath('C:\Simulaciones Aspen\MPC NL\Redes')%Carpeta con las Redes
%Red Neuronal —
%M2
load('nnt_M2D7_i100_er01_05-55_80_T100_2t.mat')%<——
%M1
%load('nnt_M1D7_i100_er01_35-60_80_T100_2t_2.mat')%<——
NN.W=nnt.W;NN.G=nnt.G;
NN.minX=nnt.D.Xe.PSx.xmin';NN.maxX=nnt.D.Xe.PSx.xmax';
NN.minY=nnt.D.Ye.PSy.xmin';NN.maxY=nnt.D.Ye.PSy.xmax';
for i=1:20 %Numero de Experimento
disp(i)
ex=de(i,:); %Experimento
fprintf('sp = %5.3f, %5.3f, %5.3f, %5.3f, %5.3f, %5.3f, %5.3f, %5.3f\n',... ex(3:10))
fprintf('pct = %5.3f, %5.3f, %5.3f, %5.3f, %5.3f, %5.3f, %5.3f, %5.3f\n',... ex(11:18))
fprintf('con = %5.3f, %5.3f\n', ex(19:20))
%Entra al folder de la mezcla
addpath(['M',sprintf('%G',ex(end))])
[U_nom,Y_nom,U_nom_pct,Y_nom_pct,MPCobj,FX] = ...
MPCsimulacion_datos_P(ex,NN);
pctNL=ex(11:18);%Perturbaciones
% Ejecutar simulink a lazo cerrado
T=1;%Tiempo de simulación
model=['M',sprintf('%G',ex(end)),'_CL_NL']; %Close loop
load_system(model);
%open_system(model);
rel1=clock;fix(rel1); % Hora y fecha simulación
tic
sim(model);
toc
%close_system(model);
%Resultados
Resu.U_nom=U_nom; Resu.Upct_CL_MPC=Upct_CL_MPC;

```

```

Resu.Upct_CL=Upct_CL; Resu.Y_nom=Y_nom;
Resu.Ying=Ying; Resu.Ypct_NL=Ypct_NL;
Resu.Umat_ing=Umat_ing;
%Genera y guarda las figuras
Plot_Resf(i,Ying,time,Ypct_NL.data,Upct_CL,Resu);
%Tiempo de estabilidad
[testab,tol] = Testabilidad(ex,Ypct_NL.data,Ying);
%Posición Setpoint
[LSP] = LSP_posicion(ex(3:end),Ypct_NL.data);
%Variación máxima de cada variable de salida
[tolmm,tolm] = Varmax(ex(1:2),tol,LSP);
%Error global cuadrado de variables de salida
egcy = eglobal(ex(1:2),tol,LSP);
%Tabla de resultados
[TR] = Tresultados(ex,tolm,tolmm,testab,egcy,rell);
%Sale del folder de la mezcla
rmpath(['M',sprintf('%G',ex(end))])
sound(y,Fs);
clearvars -except model de y Fs Tiempo M NN
end
%close_system(model);

```

A.3 Códigos MPC lineal

A.3.1 Modelo lineal y diseño de MPC

MPCdesign.m

```

function [MPCobj] = MPCdesign(U_nom,Y_nom,ex)
%MPC Optiene el controlardos MPC
% U_nom 9 variables manipulables
% Y_nom 8 variables de respuesta (setpoint)
%—————

```

```

%Step 1: Convert the CDI-generated sparse-matrices to a state-space model.
load cdi_A.dat cdi_A
load cdi_B.dat cdi_B
load cdi_C.dat cdi_C
A = full(spconvert(cdi_A));
B = full(spconvert(cdi_B));
C = full(spconvert(cdi_C));
D = zeros(8,9);
%additional checks to make sure A, B, and C have the correct dimensions
[nxAr, nxAc] = size(A);
[nxB,nu] = size(B);
[ny,nxC] = size(C);
nx = max([nxAr, nxAc, nxB, nxC]);
if nx > nxC
    C = [C, zeros(ny,nx-nxC)];
end
if nx > nxAc
    A = [A zeros(nxAr,nx-nxAc)];
end
if nx > nxAr
    nxAc = size(A,2);
    A = [A; zeros(nx-nxAr, nxAc)];
end
if nxB < nx
    B = [B; zeros(nx-nxB,nu)];
end
%Step 2: Scale the plant signals
U_span = 2.*U_nom;
U_zero = zeros(1,9);
Y_span = 2.*Y_nom;
Y_zero = [0, 0, 0, 0, 0, 0, 0, 0, 0];
U_nom_pct = (U_nom - U_zero)*100./U_span; % Convert engineering units

```

```

Y_nom_pct = (Y_nom - Y_zero)*100./Y_span;
%Scale the B and C matrices
BB = B.*(ones(nx,1)*U_span);
CC = C./(ones(nx,1)*Y_span)';
%Step 3: Define the state-space plant model
G = ss(A, BB, CC, D);
G.TimeUnit = 'hours';
G.u = {'QC','QR','RF','DF','SF','BF', 'LINF', 'VSUP', 'FF'};
G.y = {'P1','COND1','SUMPL1','FMA','FMB','FMC','P2', 'SUMPL2'};
%Step 4: Reduce model order.
[hsv, baldata] = hsvd(G);
order = find(hsv>0.01,1,'last');
Options = balredOptions('StateElimMethod', 'Truncate');
G = balred(G, order, baldata, Options);
%Design Model Predictive Controller
%Step 1: Augment the plant to model unmeasured load disturbances
Gmpc = ss(G.a, G.b(:,1:9,1:8), G.c,zeros(8,17),'TimeUnit','hours');
InputName = cell(1,11);
for i = 1:8
InputName{i} = G.InputName{i};
InputName{i+9} = [G.InputName{i}, '-UD'];
end
InputName{9} = G.InputName{9};
Gmpc.InputName = InputName;
Gmpc.InputGroup = struct('MV', 1:8, 'MD', 9, 'UD', 10:17);
Gmpc.OutputName = G.OutputName;
%—————
load PESOS_Ord.mat PesOrd
Ts = 0.01; % sample time
PH = 0.2/Ts; % prediction horizon
CH = 0.1/Ts; % control horizon
MPCobj = mpc(Gmpc, Ts, PH, CH); % MPC object

```

```
%Step 3: Specify weights
MPCobj.Weights.ManipulatedVariablesRate = PesOrd(1,11:18);
MPCobj.Weights.OutputVariables = PesOrd(1,3:10);
%Step 4: Specify nominal plant input/output values
MPCobj.Model.Nominal.U = [U_nom_pct'; zeros(8,1)];
MPCobj.Model.Nominal.Y = Y_nom_pct';
EstGain = 25; % factor of 25
Gd = getindist(MPCobj); % get default input disturbance model
Gd_new = EstGain*Gd; % create new input disturbance model
setindist(MPCobj, 'Model', Gd_new); % set input disturbance model
MV = MPCobj.MV;
OV = MPCobj.OV;
S = MPCobj.S;
% Physical bounds
for i = 1:8
    MV(i).Min = 15;
    MV(i).Max = 80;
end
for i = 1:8
    MV(i).RateMin = -15;
    MV(i).RateMax = 15;
end
for i = 4:6
    MV(i).RateMin = -5;
    MV(i).RateMax = 5;
end
MPCobj.MV = MV;
for i = 1:8
    OV(i).Min = 0;
    OV(i).Max = 100;
end
for i = [2 3 8]
```

```

OV(i).Min = 25;
OV(i).Max = 75;
end
MPCobj.OV = OV;
for i = 1:8
S(i).Min = 0;
S(i).Max = 100;
end
MPCobj.S = S;
end

```

A.3.2 Datos MPC lineal

MPCsimulacion_datos_P.m

```

function [U_nom,Y_nom,U_nom_pct,Y_nom_pct,MPCobj,FX] = ...
MPCsimulacion_datos_P(ex)
Mez=ex(end);
%Valores de las variables de entrada (manipulables) en estado estacionario
QC=[-80154.82101275578 -202651.3161430501];
QR=[99083.54550443194 196376.2313035084];
RF=[2451.417426634 7356.446504171];
DF=[816.7529697429853 907.0582908279695];
SF=[1142.195069532 949.1895558707999];
BF=[1726.505232725092 1163.279297292634];
LINF=[5037.160049963329 4941.698031246296];
VSUP=[35.05204357487258 46.92887663519267];
FF=45;
%Valores de las variables de salida (a controlar) en estado estacionario
P1=[2 1.999999999999995];
CONDL1=[0.4211675474880008 0.5736687760263445];
SUMPL1=[0.6659986999997451 1.001658985009278];
FMA=[0.9999653238065017 0.9860774430988239];

```

```

FMB=[0.9221477428167862 0.8524049566987729];
FMC=[0.9800888698685173 0.9245575974627911];
P2=[1.999999999999999 1.999999999999983];
SUMPL2=[0.5533799607500902 0.6176166632461076];
U_nom = [QC(Mez), QR(Mez), RF(Mez), DF(Mez), SF(Mez), BF(Mez),...
LINF(Mez), VSUP(Mez), FF];%Variables de entrada
Y_nom = [P1(Mez), CONDL1(Mez), SUMPL1(Mez), FMA(Mez), FMB(Mez),...
FMC(Mez), P2(Mez), SUMPL2(Mez)];%Variables de Salida Setpoints
%Step 2: Scale the plant signals
U_span = 2*U_nom;
Y_span = 2*Y_nom;
%Convert nominal condition from engineering units to percentages
U_nom_pct = (U_nom - zeros(1,length(U_nom)))*100./U_span;
Y_nom_pct = (Y_nom - zeros(1,length(Y_nom)))*100./Y_span;
%Diseño del MPC
MPCobj=MPCdesign(U_nom,Y_nom,ex);%Controlador MPC
% Composiciones en la alimentación en fracción mol
FA=ex(20); %
FB=[0.335 0.28];
FC=1-FA-FB(Mez);
FX=[FA FB(Mez) FC];
end

```

A.3.3 Simulación en Simulink MPC lineal

MPCsimulacion_c_P.m

```

clc; clear
[y,Fs] = audioread('Fin.mp3');%Alarma
load('dep6.mat'); % Cargar diseño de experimentos
addpath('Funciones P')%Carpeta con funciones
for i=1:20 %Numero de Experimento
disp(i)

```

```

ex=de(i,:); %Experimento
fprintf('sp = %5.3f, %5.3f, %5.3f, %5.3f, %5.3f, %5.3f, %5.3f, %5.3f\n',... ex(3:10))
fprintf('pct = %5.3f, %5.3f, %5.3f, %5.3f, %5.3f, %5.3f, %5.3f, %5.3f\n',... ex(11:18))
fprintf('con = %5.3f, %5.3f\n', ex(19:20))
%Entra al folder de la mezcla
addpath(['M',sprintf('%G',ex(end))])
[U_nom,Y_nom,U_nom_pct,Y_nom_pct,MPCobj,FX] = ...
MPCsimulacion_datos_P(ex);
pctNL=ex(11:18);%Perturbaciones
T=1;%Tiempo de simulación
model=['M',sprintf('%G',ex(end)),'_CL']; %Close loop
load_system(model);
%open_system(model);
rel1=clock;fix(rel1); % Hora y fecha simulación
tic
sim(model);
toc
%close_system(model);
%Resultados
Resu.U_nom=U_nom; Resu.Upct_CL_MPC=Upct_CL_MPC;
Resu.Upct_CL=Upct_CL; Resu.Y_nom=Y_nom;
Resu.Ying=Ying; Resu.Ypct_NL=Ypct_NL;
Resu.Umat_ing=Umat_ing;
%Genera y guarda las figuras
Plot_Resf(i,Ying.time,Ypct_NL.data,Upct_CL.data,Resu);
%Tiempo de estabilidad
[testab,tol] = Testabilidad(ex,Ypct_NL.data,Ying);
%Posición Setpoint
[LSP] = LSP_posicion(ex(3:end),Ypct_NL.data);
%Variación máxima de cada variable de salida
[tolmm,tolm] = Varmax(ex(1:2),tol,LSP);
%Error global cuadrado de variables de salida

```

```

egcy = eglobal(ex(1:2),tol,LSP);
%Tabla de resultados
[TR] = Tresultados(ex,tolm,tolmm,testab,egcy,rell);
%Sale del folder de la mezcla
rmpath(['M',sprintf('%G',ex(end))])
sound(y,Fs);
clearvars -except model de y Fs Tiempo M NN
end
%close_system(model);

```

A.4 Ajuste de los pesos del MPC

A.4.1 Sintonización de los pesos

Al_w_pnd.m

```

%Análisis Pesos Ponderaciones
%Se obtienen los pesos para cada ponderación
%Función Objetivo completa
clear; clc
load('W_Y_M4_L_fo_v_c.mat')%Resultados Simulación Diseño de exp
addpath('Funciones')%Carpeta con funciones
load('DM_DE.mat')%Diseño de mezclas Design Expert
d=load('BB_16_DE.mat');% Design Expert Diseño de Superficie
BB=d.dBB;
xop=zeros(length(DM),16);xme=xop;
for i=1:length(DM)
    Y=RY*DM(i,:);
    Ya(:,i)=Y;
    [~,I] = min(Y);
    Yim(i,:)=Yi(I);
    xme(i,:)=BB(Yi(I),:);
end

```

```

for i=1:length(DM)
    disp(i)
    Y=RY*DM(i,:);
    [Res{i},xop(i,:),yop(i,:),~,r2(i),~,Yp(:,i)] = fpesos_fode_C(Y,Yi);%Opt
end
r2=r2';
xmean=mean(xop);
Base=zeros(1,16);Dif=1;
xmer=[exp((Dif*xme(:,1:8)+Base(:,1:8)).*log(5)) ...
    exp((Dif*xme(:,9:16)+Base(:,9:16)).*log(5))./10];
xopr=[exp((Dif*xop(:,1:8)+Base(:,1:8)).*log(5)) ...
    exp((Dif*xop(:,9:16)+Base(:,9:16)).*log(5))./10];

```

fpesos_fode_C.m

```

function [Res,xop,yop,B,r2,GLE,Yp] = fpesos_fode_C(Y,Yi)
    %Análisis Pesos
    %Función Objetivo completa
    j=1;
    d=load('BB_16_DE.mat');% Design Expert
    BB=d.dBB;
    dBB=BB(Yi,:);
    Resu{153,2}=1;
    Resu{1}='Cte';
    ED=zeros(length(Y),136);
    for i=1:16
        Resu{i+1,1}=i;
        for h=i:16
            ED(:,j)=dBB(:,i).*dBB(:,h);
            Resu{17+j,1}=[i h];
            j=j+1;
        end
    end
end

```

```

A=[ones(length(Y),1) dBB ED];
%A2=inv(A'*A);
A2=pinv(A'*A);
Y1=Y;
B=(A2)*(A'*Y1);
Yp = modcuadint(dBB,B);
[m, P]=size(A);
SP=(sum(Y1)^2)/m;
SS1=(B')*(A'*Y1);
SSM=SS1-SP; GLM=P-1;
SST=Y1'*Y1-SP; GLT=m-1;
SSE=SST-SSM; GLE=GLT-GLM;
r2=1-SSE/SST;
disp(r2)
AA = [];bb = [];Aeq = [];beq = [];
[~,I] = min(Y);%Mejor experimento como valor inicial
x0=dBB(I,:);
[~,In] = sort(Y);
II=Yi(In,:);
x00 = BB(II(1:20),:);
x0m = mean(x00);
%Los limites alrededor para hacer la búsqueda más rápida
lb=x0m-0.5;ub=x0m+0.5;
for i=1:16%Cambia los limites
%por si los valores promedio se encuentran fuera de los limites (MV)
if x0(i)>ub(i)
ub(i)=x0(i)+0.1;
end
if x0(i)<lb(i)
lb(i)=x0(i);
end
end
end

```

```

for i=1:16%Cambia los limites si se encuentran fuera de -1 y 1
    if lb(i)<-1
        lb(i)=-1;
    end
    if ub(i)>1
        ub(i)=1;
    end
end
Res.lb=lb;Res.ub=ub;Res.x0=x0;Res.x00=x00;Res.x0m=x0m;Res.Resu=Resu;
options = struct('MaxFunctionEvaluations', 10000); %Número de iteraciones
%Optimización
xopi = fmincon(@(X) modcuadint(X,B),x0(i,:),AA,bb,Aeq,beq,lb,ub,[],options);
yopi = modcuadint(xopi(i,:),B);%Valor de Y
xop=xopi;yop=yopi;
disp(xop)
disp(yop)
end

```

A.4.2 Optimización ponderaciones

Al_weights_fo_p.m

```

%Análisis Pesos para mejor ponderación
%Función Objetivo completa
clear; clc
addpath('Funciones')%Carpeta con funciones
load('W_YP_M4_L_fo_v.mat')%Resultados
load('W_Y_M4_L_fo_v_c.mat')
load('DM_DE.mat')%Diseño de mezclas Design Expert
%Análisis ponderaciones -----
[~,A] = modcubicmez(DM(:,1:2),ones(10,1));
A2=inv(A'*A);
%A2=pinv(A'*A);

```

```

Ylp=log(Ylp);%Transformación
B=(A2)*(A'*Ylp);
Ylpp = exp(modcubicmez(DM(:,1:2),B));%————
figure(1)
clf
%plot(Ylp,Ylpp,'*b')
plot(exp(Ylp),Ylpp,'*b')
hold on
plot([-0.5 3],[-0.5 3],'-r')
[m, P]=size(A);
SP=(sum(Ylp)^2)/m;SS1=(B'*(A'*Ylp);
SSM=SS1-SP;SST=Ylp'*Ylp-SP;SSE=SST-SSM;r2=1-SSE/SST;
disp('Optimización Ponderaciones')
disp(r2)
%Grafica —————
m=100;
x1=0:1/m:1;x2=1:-1/m:0;
k=1;X1g=zeros(m+1,m+1);X2g=X1g;
for i=1:length(x1)
    j=1;
    for h=i:length(x1)
        X1g(j,k)=x1(:,i); X2g(j,k)=x2(:,h);
        j=j+1;
    end
    k=k+1;
end
X3g=1-X1g-X2g;Zg=X1g;
for i=1:length(X1g)
    for j=1:length(X1g)
        %Zg(i,j) = modcubicmez([X1g(i,j),X2g(i,j)],B);
        Zg(i,j) = exp(modcubicmez([X1g(i,j),X2g(i,j)],B));%—
    end
end

```

```

end
figure (2)
clf
%contourf(X1g,X2g,Zg)
mix_plotc(X1g,X2g,Zg)
xlabel('te')
ylabel('OS')
hold on
%plot(DM(:,1),DM(:,2),'*b')
mix_plot(DM(:,1),DM(:,2),'xb',8)
hold off
%Optimización ponderaciones -----
%Restricciones
AA = [1,1];bb = 1; % x1 + x2 <= 1
Aeq = [];beq = [];
%Limites
lb=zeros(1,2);ub=ones(1,2);
%Valor inicial
[Mz,Iz] = min(Zg);[~,Iz1] = min(Mz);
x0=[ X1g(Iz(Iz1),Iz1) X2g(Iz(Iz1),Iz1)];
x02=ones(1,2)*1/3;
xopp = fmincon(@(X) modcubicmez(X,B),x0,AA,bb,Aeq,beq,lb,ub);
xopp(3)=1-xopp(1)-xopp(2);
yopp = modcubicmez(xopp(:,1:2),B);
x0(3)=1-x0(1)-x0(2);
y0 = modcubicmez(x0(:,1:2),B);
xopp2 = fmincon(@(X) modcubicmez(X,B),x02,AA,bb,Aeq,beq,lb,ub);
xopp2(3)=1-xopp2(1)-xopp2(2);
yopp2 = modcubicmez(xopp2(:,1:2),B);
if yopp2<yopp
    xoppa=xopp2;yoppa=yopp2;
else

```

```

xoppa=xopp;yoppa=yopp;
end
if y0<yoppa
  xoppa=x0;yoppa=y0;
end
disp(xoppa)
disp(yoppa)
hold on
mix_plot(xoppa(:,1),xoppa(:,2),'*r',10)
axes1 = axes('Parent',gcf);
hold(axes1,'on');
set(axes1,'PlotBoxAspectRatio',[1.1547 1 1]);
axis off
hold off
%Mejores Pesos -----
load('DM_DE.mat')%Diseño de mezclas Design Expert
d=load('BB_16_DE.mat');BB=d.dBB;% Design Expert
Y=RY*xoppa';
%Y=log(Y);%Transformación
[~,I] = min(Y);
Yim=Yi(I);
xme=BB(Yi(I),:);
disp('Optimización Mejores Pesos')
[Res,xop,yop,~,r2b,~,~] = fpesos_fode_C(Y,Yi);
Base=zeros(1,16);Dif=1;
xmer=[exp((Dif*xme(:,1:8)+Base(:,1:8)).*log(5)) ...
  exp((Dif*xme(:,9:16)+Base(:,9:16)).*log(5))./10];
xopr=[exp((Dif*xop(:,1:8)+Base(:,1:8)).*log(5)) ...
  exp((Dif*xop(:,9:16)+Base(:,9:16)).*log(5))./10];

```