

Trabajo Fin de Grado
Grado de Ingeniería en Tecnologías Industriales

Diseño del Sistema de Control de un Evaporador

Autor: Diego Sokolowski Barrón

Tutor: Prof. Dr. Francisco Javier Gutiérrez Ortiz

Departamento de Ingeniería Química y Ambiental
Escuela Técnica Superior de Ingeniería
Universidad de Sevilla

Sevilla, 2016



Trabajo Fin de Grado
Grado de Ingeniería en Tecnologías Industriales

Diseño del Sistema de Control de un Evaporador

Autor:

Diego Sokolowski Barrón

Tutor:

Dr. Francisco Javier Gutiérrez Ortiz

Profesor Titular

Dep. de Ingeniería Química y Ambiental

Escuela Técnica Superior de Ingeniería

Universidad de Sevilla

Sevilla, 2016

Trabajo Fin de Grado: Diseño del Sistema de Control de un Evaporador

Autor: Diego Sokolowski Barrón

Tutor: Prof. Dr. Francisco Javier Gutiérrez Ortiz

El tribunal nombrado para juzgar el Proyecto arriba indicado, compuesto por los siguientes miembros:

Presidente:

Vocales:

Secretario:

Acuerdan otorgarle la calificación de:

El Secretario del Tribunal

Fecha:

Agradecimientos

A mi familia y amigos por su constante apoyo durante toda la carrera en general y durante este proyecto en particular. A mi tutor Francisco Javier Gutiérrez Ortiz por su dedicación y tiempo prestado.

Diego Sokolowski Barrón

Sevilla, 2016

Resumen

En este proyecto se desarrolla un modelo matemático de un evaporador de tubos verticales con recirculación forzada, y se diseña un sistema de control. En primer lugar, se definen las ecuaciones y variables que constituyen el proceso, especificando las salidas y entradas del sistema. Luego, se identifican los emparejamientos salida-entrada más idóneos para un control SISO, mediante la RGA. Posteriormente, se implementa el modelo no lineal en Simulink-MATLAB®, para poder observar las respuestas del sistema frente a cambios en sus variables de entrada. Gracias a la utilidad Ident Toolbox de MATLAB® se identifican modelos simplificados lineales, a partir de los cuales se aborda el diseño de los controladores por realimentación más adecuados. Finalmente, se implementan controladores avanzados anticipativos y en cascada, con el objetivo de reducir los efectos indeseables provocados por perturbaciones. Paralelamente a todo este proceso, se ha desarrollado una interfaz gráfica GUI en Simulink, que permite al usuario interactuar con el sistema controlado para observar los resultados directamente a tiempo real.

Abstract

The aim of this Project is to develop the dynamic model of a forced circulation vertical tube evaporator and to design its control system. First of all, the equations and variables which establish the process are calculated and defined, specifying the outputs and inputs of the system. The best input-output pairings for a controlled SISO system are selected due to the RGA matrix. Subsequently, the nonlinear model is implemented with Simulink-MATLAB[®] to analyze the results that the system generates in the presence of input variables changes. Thanks to Ident Toolbox-MATLAB[®], linear simplified models of the system are identified, from which the best-fitting feedback control solutions are tackled. Finally, feed-forward and cascade controllers are introduced to minimize disturbance effects. Concurrently, a visual GUI Simulink interface is developed in such a way that the results of the controlled system can be checked under real-time conditions.

Índice

Agradecimientos	vii
Resumen	ix
Abstract	xi
Índice	xiii
1 Introducción	1
1.1 <i>Objetivo del proyecto</i>	1
1.2 <i>El sistema evaporador</i>	1
2 Modelo Dinámico	5
2.1 <i>Balance de masa y energía. Evaporador</i>	5
2.1.1 <i>Modelo del intercambiador propuesto por Newell & Lee</i>	5
2.1.2 <i>Modelo final del evaporador</i>	5
2.2 <i>Balance de masa y energía. Separador.</i>	9
2.3 <i>Balance de masa y energía. Condensador.</i>	10
2.4 <i>Dinámica en el interior de la cámara del evaporador y válvulas</i>	11
2.5 <i>Consideraciones</i>	12
2.6 <i>Resumen ecuaciones modelo y variables manipulables y de control</i>	14
3 Linealización	15
4 Modelo en Simulink-Matlab	19
5 Diseño del sistema de control	27
5.1 <i>Identificación $G(s)$</i>	27
5.2 <i>Identificación $G_d(s)$</i>	33
5.3 <i>Elección emparejamientos SISO</i>	39
5.4 <i>Control SISO equivalente con controles por realimentación</i>	41
5.4.1 <i>Adimensionalización</i>	42
5.4.2 <i>Control realimentado emparejamiento $X_2'-X_{F100}$</i>	44
5.4.3 <i>Control realimentado para el emparejamiento P2-F200</i>	50
5.4.4 <i>Control realimentado emparejamiento L2-F2'</i>	53
5.5 <i>Control avanzado para rechazo de perturbaciones: acción anticipativa y control en cascada</i>	55
5.5.1 <i>Acción anticipativa para F1 en el emparejamiento $X_2'-X_{F100}$</i>	56
5.5.2 <i>Control prealimentado para T200 en el emparejamiento P2-F200</i>	57
5.5.3 <i>Acción anticipativa para F1 en el emparejamiento L2-F2'</i>	58
5.5.4 <i>Control en cascada para P0 en el emparejamiento $X_2'-x_{F100}$</i>	61
5.6 <i>Control SISO completo con todos los controladores</i>	64
5.6.1 <i>Control SISO completo frente a perturbaciones</i>	64
5.6.2 <i>Control SISO completo frente a seguimiento de referencias</i>	65
5.7 <i>Comparación con otros autores</i>	67
6 Interfaz usuario en tiempo real	69

7 Conclusiones y Líneas futuras	71
7.1 Conclusiones	71
7.2 Líneas futuras	71
Apéndice A. Códigos Matlab	73
A.1 Parámetros y puntos nominales de funcionamiento	73
A.2 Cálculo de puntos nominales de funcionamiento, (evaporador N compartimentos)	73
A.3 Representación gráfica de comparación de simulaciones	75
A.4 GUI de sistema de control Simulink	77
Apéndice B. Diagramas Simulink	93
B.1 MODELO NO LINEAL	93
B.1.1 Separador	93
B.1.2 Condensador	94
B.1.3 Evaporador	95
B.2 MODELO NO LINEAL. CONTROL SISO COMPLETO	98
Índice de ilustraciones	101
Bibliografía	105

1 INTRODUCCIÓN

1.1 Objetivo del proyecto

El objetivo de este proyecto es diseñar el sistema de control para un evaporador de tubos largos verticales de recirculación forzada a partir del desarrollo previo de un modelo matemático del sistema. Adicionalmente, se trabajará en un entorno gráfico de simulación que permita observar las respuestas del sistema en tiempo real al variar las variables de perturbación o los puntos de consigna del sistema controlado. Dicho entorno, que se implementará en MATLAB®-Simulink, permite el estudio y la visualización de los resultados del proceso de una manera más rápida e intuitiva.

Dado que los evaporadores industriales disponen de múltiples entradas manipulables, perturbaciones, y salidas, la complejidad de la solución de control radica en la naturaleza multivariable de la planta a estudiar.

Por tanto, el desarrollo de este documento se centra en tres bloques principales:

- Recopilación de información y obtención del modelo dinámico.
- Desarrollo de la solución de control del sistema multivariable.
- Diseño del entorno gráfico.

En comparación con otros estudios previos, este trabajo profundiza en mayor detalle en el modelo dinámico del sistema y ofrece una interfaz gráfica que dota al conjunto de un carácter didáctico.

1.2 El sistema evaporador

Un evaporador es un intercambiador de calor en el que se produce una transferencia de energía térmica entre un fluido caliente en estado gaseoso (típicamente vapor de agua) y un fluido líquido a menor temperatura. El fluido gaseoso se enfría y se condensa, y el fluido líquido se calienta. Su nombre se debe al cambio de estado producido en el líquido, pues parte de éste se evapora al recibir el calor del otro fluido.

Los evaporadores se encuentran en todo tipo de industrias. Uno de sus usos más frecuentes es como sistema de refrigeración, en equipos como cámaras frigoríficas, sistemas de aire acondicionado y neveras. También son comúnmente utilizados en procesos químicos de precipitación, cristalización y extracción de líquidos, entre otros. Dependiendo de su aplicación y carga térmica su diseño es diferente, variando su tamaño, capacidad y estructura.

Existen diferentes configuraciones, entre las que destacan: evaporador de película descendente, evaporador de película ascendente, evaporador de tubos verticales horizontales o verticales con recirculación forzada, evaporador de placas, etc.

Debido a su especial diseño, los evaporadores con tubos verticales y recirculación forzada son adecuados para concentrar disoluciones con tendencia a la cristalización, incrustación, o que tienen gran viscosidad. Se usan mayoritariamente en la industria alimenticia e industria farmacéutica y química, así como en procesos medioambientales como el tratamiento de aguas. Es el tipo de evaporador elegido en este proyecto. Dicho sistema evaporador consta de:

Equipos

- Intercambiador de calor de tubos verticales (evaporador).
- Separador flash líquido-gas.
- Condensador.
- Tubería de recirculación entre el separador líquido-gas y el intercambiador de calor, así como una bomba de impulsión para la recirculación.

Corrientes

- Una entrada de alimentación de disolución en el circuito de recirculación.
- Una entrada de vapor en la cámara exterior a los tubos del intercambiador de calor.
- Una corriente de entrada de agua refrigerada en el condensador.
- Una salida de concentrado del circuito de recirculación.
- Una salida de condensado en el intercambiador de calor.
- Una salida de condensado procedente del flujo del vapor del separador que pasa a través del condensador .

El proceso de funcionamiento es el siguiente:

- 1) La solución líquida que se quiere concentrar es alimentada por el tubo de entrada inferior al evaporador y fluye de abajo hacia arriba a través de los tubos impulsada por la bomba.
- 2) En los tubos del evaporador el líquido se calienta y parte del agua se evapora debido al vapor presente en la cámara que rodea a los tubos. De este modo, se genera en el interior de los tubos una mezcla líquido-vapor saturada. En concreto, al aumentar el caudal de alimentación de vapor (ya sea por la variación de la presión de alimentación proveniente de una caldera o por la variación en la apertura de la válvula), aumentará la presión en el interior de la cámara del intercambiador, lo que incrementará también su temperatura y, por tanto, el flujo de calor hacia el líquido contenido en los tubos.
- 3) A la salida de los tubos del evaporador, el vapor de agua se separa del líquido concentrado en un separador líquido-vapor.
- 4) El vapor de agua producido es conducido a través de un condensador.
- 5) El líquido concentrado del separador es recirculado de nuevo hacia el evaporador mediante una bomba. Aguas arriba de la bomba se encuentra una salida de concentrado, y posteriormente de nuevo la entrada del líquido a concentrar.

El esquema del sistema evaporador con sus variables se muestra en la **Figura 1.1** y **Tabla 1.1** [6].

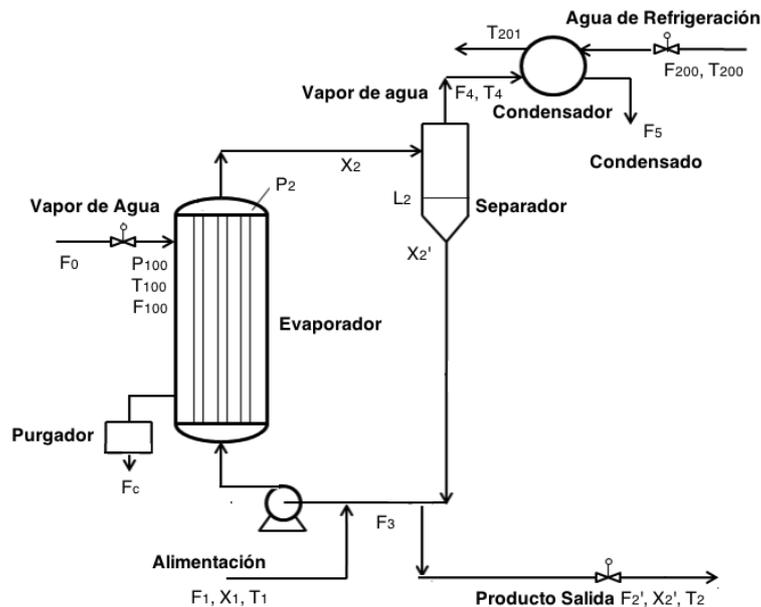


Figura 1.1 Esquema de funcionamiento Evaporado

Variables	Descripción	Punto nominal	Máximo	Mínimo	Unidades
F ₁	Caudal de alimentación	10.0			kg/min
F ₂	Caudal de líquido salida evaporador	52.0	-	-	kg/min
F' ₂	Caudal de producto	2.0	5	0	kg/min
F ₃	Caudal de recirculación	50.0	-	-	kg/min
F ₄	Caudal de vapor	8.0	-	-	kg/min
F ₅	Caudal de condensado	8.0	-	-	kg/min
X ₁	Concentración de alimentación	5.0			% $\left(\frac{\text{kg s.}}{\text{kg d.}}\right)^*$
X ₂	Concentración en el separador	25.0	40	0	% $\left(\frac{\text{kg s.}}{\text{kg d.}}\right)$
X' ₂	Concentración de producto de salida	25.0	40	0	% $\left(\frac{\text{kg s.}}{\text{kg d.}}\right)$
T ₁	Temperatura de alimentación	40.0			°C
T ₂	Temperatura de producto de salida	84.6	-	-	°C
T ₄	Temperatura del vapor de agua	80.6	-	-	°C
L ₂	Nivel del separador	1.0	2	0	m
P ₀	Presión de vapor de la caldera	500			kPa
P ₂	Presión en el evaporador	50.5	100	0	kPa
F ₁₀₀	Caudal de vapor de alimentación	9.3	-	-	kg/min
X _{F100}	Apertura de válvula de caudal de vapor	0.5	1	0	-
F _c	Caudal de condensado de la carcasa	9.3	-	-	kg/min
T ₁₀₀	Temperatura de vapor de alimentación	119.9	-	-	°C
P ₁₀₀	Presión de vapor de alimentación	194.7	-	-	kPa
Q ₁₀₀	Flujo de calor en el intercambiador	339.0	-	-	kW
F ₂₀₀	Caudal de refrigeración	208.0	400	0	kg/min
T ₂₀₀	Temperatura de entrada líquido de refrigeración	25.0			°C
Q ₂₀₀	Flujo de calor en el condensador	307.9	-	-	kW
T ₂₀₁	Temperatura de salida líquido de refrigeración	46.1	-	-	°C

Tabla 1.1 Variables del sistema

* kg s. (kg de soluto), kg d. (kg de disolución)

2 MODELO DINÁMICO

Para obtener el modelo dinámico se usan principalmente las ecuaciones de conservación de energía y de masa, así como las ecuaciones de transferencia de calor y las ecuaciones de estado termodinámicas. Se toma como referencia base el modelo de Newell and Lee, al que se añaden diversas modificaciones en base al modelado de plantas químicas [7].

2.1 Balance de masa y energía. Evaporador

2.1.1 Modelo del intercambiador propuesto por Newell & Lee

El modelo propuesto por Newell and Lee [6] estipula las siguientes consideraciones:

La temperatura del líquido es:

$$T_2 = 0.5616P_2 + 0.3126X_2 + 48.43 \quad (2-1)$$

La fórmula anterior proviene de la linealización de la curva de equilibrio temperatura-presión de líquido saturado del agua alrededor del punto nominal de funcionamiento, e incluye un término que aproxima el punto de ebullición debido a la presencia del soluto (aumento ebulloscópico). Dicho término se obtiene linealizando las gráficas de Duhring para un líquido en el que se encuentra disuelto NaCl, que es la sal usada en este modelo.

La dinámica del balance de energía se considera muy rápida, por lo que:

$$F_4 = (Q_{100} - F_1 \cdot C_p(T_2 - T_1))/\lambda \quad (2-2)$$

C_p es la capacidad calorífica de la disolución y se asume constante e igual a 0.07 kW/K(kg/min) (4.2 KJ/kgK para agua saturada a 80 °C) y λ es el calor latente de vaporización de la disolución y se asume constante igual a 38.5 kW/(kg/min) (2304.11 KJ/kg a 50.5 KPa).

2.1.2 Modelo final del evaporador

Las ecuaciones de Newell y Lee conducen a un modelo demasiado simplificado que considera una circunstancia inadmisibles en este proyecto: La temperatura del líquido en el evaporador es la temperatura final T_2 en el mismo instante en el que entra en el inicio de los tubos. Es decir, el modelo de Newell y Lee considera que el intercambiador de calor de tubos largos actúa como un tanque homogéneo simplificado de mezcla perfecta.

Para solucionar esto, se propone un modelado diferente, en el que el evaporador se divide en "N" compartimentos secuenciales, continuos e iguales en tamaño. Cada compartimento actúa como un tanque homogéneo, teniendo por tanto cada uno de ellos corrientes de entrada y de salida. Puede subdividirse cada corriente en una fracción líquida y otra vapor, pero para mayor claridad, se tratará como si hubiese dos corrientes (líquida y vapor) tanto en la entrada como en la salida de los compartimentos. La temperatura en el interior de cada compartimento es, por tanto, igual a la temperatura de las corrientes de salida de ese compartimento. La siguiente Figura aclara la idea expuesta:

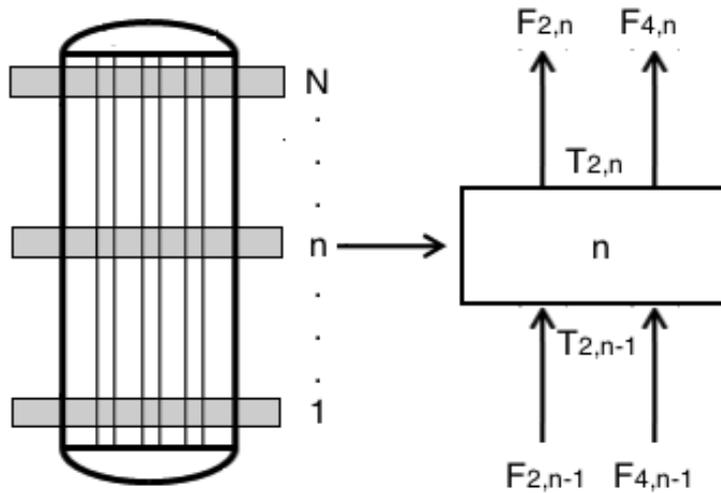


Figura 2.1 Esquema subdivisión el evaporador

La entrada del evaporador, F_2 , es la suma de la corriente de alimentación F_1 y la corriente de recirculación F_3 .

Planteando los balances de energía y de masa de cada subdivisión se obtiene el siguiente sistema de ecuaciones, para $n=\{1, \dots, N\}$:

$$\left. \begin{aligned} \frac{M}{N} C_{pL} \frac{dT_{2,n}}{dt} &= (F_{2,n-1}T_{2,n-1} - F_{2,n}T_{2,n})C_{pL} + (F_{4,n-1}T_{2,n-1} - F_{4,n}T_{2,n})C_{pV} \\ &\quad - (F_{4,n} - F_{4,n-1})H_v(T_{2,n}) + h_{int}A_{int}(T_{w,n} - T_{2,n})/N \end{aligned} \right\} \quad (2-3)$$

$$\frac{M}{N} \frac{dX_{2,n}}{dt} = F_{2,n-1}X_{2,n-1} - F_{2,n}X_{2,n} \quad (2-4)$$

$$T_{2,n} = 0.5616P_2 + 0.3136X_{2,n} + 48.43 \quad (2-5)$$

$$F_{2,n} = (F_{2,n-1} + F_{4,n-1}) - F_{4,n} \quad (2-6)$$

$$M_w C_{p_w} \frac{dT_{w,n}}{dt} = (Q_{100} - h_{int}A_{int}(T_{w,n} - T_{2,n})) \quad (2-7)$$

Cuanto mayor sea el valor de “N”, mejor será el modelado del evaporador, pero mayor será la complejidad al aumentar el número de incógnitas y ecuaciones. En este proyecto se toma el valor $N=3$, pues se ha calculado previamente hasta $N=5$ y a partir de $N=3$ la mejoría es cada vez más pequeña, aumentando excesivamente la complejidad de los diagramas en Matlab®-Simulink. En cuanto a las constantes y variables introducidas en el anterior sistema de ecuaciones, a continuación se recogen sus descripciones y valores conocidos:

- **M**: Masa de líquido en el evaporador. Se asume constante e igual a 20 kg.
- **C_{pL}**: Capacidad calorífica de la disolución líquida. Se asume constante e igual al C_p del agua líquida (0.08 kW/K(kg/min)), según las tablas de líquido-gas saturado del agua [4].
- **C_{pV}**: Capacidad calorífica del vapor de agua. Se asume constante e igual a 0.025 kW/K(kg/min),

según las tablas de líquido-gas saturado del agua [4].

- $T_{2,n-1}$ y $T_{2,n}$: Temperatura de las corrientes de entrada y de salida en cada compartimento, respectivamente. Son variables del sistema de ecuaciones y tienen unidades de K.
- $F_{2,n-1}$ y $F_{2,n}$: Caudal másico (kg/min) del fluido líquido de entrada y de salida en cada compartimento, respectivamente. Son variables.
- $F_{4,n-1}$ y $F_{4,n}$: Caudal másico (kg/min) del fluido gaseoso de entrada y de salida en cada compartimento, respectivamente. Son variables.
- N : número total de divisiones realizadas en el evaporador (3 en este caso), $n = \{1, \dots, N\}$.
- $H_v(T_{2,n})$: Entalpía del vapor evaluada a la temperatura $T_{2,n}$. Unidades: kW·min/kg. Teniendo en cuenta que $H_v(40\text{ °C}) = H_v(n = 1) = 42.892\text{ kWmin/kg}$ y que $H_v(84.6\text{ °C}) = H_v(n = 3) = 44.178\text{ kWmin/kg}$ [4], se linealiza entre los dos puntos para obtener una expresión lineal de la entalpía en función de la porción del intercambiador que se esté analizando (de un total de N porciones):

$$H_v(n) - H_v(0) = H'_v(n) \cdot (n - n(0)) \rightarrow$$

$$H_v(n) = 42.892 + \frac{44.178 - 42.892}{3 - 1}(n - 1) = 42.892 + 0.643(n - 1) \quad (2-8)$$

- $X_{2,n-1}$ y $X_{2,n}$: Concentración de la disolución a la entrada y a la salida de la porción, respectivamente. Unidades medidas en % (gramos de soluto entre gramos de disolución).
- P_2 : Presión de operación en el interior del evaporador (en kPa).

La ecuación (2-7) proviene del estudio de la transferencia de calor entre el vapor de entrada en la cámara y el líquido que circula en el interior de los tubos, según se detalla en el apartado siguiente:

2.1.2.1 Transferencia de calor en los tubos interiores del intercambiador de calor

El vapor que entra en el intercambiador proporciona un suministro de calor que calienta la pared exterior de los tubos interiores. Este calentamiento produce un cambio de temperatura en la pared exterior e interior de los tubos, que produce un aumento de la temperatura del fluido que circula por el interior de los tubos (ver **Figura 2.2**) [4].

El calor del vapor de entrada (Q_{100}) se invierte, por tanto, en calentar la pared de los tubos ($M_w C_{p_w} \frac{dT_{w,n}}{dt}$) y el fluido interior ($Q = UA_i(T_{100} - T_2) = \frac{1}{(KL)^{-1} + (h_{int}A_{int})^{-1}}(T_{w1,n} - T_{2,n}) = KL(T_{w1} - T_{w2}) = h_{int}A_{int}(T_{w2,n} - T_{2,n})$), como se ve reflejado en la ecuación (2-7).

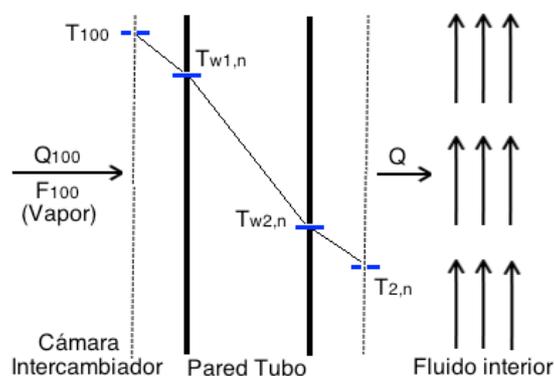


Figura 2.2 Sección longitudinal interior de una de las mitades simétricas del intercambiador de calor

- Q_{100} : Ver **Tabla 1.1**.

- $T_{w,n}$: Temperatura de la pared interior de los tubos del intercambiador en la porción “n”. Variable medida en K. Su valor nominal inicial se estima $T_{w,n}^0 = 88$ °C. En **Figura 2.2** se usa el subíndice $T_{w2,n}$ para esta variable para distinguir adecuadamente la pared exterior de la interior. Desde este punto en adelante se utiliza $T_{w,n}$ para dicha variable a fin de simplificar los índices.
- $h_{int}A_{int}$: Producto de la constante de convección entre la pared interior de los tubos y la corriente interior a los mismos por el área interior de los tubos. Constante e igual a 100.11 kW/K. Dicho valor proviene de la ecuación evaluada en torno al punto nominal de (2-7):

$$p. n. \rightarrow Q_{100}^0 = h_{int}A_{int}(T_{w,n}^0 - T_{2,n}^0) \rightarrow h_{int}A_{int} = \frac{F_{100}^0 \cdot \lambda_s}{T_{w,n}^0 - T_{2,n}^0} = \frac{9.3 \cdot 36.6}{88 - 84.6} = 100.11 \text{ kW/K} \quad (2-9)$$

- M_w : Masa de los tubos. Esta constante es un valor intrínseco del intercambiador y depende lógicamente del material y el tamaño del mismo. Como estimación se toma el valor $M_w=200$ kg.
- Cp_w : Capacidad calorífica de los tubos del intercambiador. Depende del material de los mismos. Se estima constante e igual a 0.02 kW/K(kg/min) [4].

De esta manera, quedan definidas todos los parámetros y variables del sistema de ecuaciones. Para analizar los grados de libertad más adelante, hay que tener en cuenta que el primer compartimento y el último en los que se divide el intercambiador poseen las particularidades indicadas en la **Tabla 2.1**.

n=1	Aclaración	n=3	Aclaración
$F_{2,0} = F_1 + F_3'$	La entrada de la primera porción es el líquido de alimentación	$F_{2,3} = F_2$	La salida de la última porción es igual a la salida total del intercambiador
$F_{4,0} = 0$	No hay vapor en la entrada del intercambiador	$F_{4,3} = F_4$	La salida de la última porción es igual a la salida total del intercambiador
$T_{2,0} = \frac{C_p(F_1 T_1 + F_3' T_{2,N}^0)}{(F_1 + F_3') C_p}$	Temperatura de entrada del líquido de alimentación	$T_{2,3} = T_2$	Temperatura del líquido de salida
$X_{2,0} = \frac{X_1 F_1 + X_{2,N}^0 F_3'}{F_1 + F_3'}$	Concentración del líquido de alimentación	$X_{2,3} = X_2$	Concentración del líquido de salida

Tabla 2.1 Particularidades primera y última porción

El calor transferido al líquido de proceso también se puede expresar en función de la temperatura del vapor en el interior de la cámara:

$$Q_{100,n} = UA_1(T_{100} - T_{2,n}) \rightarrow Q_{100} = \sum_n Q_{100,n} \quad (2-10)$$

donde UA_1 es el coeficiente global de transferencia de calor entre el vapor de la cámara y el líquido de recirculación de los tubos. Evaluando la ecuación anterior en el punto nominal de funcionamiento se puede obtener un valor aproximado de UA_1 :

$$Q_{100}^0 = UA_1(T_{100}^0 - T_2^0) \rightarrow UA_1 = \frac{339}{119.9 - 84.6} = 9.6 \text{ kW/K} \quad (2-11)$$

La presión de vapor P_{100} es una variable que determina la temperatura del vapor en condiciones de saturación. Se puede obtener una ecuación que relaciona la temperatura del vapor con su presión aproximando la curva de equilibrio temperatura-presión de vapor saturado por su linealización en el punto nominal de funcionamiento:

$$T_{100} = 0.1538P_{100} + 90.0 \quad (2-12)$$

2.2 Balance de masa y energía. Separador.

La **Figura 2.3** muestra las variables que conforman la dinámica del separador, la bomba de impulsión y la salida de producto:

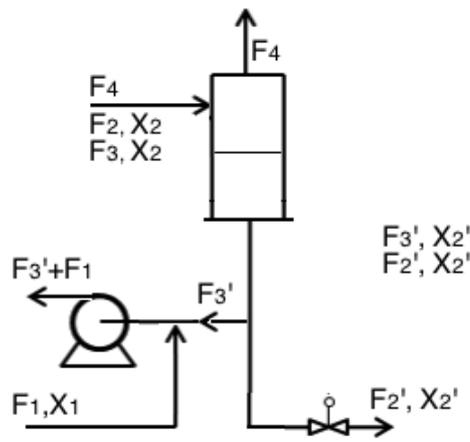


Figura 2.3 Variables separador, bomba y producto

El balance de masa realizado sobre el total del líquido de proceso del sistema se puede formular como sigue, efectuado en el separador:

$$\rho A \frac{dL_2}{dt} = (F_2 + F_3) - (F'_2 + F'_3) \quad (2-13)$$

El producto ρA (ρ es la densidad del líquido y A el área transversal del separador) se asume como constante de valor 20 kg/m.

Como se asume que la variación de masa con respecto al tiempo en el evaporador es nula, se obtiene:

$$\frac{dM}{dt} = (F'_3 + F_1) - (F_2 + F_3 + F_4) = 0 \rightarrow F_2 + F_3 + F_4 = F'_3 + F_1 \quad (2-14)$$

Sustituyendo la ecuación anterior en (2-13) se llega al siguiente resultado:

$$\rho A \frac{dL_2}{dt} = F_1 - F_4 - F'_2 \quad (2-15)$$

Desarrollando el balance de masa de la sal disuelta en el separador se obtiene:

$$\begin{aligned} \rho A \frac{d(L_2 X'_2)}{dt} &= (F_2 + F_3)X_2 - (F'_2 + F'_3)X'_2 \rightarrow \\ \rho A L_2 \frac{dX'_2}{dt} + \rho A X'_2 \frac{dL_2}{dt} &= (F_2 + F_3)X_2 - (F'_2 + F'_3)X'_2 \end{aligned} \quad (2-16)$$

Finalmente, al sustituir resulta:

$$\rho A L_2 \frac{dX'_2}{dt} = (F_1 + F'_3 - F_4) \cdot (X_2 - X'_2) \quad (2-17)$$

nota: se asume que $F'_3 = F_3 = F_3^0 = 50 \text{ kg/min}$

El balance de masa aplicado al vapor de proceso del evaporador expresa la masa total del vapor de agua en términos de presión existente en el sistema:

$$C \frac{dP_2}{dt} = F_4 - F_5 \quad (2-18)$$

C es una constante que convierte la masa de vapor en una presión equivalente y se asume fija en 4 kg/kPa. Dicha constante depende de la geometría del separador. La ecuación (2-18) expresa una variación de presión en el sistema separador dependiente de la diferencia del caudal de vapor y del caudal de condensado.

La temperatura del vapor se aproxima por (2-19):

$$T_4 = 0.507P_2 + 55.0 \quad (2-19)$$

Que es una linealización de la curva de equilibrio temperatura-presión de líquido saturado del agua alrededor del punto nominal de funcionamiento. Se asume que el líquido de proceso siempre se encuentra en su temperatura de ebullición y está perfectamente mezclado (debido a su alta velocidad y presión de circulación).

2.3 Balance de masa y energía. Condensador.

El caudal de agua refrigerante F_{200} es una variable manipulable y la temperatura T_{200} es una perturbación. El balance de energía aplicado al agua refrigerante es el siguiente:

$$Q_{200} = F_{200} C_p (T_{201} - T_{200}) \quad (2-20)$$

donde C_p es la capacidad calorífica del agua refrigerante y se asume constante e igual a 0.07 kW/(kg/min) [4] (debido a que la diferencia de temperatura es pequeña como para provocar un cambio significativo en C_p). Aplicando al mismo sistema la ecuación de transferencia de calor se concluye:

$$Q_{200} = UA_2 (T_4 - 0.5(T_{200} + T_{201})) \quad (2-21)$$

Donde UA_2 es el coeficiente global de transferencia de calor, y se asume constante e igual a 6.84 kW/K. Las ecuaciones (2-20) y (2-21) se pueden combinar eliminando de esta manera el término T_{201} :

$$Q_{200} = \frac{UA_2 (T_4 - T_{200})}{1 + \frac{UA_2}{2C_p F_{200}}} \quad (2-22)$$

$$T_{201} = T_{200} + \frac{Q_{200}}{F_{200} C_p} \quad (2-23)$$

El caudal másico de condensado es:

$$F_5 = Q_{200} / \lambda \quad (2-24)$$

donde λ es calor latente de vaporización del agua, que se asume constante e igual a 38.5 kW/ K(kg/min) [4].

La dinámica dentro del condensador se asume muy rápida con respecto a la dinámica total del sistema a controlar.

2.4 Dinámica en el interior de la cámara del evaporador y válvulas

Aplicando el balance de masa en la carcasa del evaporador, la variación de masa con respecto al tiempo en su interior depende del caudal másico de vapor que entra y del caudal másico de condensado que sale:

$$V_s \frac{d\rho_v}{dt} = F_v - F_c \quad (2-25)$$

donde V_s y ρ_v son el volumen de la carcasa y la densidad del vapor en la carcasa, respectivamente. $F_v = F_{100}$ y F_c es el caudal másico de líquido condensado que sale de la carcasa.

Por otro lado, la variación de la energía interna del agua en el interior de la carcasa depende de la energía que entra, de la que sale, y de la transmitida hacia el interior de los tubos:

$$V_s \frac{d(\rho_v u_v)}{dt} = F_{100} H_v(T_v) - F_c H_L(T_v) - UA(T_v - T_2), \quad U \cong C_v]_{v=cte} \quad (2-26)$$

De nuevo, los subíndices “v” se refieren al vapor y por concordancia con los términos de anteriores definiciones, se sustituirá finalmente por el subíndice “100”. El término izquierdo de la ecuación anterior se puede desarrollar como sigue:

$$V_s \rho_v c_v \frac{dT_v}{dt} + V_s T_v c_v \frac{d\rho_v}{dt} \quad (2-27)$$

$$V_s T_v c_v \frac{d\rho_v}{dt} = F_{100} c_v T_v - F_c c_v T_v \quad (2-28)$$

Sustituyendo en la ecuación (2-26) se llega a la siguiente expresión:

$$M_{100} C_{Vv} \frac{dT_{100}}{dt} = F_{100} (H_v(T_{100}) - C_p T_{100}) - F_c (H_L(T_{100}) - C_{pv} T_{100}) - UA(T_{100} - T_2) \quad (2-29)$$

Entre las variables anteriores, se hacen las siguientes aclaraciones:

- M_{100} : Es la masa de vapor en el interior de la carcasa (kg). Se calcula como resultado del producto de la densidad de vapor por el volumen de la carcasa. Considerando un volumen de carcasa = 10 m³, la masa de vapor es igual a:

$$M_{100} = \rho_{100} \cdot V_s = \frac{p_{100}(\text{atm}) \cdot PM(H_2O)}{0.082 \cdot T_{100}(\text{K})} \cdot 10 = \frac{18p_{100}}{0.082T_{100}} \cdot 10 \quad (2-30)$$

- Tanto C_{pv} , como H_L y H_v varían con la temperatura. Sin embargo, se comprueba que aunque haya variaciones considerables en T_{100} , los resultados de las restas $[H_v(T_{100}) - C_{pv} T_{100}]$ y $[H_L(T_{100}) - C_p T_{100}]$, se pueden aproximar por el valor de las mismas para T_{100} en el punto nominal de funcionamiento.

Así, por ejemplo, consultando las tablas, para un valor de $T_{100} = 150$ °C y para un valor $T_{100} = T_{100}^0$ °C, el cociente $\frac{H_v(150 \text{ °C}) - C_{pv}(150 \text{ °C})(150+273)}{H_v(119.9 \text{ °C}) - C_{pv}(119.9 \text{ °C})(119.9+273)} = 1.017$ [4]. Dado que la variación es pequeña, se aproximan dichas diferencias por su valor para $T_{100} = T_{100}^0$.

Los caudales F_{100} y F_c se pueden regular mediante válvulas. En concreto, se han elegido para el modelado válvulas de control neumático Samson Type 3241-1 [8].

En el caso de caudal de vapor, la válvula regula la entrada de vapor procedente de una caldera y desemboca

dicho caudal en la carcasa del intercambiador. Matemáticamente Samson propone el siguiente modelo:

$$F_{100} = 60 \cdot K_{vs} \cdot xF_{100} \cdot 14.2 \cdot m \cdot \sqrt{P_0 \cdot \rho_0} \quad (2-31)$$

- **m**: Coeficiente de pérdida de presión. Para valores de $\frac{p_{100}}{p_0} < 0.6$, $m=1$. En este caso, $\frac{p_{100}^0}{p_0^0} = \frac{1.947}{5} = 0.4$, por lo que se cumple dicha condición para el punto de funcionamiento y valores cercanos.
- **xF_{100}** : Fracción de abertura de la válvula (0 implica válvula totalmente cerrada, y 1 totalmente abierta). Variable manipulable.
- **P_0, ρ_0** : Presión y densidad del vapor proveniente de la caldera. Unidades en bar y kg/m^3 , respectivamente.
- **K_{vs}** : Coeficiente de caudal. Para su cálculo es necesario evaluar la ecuación de la válvula en el punto de equilibrio. Tomando $xF_{100}^0 = 0.5$, se obtiene el siguiente valor de k_{vs} :

$$F_{100}^0 = 60 \cdot K_{vs} \cdot xF_{100}^0 \cdot 14.2 \cdot \sqrt{P_0^0 \cdot \rho_0^0} \rightarrow 9.3 = (60 \cdot K_{vs}) \cdot 0.5 \cdot 14.2 \cdot \sqrt{5 \cdot 0.375^{-1}} \rightarrow$$

$$K_{vs} = 0.005980 \frac{\text{m}^3/\text{h}}{\text{bar}^{0.5}} \quad (2-32)$$

Por otro lado, el caudal de condensado es expulsado al ambiente, mediante otra válvula. De manera previa a la válvula existe un pequeño tanque de almacenamiento de líquido, que tiene como objetivo almacenar líquido para que ante crecidas o caídas del caudal de condensado no se de el caso en el que a la descarga de la válvula llegue vapor en lugar de líquido.

$$F_c = 60 \cdot K_{vsc} \cdot \rho_{\text{agua}} \cdot x_c \cdot \sqrt{p_{100} + \frac{\rho_{\text{agua}}gh}{10^5}} - 1 \quad (2-33)$$

$$K_{vsc} = 0.003153 \frac{\text{m}^3/\text{h}}{\text{bar}^{0.5}}$$

- **ρ_{agua}** : Densidad del agua (1000 kg/m^3).
- **x_c** : Fracción de abertura de la válvula (0 implica válvula totalmente cerrada, y 1 totalmente abierta). En este proyecto, se mantiene constante en el punto de funcionamiento $x_c^0 = 0.5$.
- **g, h** : constante gravitatoria (9.81 m/s^2) y nivel de agua en el tanque de almacenamiento (0.2 metros), respectivamente.

2.5 Consideraciones

El modelo dinámico aquí utilizado considera diversas consideraciones y simplificaciones que se aclaran a continuación:

Presión P_2

- Se asume que la presión del sistema es P_2 en la entrada y salida del evaporador, así como en el separador. Esta afirmación no es del todo cierta puesto que es necesaria una diferencia de presión para que el líquido circule. En realidad, la presión a la entrada del evaporador es la presión P_2 más un incremento de presión, que se corresponde con el incremento de presión que proporciona la bomba.

Por otro lado, la altura geométrica a vencer depende de la configuración física de las tuberías de recirculación y del separador, y no suele ser muy significativa. En cabeza del evaporador, la altura del líquido es similar al nivel del separador, de modo que la bomba proporciona la energía necesaria para que el líquido pueda vencer las pérdidas de carga a través de los tubos del evaporador.

Por tanto, el incremento de presión de la bomba no se considera, de modo que P_2 es constante a lo largo del evaporador.

- La presión también depende del nivel de líquido al estar en un espacio confinado en el separador L-V (al descender líquido, el vapor se expande, y baja su presión). Esto no se ve reflejado en las ecuaciones pero la aproximación puede ser válida si el nivel de líquido no varía mucho con respecto al punto nominal (caso correspondiente al hecho de controlar L_2 con F_2 , como se verá mas adelante).
- El producto de salida se extrae desde la zona inferior de las tuberías de recirculación. Entre la salida del separador y la salida del producto final hay una columna de líquido que repercutirá en que la presión final del líquido extraído sea superior a la presión del líquido en el separador. De cara a la simplificación, se considera que la pérdida de presión producida en la válvula de extracción es aproximadamente igual al aumento de presión producido por la columna de líquido existente entre la salida del separador y la salida del producto. De esta manera, la presión de salida es aproximadamente igual a P_2 y la temperatura de dicho líquido es similar a T_2 .

Válvulas

- Los cambios en el grado de abertura de las válvulas suponen una evolución dinámica del caudal que no es despreciable. Sin embargo, puesto que el sistema responde en el orden de minutos, y los cambios en el caudal debido a las válvulas suponen una dinámica de segundos, sí se puede despreciar dicho efecto y, por tanto, considerar cambios instantáneos de caudal. Hay que tener en cuenta los límites superiores e inferiores físicos del caudal en la tubería, que se ven reflejados en la **Tabla 1.1** y que habrá que considerar en la simulación para no operar con valores físicamente inalcanzables.
- Por último, puesto que el sistema trabaja a una presión menor que la atmosférica (vacío relativo), es necesario un eyector aguas abajo del condensador que haga posible la extracción de caudal al exterior. Dicha situación no se ve reflejada en el modelado dinámico porque aumentaría la complejidad del mismo.

Evaporador

- De cara a mejorar la dinámica de T_2 se ha realizado el modelado del evaporador como sucesión de “N” tanques homogéneos. Se ha considerado que todas las porciones reciben la misma cantidad de calor proveniente del vapor existente en el carcasa. De esta manera, cada compartimento recibe una cantidad de calor igual a Q_{100}/N .
- Los coeficientes de transferencia de calor dependen en realidad de muchos factores, entre los que destaca el caudal, la turbulencia, viscosidad, etc. Como simplificación, se han considerado constantes estos coeficientes.

Caudal F_3

- El caudal F_3 es el caudal másico de recirculación del evaporador. La variación de este caudal depende de diversos factores: el nivel del separador, la altura de la columna de líquido entre el separador y la salida del producto y el caudal de salida de producto. Es, por tanto, una variable.

Dado que en este proyecto el nivel del separador se va a controlar directamente con el caudal de salida de producto manteniendo un punto de consigna fijo en el nivel, las variaciones de F_3 se consideran lo suficientemente pequeñas como para considerar dicho caudal constante e igual a 50 kg/min.

Diversos autores (Newell and Lee, otros) consideran un sistema evaporador ligeramente diferente en el cual la bomba de impulsión está situada con antelación a la salida de producto. Esto produce que el caudal F_3 sea dependiente de las fluctuaciones de caudal provocadas por la bomba además de las variables citadas anteriormente. La solución que proponen estos autores es considerar F_3 como una perturbación o como una variable manipulable. No se consideran como plausibles dichas propuestas

en este proyecto puesto que el caudal de recirculación no es una magnitud física que se pueda manipular directamente ni es una perturbación externa al sistema.

Las simplificaciones tomadas permiten realizar el estudio del sistema, desde el punto de vista de modelo dinámico, en un nivel académico aceptable y que mejora modelos anteriores. Si se quisiese hacer un estudio más preciso del sistema sería necesario acudir a un modelado más complejo que incluyese todos los detalles del evaporador real.

El diseño de controladores que se efectúa en este proyecto es consecuente con las ecuaciones aquí planteadas, pero es igualmente aplicable a un evaporador real con las limitaciones indicadas para el modelo. En dicho caso, deberían analizarse las funciones de transferencia según modelos SOPDT y FOPDT, obtenidos directamente de las respuestas del evaporador en funcionamiento, y luego se sintonizarían los controladores siguiendo el procedimiento recogido en este trabajo.

2.6 Resumen ecuaciones modelo y variables manipulables y de control

Las ecuaciones independientes del sistema son las siguientes:

(2-15),(2-17),(2-18),(2-19),(2-3),(2-4),(2-5),(2-6),(2-7),(2-10),(2-12),(2-22),(2-23),(2-24),(2-29),(2-31),(2-33)

Teniendo en cuenta que las 5 ecuaciones 2-10 a 2-14 desarrollan en realidad 5x3 ecuaciones ($N=3$), el sistema tiene un total de 27 ecuaciones independientes.

Por otro lado, a las 24 variables existentes declaradas en la **Tabla 1.1**, hay que añadir las variables del modelo del evaporador como subdivisiones de tanques homogéneos. En concreto, dichas variables son F_{21} , F_{22} , F_{41} , F_{42} , T_{21} , T_{22} , X_{21} , X_{22} , T_{w1} , T_{w2} , T_{w3} . Esto hace un total de 35 variables.

El número de grado de libertad es, por tanto, $35-27=8$. Es necesario aportar el valor de ocho de las variables para obtener una solución única, que son las variables de entradas manipulables y las perturbaciones. En concreto:

- 3 variables manipulables de entrada: $x_{F_{100}}$, F'_2 , F_{200} .
- 5 perturbaciones externas: F_1 , X_1 , T_1 , T_{200} , P_0 .

Las variables medibles de salida que se requieren controlar son:

- 3 salidas: X'_2 , P_2 , L_2 .

En consecuencia, el modelo dinámico planteado es consistente.

3 LINEALIZACIÓN

La herramienta matemática necesaria para poder trabajar cómodamente con el modelo dinámico en el diseño del sistema de control es la linealización. De esta manera, se consigue un sistema de ecuaciones lineales de un determinado orden en el dominio del tiempo, que se puede trasladar fácilmente al dominio de Laplace. Una vez se dispone de las transformadas y de las funciones de transferencia, se puede aplicar las técnicas de control lineal para controlar el sistema.

Para linealizar, se tomará como aproximación del sistema no lineal el primer término de la serie de Taylor alrededor del punto de equilibrio de las 27 ecuaciones independientes que conforman el modelo dinámico. La linealización de cada ecuación viene dada por la siguiente fórmula:

$$f(x_1, \dots, x_n) = \sum_{i=0}^n \left[\frac{\partial f}{\partial x_i} \right]_{pf} \cdot \Delta x_i \quad (3-1)$$

- f : función a linealizar
- x_i : variable de la ecuación dependiente del tiempo
- pf : punto de funcionamiento (equilibrio)
- n : número de variables de la ecuación
- Δx_i : variable incremental = $x_i - x_{pf}$

Nota: Toda función f así como cada variable x_i depende del tiempo (t). Por simplificación, se entiende que todas las variables son dependientes del dominio del tiempo o de la variable compleja 's' en el dominio de Laplace, en función de en qué dominio se esté trabajando, por lo que se omitirán los paréntesis aclaratorios en las variables.

A continuación se muestra la linealización de la ecuación (2-17):

$$\begin{aligned} f(F_1, F_4, X_2, X'_2, dX'_2, L_2) &= (F_1 + F_3^0 - F_4) \cdot (X_2 - X'_2) - \rho AL_2 \frac{dX'_2}{dt} = 0 \rightarrow \\ \left[\frac{\partial f}{\partial F_1} \right]^0 \cdot \Delta F_1 + \left[\frac{\partial f}{\partial F_4} \right]^0 \cdot \Delta F_4 + \left[\frac{\partial f}{\partial X_2} \right]^0 \cdot \Delta X_2 + \left[\frac{\partial f}{\partial X'_2} \right]^0 \cdot \Delta X'_2 + \left[\frac{\partial f}{\partial (dX'_2)} \right]^0 \cdot \frac{\Delta dX'_2}{dt} + \left[\frac{\partial f}{\partial L_2} \right]^0 \cdot \Delta L_2 &= 0 \quad (3-2) \\ (F_1^0 + F_3^0 - F_4^0) \cdot \Delta X_2 - (F_1^0 + F_3^0 - F_4^0) \cdot \Delta X'_2 - \rho AL_2^0 \cdot \frac{\Delta dX'_2}{dt} &= 0 \end{aligned}$$

Nota: el superíndice '0' sobre las variables hace referencia al punto nominal de funcionamiento

Aplicando las relaciones de transformación siguientes del dominio del tiempo al dominio de Laplace, en cada variable:

$$x(s) = \Delta x(t), \quad s \cdot x(s) = \frac{\Delta dx(t)}{dt} \quad (3-3)$$

Se obtiene:

$$X'_2(s) = \frac{1}{\frac{\rho AL_2^0}{(F_1^0 + F_3^0 - F_4^0)} s + 1} X_2(s) \quad (3-4)$$

Realizando el mismo proceso para cada una de las ecuaciones, se obtienen los resultados recogidos en la siguiente tabla:

Ecuación lineal alrededor del punto de funcionamiento (dominio de Laplace)	ec. no lineal	Elemento del evaporador
$\rho A \cdot sL_2 = F_1 - F_4 - F'_2$	(2-15)	separador
$X'_2 = \left(\frac{\rho AL_2^0}{(F_1^0 + F_3^0 - F_4^0)} s + 1 \right)^{-1} X_2$	(2-17)	separador
$C \cdot sP_2 = F_4 - F_5$	(2-18)	separador
$T_4 = 0.507P_2$	(2-19)	separador
$\frac{M}{N} Cp_L sT_{2,n-1} = (T_{2,n-1}^0 Cp_v + H_v(T_{2,n}^0)) F_{4,n-1} - (T_{2,n}^0 Cp_v + H_v(T_{2,n}^0)) F_{4,n} + (T_{2,n-1}^0 Cp_L) F_{2,n-1} - (T_{2,n}^0 Cp_L) F_{2,n} + (F_{2,n-1}^0 Cp_L + F_{4,n-1}^0 Cp_v) T_{2,n-1} - (F_{2,n}^0 Cp_L + F_{4,n}^0 Cp_v - \frac{h_{int} A_{int}}{N}) T_{2,n} + (\frac{h_{int} A_{int}}{N}) T_{w,n}$	(2-3)	Intercambiador
$\frac{M}{N} \cdot sX_{2,n} = (X_{2,n-1}^0) F_{2,n-1} + (F_{2,n-1}^0) X_{2,n-1} - (X_{2,n}^0) F_{2,n} - (F_{2,n}^0) X_{2,n}$	(2-4)	Intercambiador
$T_{2,n} = 0.5616P_2 + 0.3136X_{2,n}$	(2-5)	Intercambiador
$F_{2,n} = F_{2,n-1} + F_{4,n-1} - F_{4,n}$	(2-6)	Intercambiador
$M_w Cp_w \cdot sT_{w,n} = Q_{100} - h_{int} A_{int} T_{w,n} + h_{int} A_{int} T_{2,n}$	(2-7)	Intercambiador
$Q_{100} = UA_1 T_{100} - UA_1 T_2$	(2-10)	Intercambiador
$T_{100} = 0.1538P_{100}$	(2-12)	Intercambiador
$Q_{200} = \frac{UA_2}{1 + \frac{UA_2}{2C_p F_{200}^0}} T_4 - \frac{UA_2}{1 + \frac{UA_2}{2C_p F_{200}^0}} T_{200} + \frac{UA_2 Q_{200}^0}{F_{200}^0 (2C_p F_{200}^0 + UA_2)} F_{200}$	(2-22)	Condensador
$T_{201} = T_{200} + \frac{1}{C_p F_{200}^0} Q_{200} - \frac{Q_{200}^0}{C_p F_{200}^0} F_{200}$	(2-23)	Condensador
$F_5 = \lambda^{-1} Q_{200}$	(2-24)	Condensador
$\frac{18p_{100}^0}{0.082T_{100}^0} Cp_v \cdot sT_{100} = UA_1 T_2 - UA_1 T_{100} + (H_v(T_{100}^0) - Cp_v T_{100}^0) F_{100} + (H_v(T_{100}) - Cp_v T_{100}) F_{100}^0 - (H_L(T_{100}^0) - Cp_v T_{100}^0) F_c - (H_L(T_{100}) - Cp_v T_{100}) F_c^0$	(2-29)	Cámara

$F_{100} = \left(14.2 \cdot 60K_{vs} \sqrt{P_0^0 \rho_0^0} \right) xF_{100} + \left(\frac{14.2 \cdot 60K_{vs} \cdot xF_{100}^0 \cdot \rho_0^0}{2\sqrt{P_0^0 \rho_0^0}} \right) P_0$ $+ \left(\frac{14.2 \cdot 60K_{vs} \cdot xF_{100}^0 \cdot \rho_0^0}{2\sqrt{P_0^0 \rho_0^0}} \right) \rho_0$	(2-31)	Válvulas
$F_c = \frac{K_{vsc} \cdot 60 \cdot \rho_{agua} \cdot x_c}{2\sqrt{P_{100}^0 + \frac{\rho_{agua}gh}{10^5}} - 1} P_{100}$	(2-33)	Válvulas

Tabla 3.1 Linealizaciones en torno al punto de funcionamiento

Los puntos nominales de las variables, dependerán del evaporador que se esté usando. Para este proyecto, se usan los puntos nominales recogidos en la **Tabla 1.1**. Por otro lado, los puntos nominales de las variables del modelo del evaporador dividido en ‘N’ tanques homogéneos, se calculan con el modelo estático, anulando las derivadas de dichas ecuaciones y despejando los valores de equilibrio. El código utilizado para el cálculo de estos puntos de funcionamiento se pueden ver en Anexo A.

n	T _{2,n} (°C)	T _{w,n} (°C)	X _{2,n} (%)	F _{4,n} ($\frac{kg}{min}$)	F _{2,n} ($\frac{kg}{min}$)
1	83.82	87.2	22.5	2.25	57.75
2	84.2	87.6	23.7	5.15	54.85
3	84.6	88	25	8	52

Tabla 3.2 Puntos de funcionamiento variables intermedias evaporador

El sistema lineal proporciona una aproximación del modelo no lineal alrededor del punto nominal de funcionamiento. Estos resultados permiten obtener una idea aproximada de la relación entre las variables. De esta manera, se puede observar qué pasaría frente a cambios en ciertos puntos de funcionamiento (diferente comportamiento según el evaporador usado), y el diferente peso que tienen unas variables sobre otras. Por ejemplo, en la ecuación (2-23) linealizando en el punto nominal:

$$T_{201} = 1 \cdot T_{200} + 0.069 \cdot Q_{200} - 0.1 \cdot F_{200} \quad (3-5)$$

De donde se puede concluir que la variable que más afecta a la temperatura de salida del refrigerante, es la temperatura de entrada del mismo, seguido del caudal de refrigerante, ante cambios unitarios en las variables con las unidades elegidas.

Aunque con estas ecuaciones se pueden obtener las funciones de transferencia del sistema, el proceso es tedioso, por lo que más adelante (capítulo 5), se identificarán estas funciones G(s) por otro método más rápido y comúnmente utilizado si se dispone del sistema físico en funcionamiento o de un simulador.

4 MODELO EN SIMULINK-MATLAB

La implementación del modelo dinámico se ha realizado con la utilidad SIMULINK de MATLAB®. Esta herramienta permite crear un diagrama de bloques que integra las ecuaciones del modelo. En dicho diagrama, se pueden cambiar manualmente los valores de los puntos de equilibrio, entradas y perturbaciones para observar mediante un entorno gráfico el efecto de estos cambios en las salidas seleccionadas. Esta herramienta ofrece por tanto un método rápido e intuitivo para analizar y observar la dinámica del evaporador.

Se ha desarrollado el modelo de Simulink que implementa el modelo no lineal del evaporador, denominado “modelo_nolineal_final.slx”. Para abrir el archivo, se realiza la siguiente secuencia:

1. Abrir la aplicación “Matlab”, escribir “Simulink” y pulsar el botón “enter” para iniciar la aplicación. El tiempo de carga puede ser de un par de minutos. Es necesario tener los archivos “parametros.m” y “modelo_nolineal_final.slx” en el directorio de trabajo de Matlab (“Current Folder”).
2. Ejecutar el archivo “parámetros.m” y pulsar sobre el botón “Run”. Este archivo carga todos los puntos nominales de funcionamiento, así como las constantes del sistema y otros parámetros.
3. Abrir el archivo “modelo_nolineal_final.slx”. Aquí se puede observar el diagrama completo ya creado del modelo dinámico.
4. Para ejecutar la simulación, sólo es necesario especificar el periodo de tiempo en la parte superior de la ventana y pulsar el botón “Run”. Al hacer esto, y transcurridos unos segundos, se habrá realizado la simulación para el tiempo especificado (en minutos, según las unidades utilizadas en el modelo) y se puede acceder a las ventanas “Scope” para observar la respuesta de las variables de salida.
5. Para realizar diferentes simulaciones, sólo es necesario cambiar los parámetros deseados, que son, típicamente, las variables manipulables y las perturbaciones, y volver a pulsar el botón “Run”. Por defecto, la configuración inicial del archivo dispone de todos los parámetros en sus puntos nominales de funcionamiento.

En la **Figura 4.1** se puede observar la pantalla principal del diagrama de bloques del modelo no lineal:

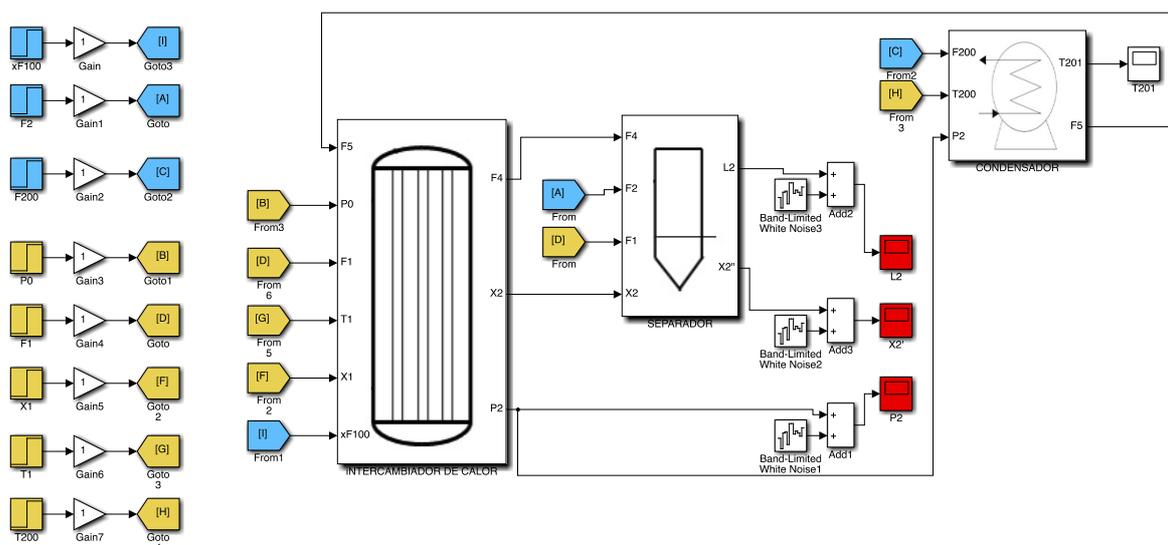


Figura 4.1 Pantalla principal modelo no lineal Simulink

- A la izquierda de la pantalla hay una columna de escalones. Estos escalones representan los puntos de funcionamiento de las 3 variables manipulables y las 5 perturbaciones del sistema. Las ganancias conectadas a continuación son el valor por el que se multiplica el punto de funcionamiento, dando como resultado las variaciones de estas variables, modificables por el usuario.

- Hay tres bloques principales: “EVAPORADOR(Intercambiador de calor)”, “SEPARADOR” y “CONDENSADOR”. Pulsando sobre cada uno de ellos se pueden observar varios niveles de bloques que comprenden todas las ecuaciones desarrolladas en el capítulo 2 de este proyecto. Para ver dichos niveles se puede abrir directamente el archivo de Simulink adjunto al proyecto o acudiendo al ANEXO B donde se recogen también las imágenes de estos diagramas.
- Diversos bloques tienen un color específico. Estos colores representan la diferente funcionalidad de los bloques, en concreto:
 - Azul: Variables manipulables del sistema, típicamente escalones.
 - Amarillo: Variables de perturbación del sistema, típicamente escalones.
 - Rojo: Salidas del sistema, típicamente bloques “Scope” que permiten observar su evolución temporal.
 - Verde: Variables intermedias.

Para la simulación, el usuario sólo tiene que cambiar las ganancias conectadas a los escalones para modificar el valor de estas variables con respecto al punto de funcionamiento. Es necesario tener en cuenta los límites establecidos en la **Tabla 1.1**

A modo de ejemplo, para observar el efecto de una variación en x_{F100} sobre X_2' , se introduce el valor 1.1 en “Gain” (incremento del 10% de x_{F100} con respecto a su punto de funcionamiento), obteniéndose el siguiente resultado al pulsar sobre el bloque Scope X_2' :

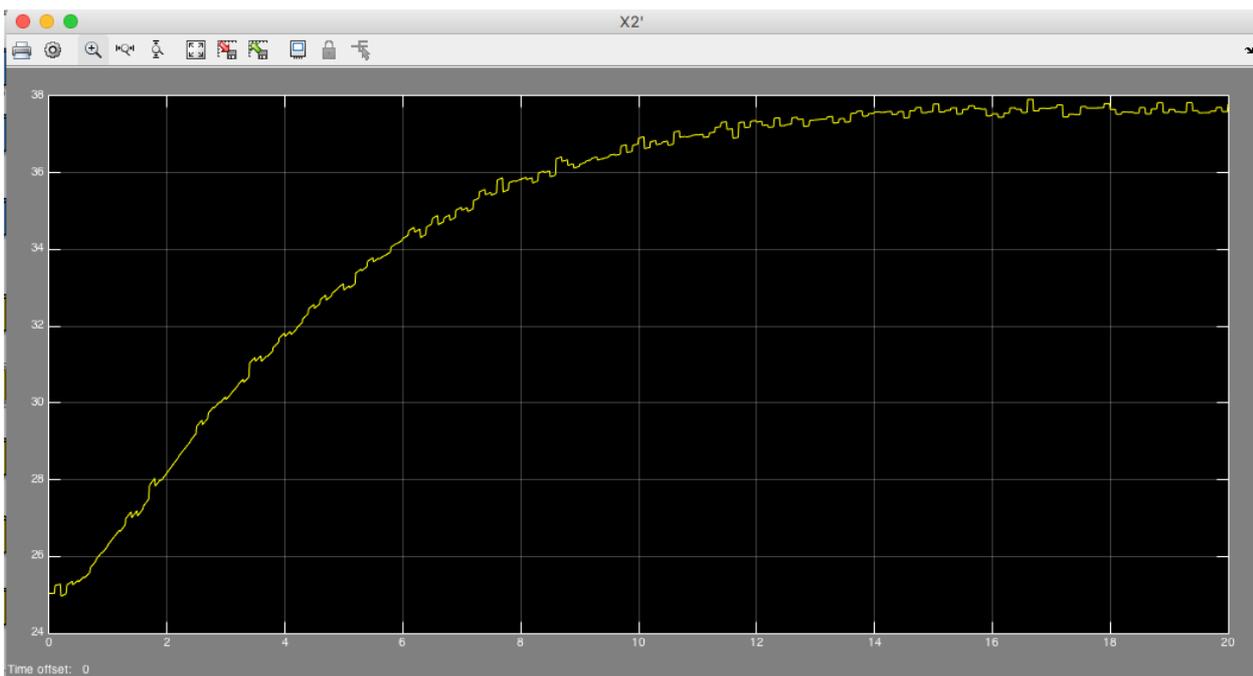
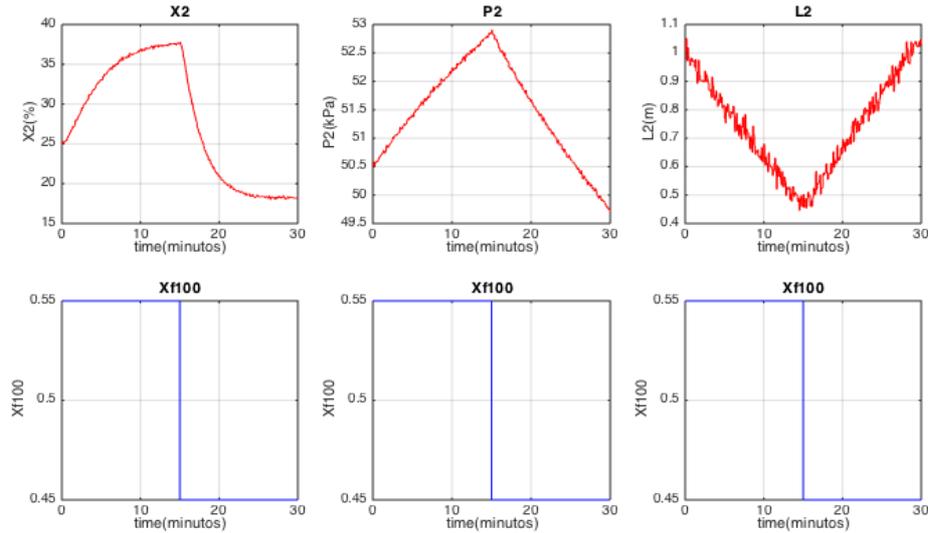


Figura 4.2 Ejemplo de respuesta de simulación

En los capítulos siguientes se utiliza este modelo para las simulaciones, así como la modificación pertinente del diagrama para la implementación de control, como se indicará en los correspondientes apartados y que viene recogida también en el ANEXO B.

A continuación se muestra y se comenta el efecto de cada variable manipulable sobre cada variable de control:



- Un cambio en la apertura de la válvula X_{F100} provoca un cambio de tipo “2º orden” en la concentración final de producto. Esto se debe al cambio de temperatura y, por tanto, al cambio de calor transferido que se produce al líquido de los tubos al introducir una mayor o menor cantidad de flujo vapor en la cámara del intercambiador.
- Un cambio en la apertura de la válvula X_{F100} provoca un cambio en la cantidad de vapor producido en el evaporador, que afecta directamente a la presión P_2 . En este caso la dinámica es más lenta que la evolución de X_2' , y es de primer orden.
- El nivel del separador se ve directamente afectado por el aumento o disminución de cantidad de vapor producida por el cambio en la apertura de X_{F100} , presentando una respuesta de tipo integrador.

Figura 4.3 Respuestas variables de control frente a variación en X_{F100}

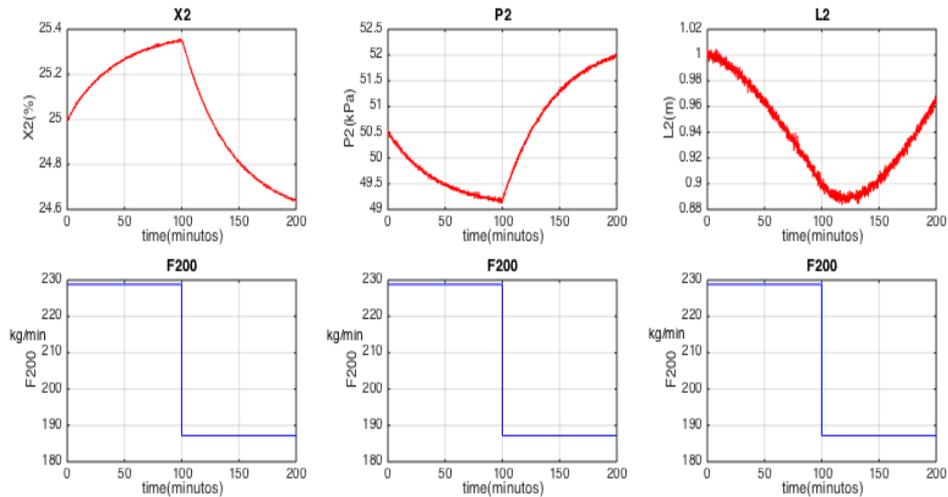


Figura 4.4 Respuestas variables de control frente a variación en F_{200}

- Al producirse una variación en F_{200} , X_2' reacciona como un sistema de primer orden. En la **Ilustración 4.4** se refleja la evolución producida en dicha variable de control. Al aumentar la refrigeración, el sistema evaporador es capaz de enfriar un mayor caudal de vapor, lo que permite una mayor evaporación y por tanto una mayor concentración de producto. De forma opuesta es la respuesta del sistema al disminuir la refrigeración.
- La presión del evaporador disminuye al aumentar la refrigeración, y aumenta al bajar la corriente de refrigeración, presentando una respuesta que se ajusta a un sistema de primer orden.

- L_2 varía en relación a X_2 con una respuesta de tipo integrador. Al cambiar la corriente refrigerante, el mayor o menor caudal de vapor producido afecta directamente a la cantidad de líquido en el separador.

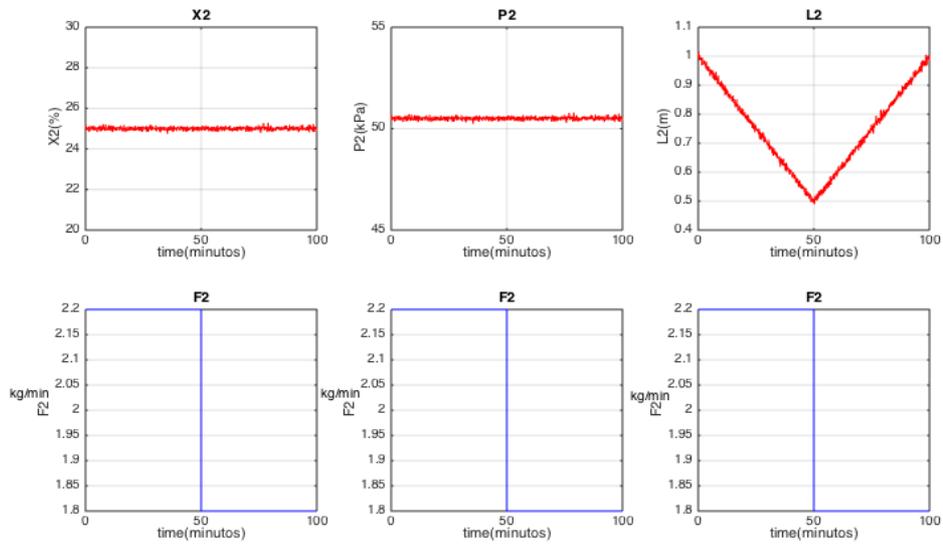


Figura 4.5 Respuestas variables de control frente a variación en F_2

- Frente a cambios en la corriente de producto, la variación de la concentración de producto y la variación de la presión del evaporador. Esto se debe a las consideraciones tomadas con respecto a F_3 (ver 2.6 Consideraciones).
- El nivel de líquido del separador es directamente afectado por el caudal de salida de producto. Así, al aumentar dicho caudal, disminuye la cantidad de líquido en el separador, y viceversa. La respuesta que se obtiene es de tipo integrador.

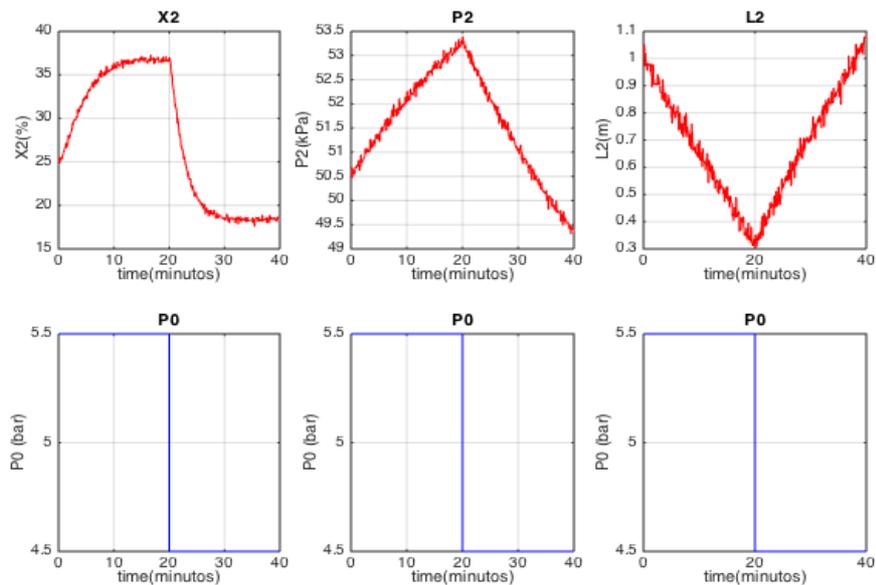


Figura 4.6 Respuestas variables de control frente a variación en P_0

- Los cambios en la presión de vapor proveniente de una caldera afectan a las variables de control de manera similar a los cambios producidos por variaciones en la abertura de la válvula. Esto se debe a que el flujo de entrada a la cámara de evaporador F_{100} depende proporcionalmente tanto de X_{F100} como de P_0 .

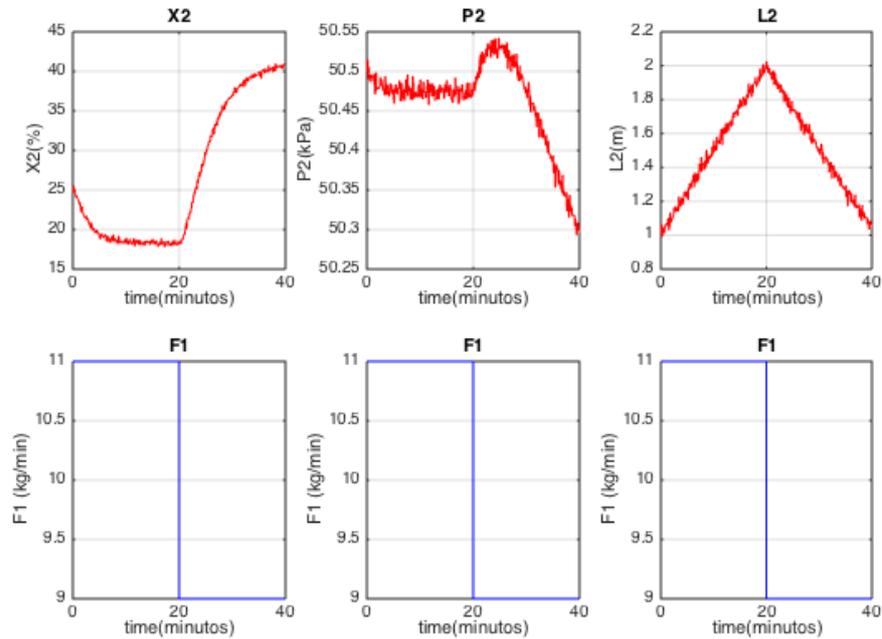


Figura 4.7 Respuestas variables de control frente a variación en F_1

- Los cambios en la corriente de alimentación son ilustrativos de la no linealidad del sistema. Así, un aumento de dicho caudal produce una caída de concentración de producto mucho más rápida que la subida de concentración provocada por una disminución de caudal de alimentación (ambos con respuesta de 2º orden). Dicho comportamiento no lineal también se ve reflejado en la variación de la presión del evaporador frente a cambios en f_1 .
- La relación entre el nivel del separador y el caudal de alimentación es más directa. Aumentar el caudal de entrada implica, según el balance másico, un aumento en el nivel de líquido del separador, y viceversa. Este cambio responde a una función de tipo integrador.

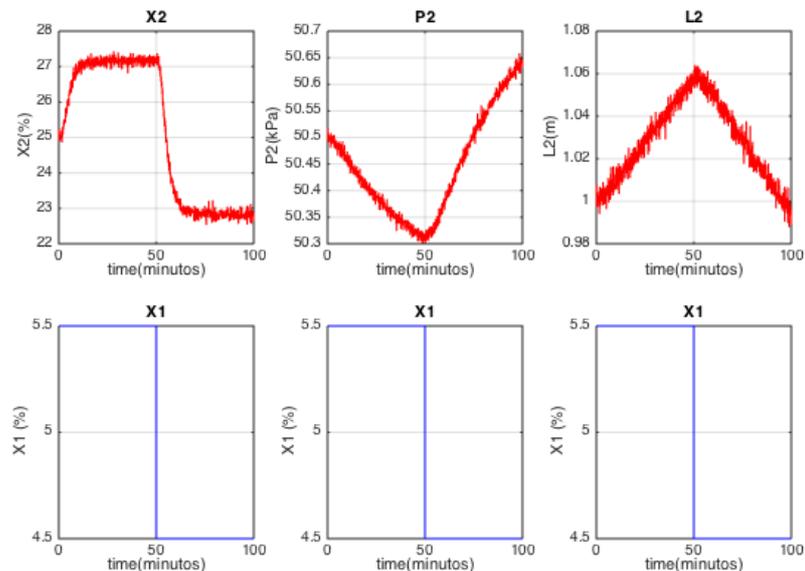


Figura 4.8 Respuestas variables de control frente a variación en X_1

- La concentración del líquido de alimentación afecta directamente a la concentración final de producto. X_2 evoluciona como un sistema de 2 orden frente a cambios en X_1 .
- La presión del evaporador responde de manera inversa ante los cambios en la concentración de entrada. Un aumento de X_1 produce una disminución en P_2 y viceversa. Esto se debe a que una mayor concentración de sal en el líquido de los tubos provoca que frente a la misma cantidad de calor transferido se produzca una menor cantidad de vapor de agua, lo que disminuye la presión del sistema.
- El nivel del separador está relacionado, con la cantidad de vapor de agua producido y la concentración final de producto, presentando una respuesta de tipo integrador.

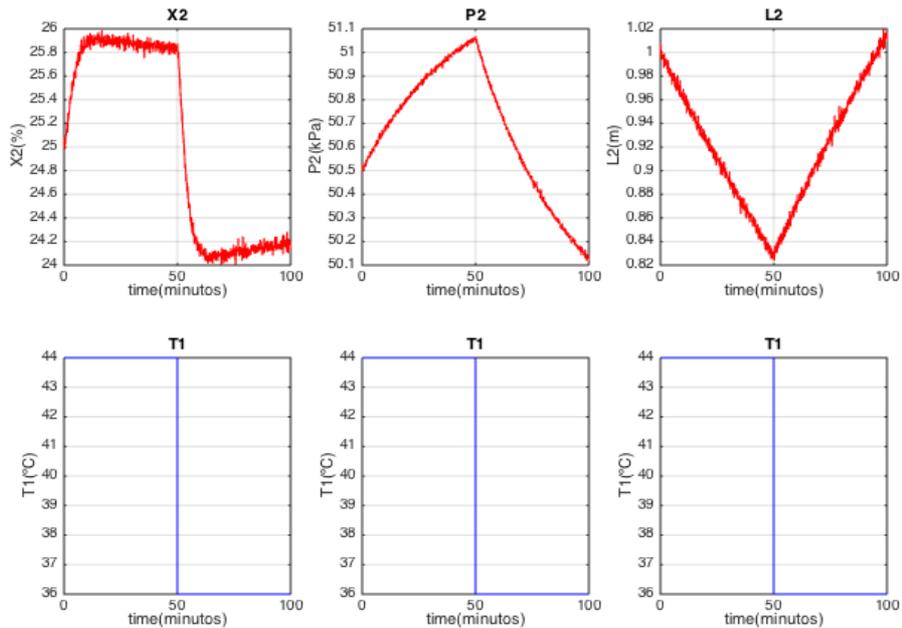


Figura 4.9 Respuestas variables de control frente a variación en T_1

- Los cambios positivos en la temperatura del líquido de alimentación producen un aumento de la presión del evaporador, ya que el sistema pierde capacidad de refrigeración al aumentar la temperatura de la mezcla sin variar el caudal de entrada en el condensador. La respuesta de P_2 frente a T_1 es de tipo lineal.
- Frente a cambios T_1 , X_2 reacciona con una respuesta de segundo orden proporcional a la entrada, y L_2 reacciona con una respuesta de tipo integrador, inversamente proporcional con respecto a cambios en T_1 .

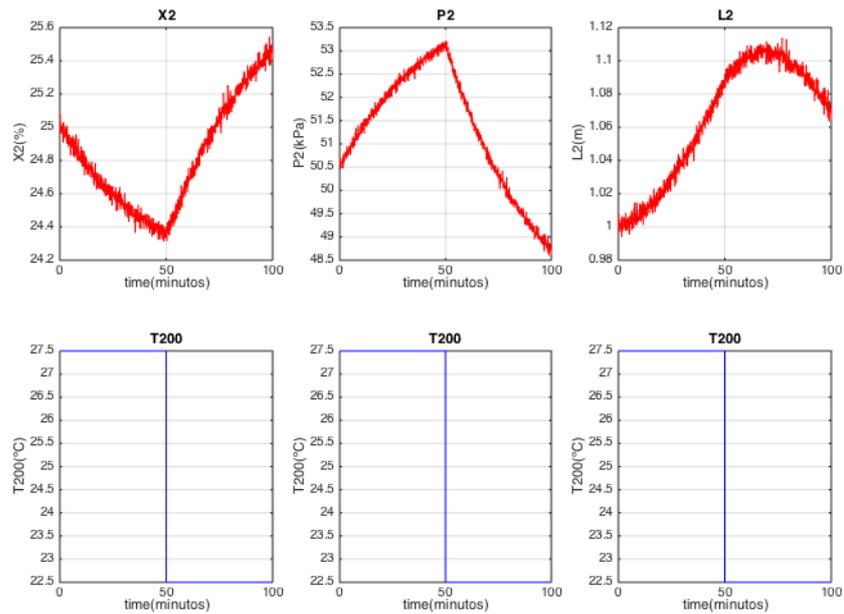


Figura 4.10 Respuestas variables de control frente a variación en T_{200}

- Los cambios en la temperatura del líquido refrigerante afectan directamente a la presión del evaporador. Cuanto mayor sea la temperatura de refrigeración, menos vapor se podrá condensar y por tanto mayor será la presión del evaporador (y viceversa). La respuesta de P_2 frente a cambios en T_{200} es de primer orden.
- El nivel del separador cambio de manera directamente proporcional frente a cambios en T_{200} . La respuesta de L_2 se asemeja a un sistema integrador con retardo.
- La concentración de producto final es inversamente proporcional a los cambios en la temperatura del refrigerante. El sistema no puede condensar la cantidad de vapor del sistema funcionando en torno a su punto nominal, por lo que la concentración final de producto disminuye ante un aumento en la temperatura del refrigerante. La respuesta es de segundo orden.

5 DISEÑO DEL SISTEMA DE CONTROL

5.1 Identificación G(s)

El sistema físico a controlar consta de 9 parejas variable manipulada-variable controlada. Para el desarrollo posterior de la DRGA y el sistema de control, es necesario conocer la función de transferencia de dichos emparejamientos. En este proyecto, se utilizan modelos de primer orden con retardo puro o tiempo muerto (FOPDT – First Order Plus Dead Time), excepto en los casos donde esta simplificación no represente adecuadamente la dinámica del sistema, casos que se detallarán más adelante.

Dichas funciones se pueden obtener por medio de diversos métodos. Un método tradicional es obtener experimentalmente los parámetros de la función de transferencia observando la curva de reacción de la variable de salida frente a un cambio de tipo escalón de cierta amplitud en la entrada, atendiendo a las siguientes relaciones [7]:

$$G(s) = \frac{K}{\tau s + 1} e^{-\theta s} \quad (5-1)$$

Sea “u” la entrada e “y” la salida:

$$t_1 = \text{tiempo para } dy = 0.283(dy)_\infty$$

$$t_2 = \text{tiempo para } dy = 0.632(dy)_\infty$$

$$K = \text{ganancia} = \frac{(dy)_\infty}{(du)_\infty}$$

$$\tau = \text{constante de tiempo} = 1.5(t_2 - t_1)$$

$$\theta = \text{tiempo muerto} = t_2 - \tau$$

Nota: el subíndice ‘∞’ indica régimen permanente

Si θ saliese negativo, se usa:

$$\tau = t_2, \quad \theta = 0$$

Aunque este método suele ser válido, existen otras herramientas que agilizan el proceso y ofrecen resultados más precisos. Para el desarrollo final de la matriz de funciones de transferencia, se ha utilizado la herramienta “simulation toolbox MATLAB®”.

Para ello, se guardan los vectores “u” e “y” de cada emparejamiento “entrada-salida”, que almacenan una lista de datos muestreados de las entradas y salidas, respectivamente. Dichos vectores se pueden obtener guardando como variable el resultado de los bloques “Scope” del modelo de “Simulink”. Posteriormente, y con dicha información almacenada, se ejecuta el comando “ident”, que abre la GUI “System Identification”.

Finalmente, en dicha interfaz se selecciona los emparejamientos de entradas y salidas y el tipo de función de transferencia al que se quiere ajustar. A continuación, se explica el procedimiento seguido para la pareja X2'-xF100, realizando un cambio de tipo escalón de amplitud 5% en la entrada:

1. Simulación Scope y guardado de la respuesta en la variable “y” (de la misma manera se guarda la entrada como “u”), como se puede apreciar en la **Figura 5.1**.

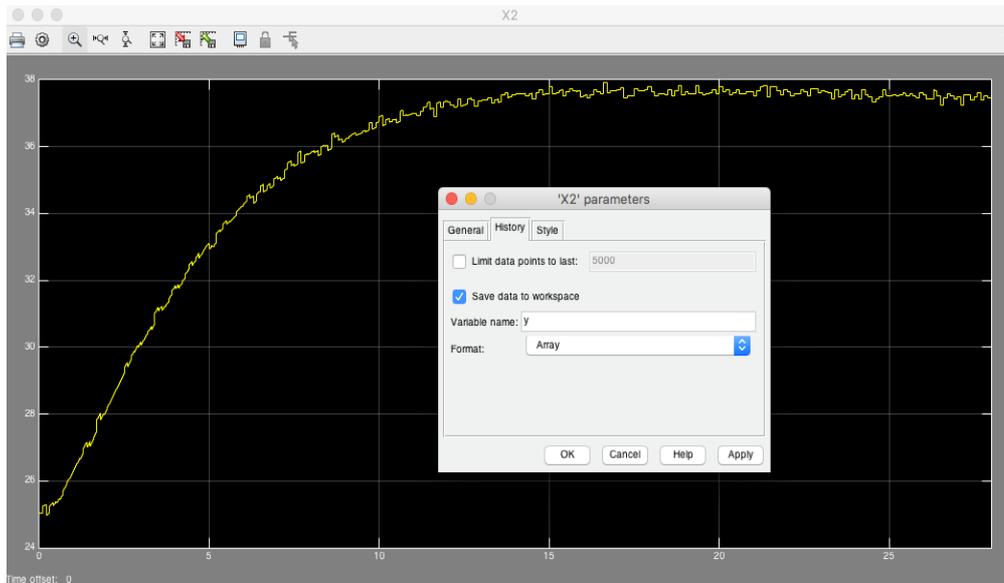


Figura 5.1 respuesta en scope-Simulink

2. Dado que dichas variables son matrices que contienen dos vectores columnas de tiempo (eje x) y de P2 (eje y), se asignan 2 nuevas variables en Matlab, que almacenen únicamente el vector columna de la respuesta (sin el tiempo). Además, se le resta a todos los elementos del vector el punto de equilibrio, para así obtener variables de desviación, y poder identificar la función de transferencia correctamente. Los comandos a introducir en Matlab son, por tanto:

$$u = u(:,2) - xF100E;$$

$$y = y(:,2) - X2E;$$

3. Una vez almacenadas dichas variables de entrada y salida, se ejecuta el comando “ident” para abrir la interfaz “System Identification”.
4. Dentro de la interfaz, se selecciona “Import Data”-> “Time Domain Data”, y se introducen los datos correspondientes en “input” y “output”. Los parámetros “starting time” y “sample time”, toman respectivamente los valores 0 y 0.01 (este último debido a que se ha configurado en Matlab que se guarden las variables procedentes de “Scope” con un muestreo temporal de 0.01).
5. Finalmente, se selecciona la opción “Estimate”->”Process Model”. En la ventana emergente se puede elegir la opción correspondiente al tipo de sistema lineal al que se quiere ajustar la respuesta. Como se mencionó previamente, normalmente se utilizará un modelo FOPDT, obteniendo en este caso los siguientes resultados (**Figura 5.2**):

$$K = 199.16 \frac{\% \text{ soluto}}{1}, \quad \tau = 2.7 \text{ min}, \quad \theta = 1.1 \text{ min}$$

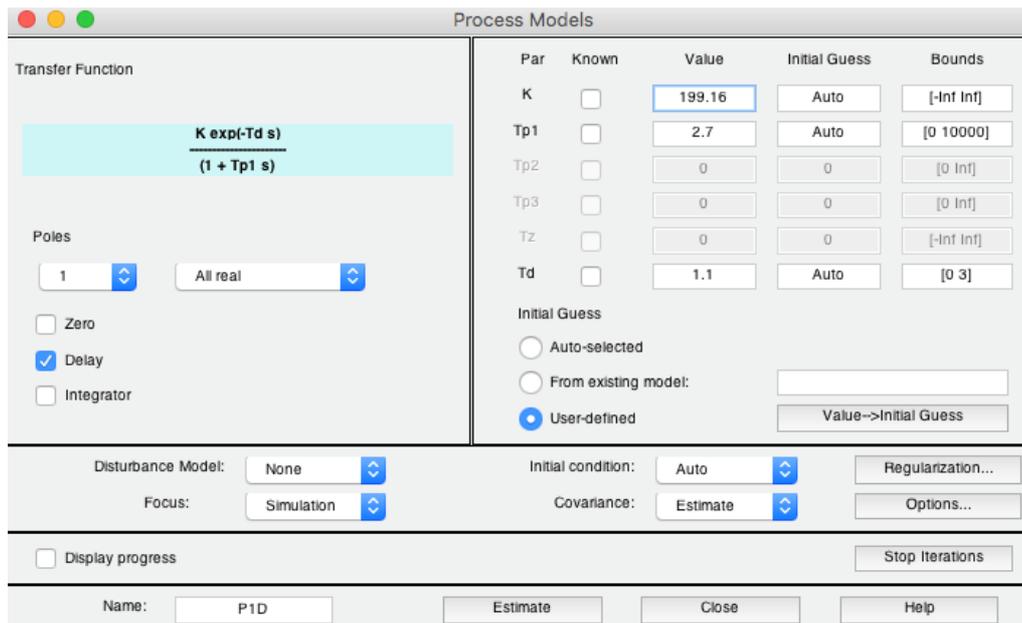


Figura 5.2 Pantalla de estimación de modelo Ident

6. Clicando sobre “Model output” en la interfaz principal, se puede observar la respuesta del modelo original y del modelo simplificado, así como el porcentaje de ajuste de la función de transferencia aproximada. En la **Figura 5.3** se puede observar esta comparación para el modelo FOPDT de X_2/X_{F100} obtenido para un escalón en la entrada del 5%:

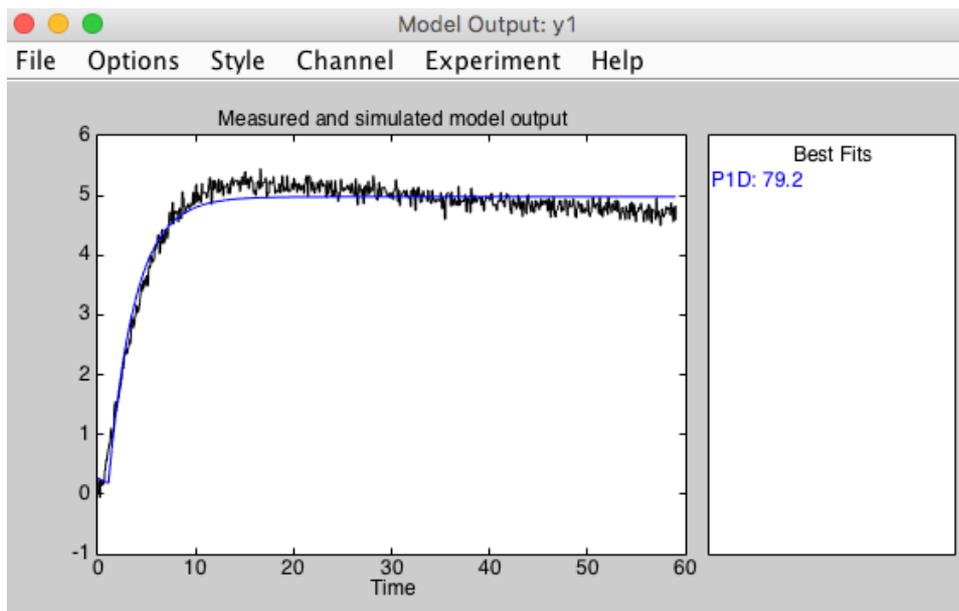


Figura 5.3 test ajuste respuesta

Siguiendo el mismo proceso para los 9 emparejamientos variable controlada-variable manipulada se obtiene la siguiente matriz:

$$\begin{bmatrix} X'_2(s) \\ P_2(s) \\ L_2(s) \end{bmatrix} = G(s) \cdot \begin{bmatrix} xF_{100}(s) \\ F_{200}(s) \\ F'_2(s) \end{bmatrix} \quad (5-2)$$

La matriz dada por la ecuación (5-3) representa las funciones de transferencias frente a entradas de escalón positivo del 10% con respecto al punto nominal de funcionamiento.

$$G_+(s) = \begin{bmatrix} \frac{199.16}{2.7s + 1} e^{-1.1s} & \frac{0.0185}{43.371s + 1} e^{-0.7s} & 0 \\ \frac{161.64}{43.76s + 1} & \frac{-0.07123}{42.944s + 1} & 0 \\ \frac{-0.6889}{s} & \frac{-5.327 \cdot 10^{-5}}{s} & \frac{-0.05}{s} \end{bmatrix} \quad (5-3)$$

El mismo procedimiento se debe realizar estudiando el proceso con escalones negativos. Así, si se realiza el mismo proceso anterior de identificación, pero con escalones del -10% con respecto al punto nominal, se obtiene la (5-4):

$$G_-(s) = \begin{bmatrix} \frac{142.9}{2.25s + 1} e^{-0.76s} & \frac{0.0215}{43.900s + 1} e^{-0.3s} & 0 \\ \frac{157.62}{41.71s + 1} & \frac{-0.08722}{44.702s + 1} & 0 \\ \frac{-0.6860}{s} & \frac{-6.353 \cdot 10^{-5}}{s} & \frac{-0.05}{s} \end{bmatrix} \quad (5-4)$$

Los modelos obtenidos con esta herramienta pueden ser tan complicados y de tan alto orden como se desee. Sin embargo, para un sistema de control realimentado, interesa que el modelo sea lo más sencillo posible sin perder la esencia de la dinámica del sistema. Por eso, se han elegido modelos FOPDT o modelos integradores.

Es importante destacar también el hecho de que los términos $G_{13}(s)$ y $G_{23}(s)$ sean 0. Esto se debe a una suposición hecha anteriormente en el modelo dinámico según la cual la variación de F_3 es nula.

La matriz $G(s)$ concuerda con esta suposición, pues la variación de las salidas X'_2 y P_2 dependen directamente del caudal F_3 . Esta variación de F_3 es prácticamente nula al controlar L_2 con F'_2 , como se verá más adelante.

Como se puede apreciar, las funciones de transferencia son diferentes en función al escalón negativo o positivo introducido en la entrada. Esto se debe a la no linealidad del sistema. Como aproximación, se toma una $G(s)$ final cuyos parámetros son la media aritmética de los parámetros de las 2 matrices anteriores [6]:

$$G(s) = \begin{bmatrix} \frac{171}{2.475s + 1} e^{-0.93s} & \frac{0.02}{43.64s + 1} e^{-0.5s} & 0 \\ \frac{159.6}{42.74s + 1} & \frac{-0.07923}{43.82s + 1} & 0 \\ \frac{-0.6875}{s} & \frac{-5.84 \cdot 10^{-5}}{s} & \frac{-0.05}{s} \end{bmatrix} \quad (5-5)$$

Las respuestas de la variable de control L_2 presentan siempre una respuesta de tipo integrador. Esto se debe a la naturaleza dinámica del separador. Cualquier variación permanente con respecto al punto de nominal de funcionamiento de cualquier variable manipulable provocará una bajada o subida inestable del nivel del separador en régimen permanente (hasta alcanzar los límites de funcionamiento 0-2 m).

A continuación se muestran las comparaciones entre las respuestas de las variables de control del sistema no lineal y de las funciones lineales, así como la bondad de los ajustes, frente a variaciones en las variables manipulables (a la izquierda frente a cambios del +10% y a la derecha frente a cambios del -10%). El color negro representa la respuesta de las variables de control del sistema no lineal frente a escalones en la entrada de amplitud $\pm 10\%$. El color azul representa la respuesta de las variables de control del sistema lineal $G_+(s)$ (5-3) frente a escalones en la entrada de amplitud $\pm 10\%$, el color verde representa la respuesta de las variables de control del sistema lineal $G_-(s)$ (5-4) frente a escalones en la entrada de amplitud $\pm 10\%$, y el color rojo representa la respuesta de las variables de control del sistema lineal $G(s)$ (5-5) frente a escalones en

la entrada de amplitud $\pm 10\%$.

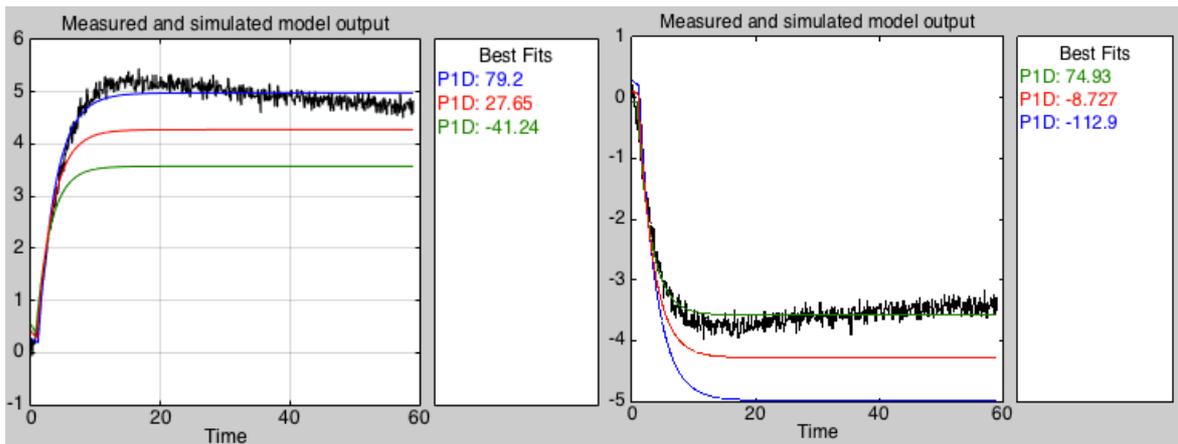


Figura 5.4 Respuesta sistema no lineal-lineal. X_2' frente a escalón amplitud $\pm 10\%$ en X_{F100} , y ajuste

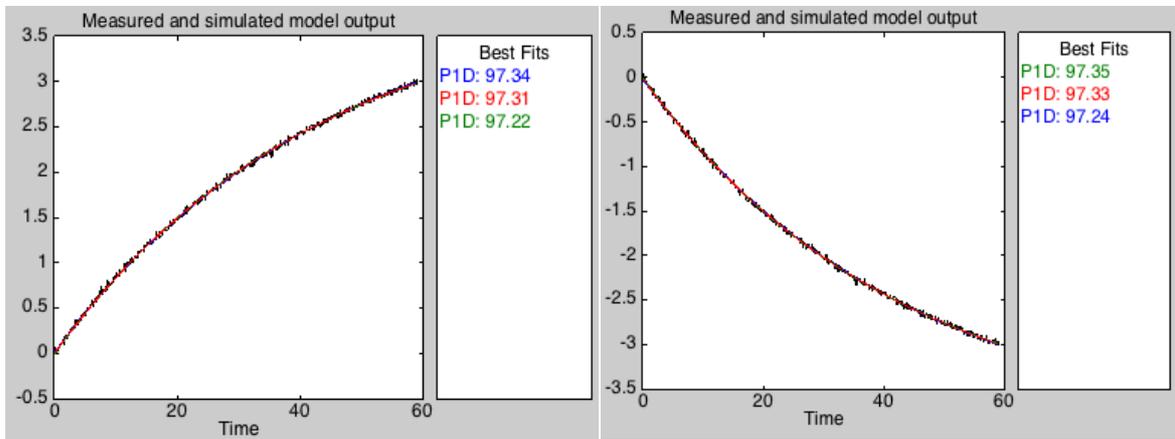


Figura 5.5 Respuesta sistema no lineal-lineal. P_2 frente a escalón amplitud $\pm 10\%$ en X_{F100} , y ajuste

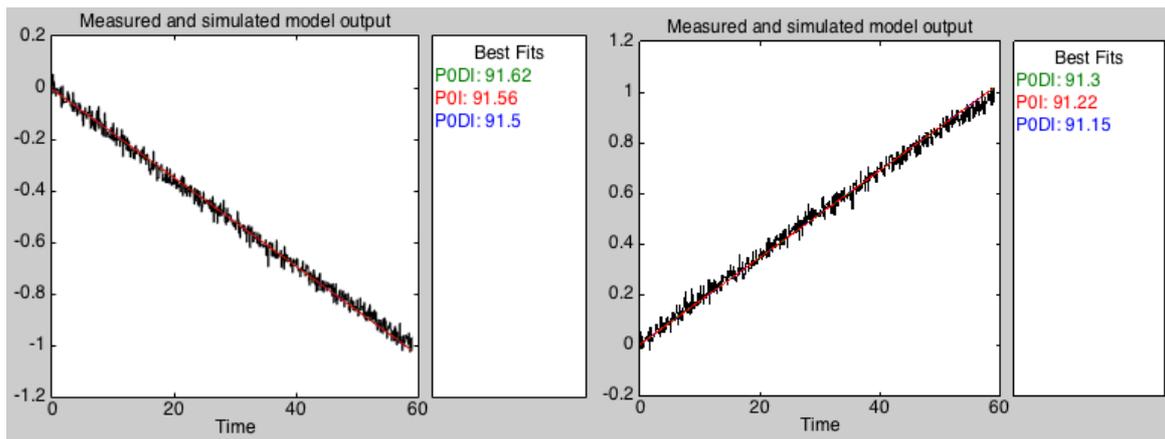


Figura 5.6 Respuesta sistema no lineal-lineal. L_2 frente a escalón amplitud $\pm 10\%$ en X_{F100} , y ajuste

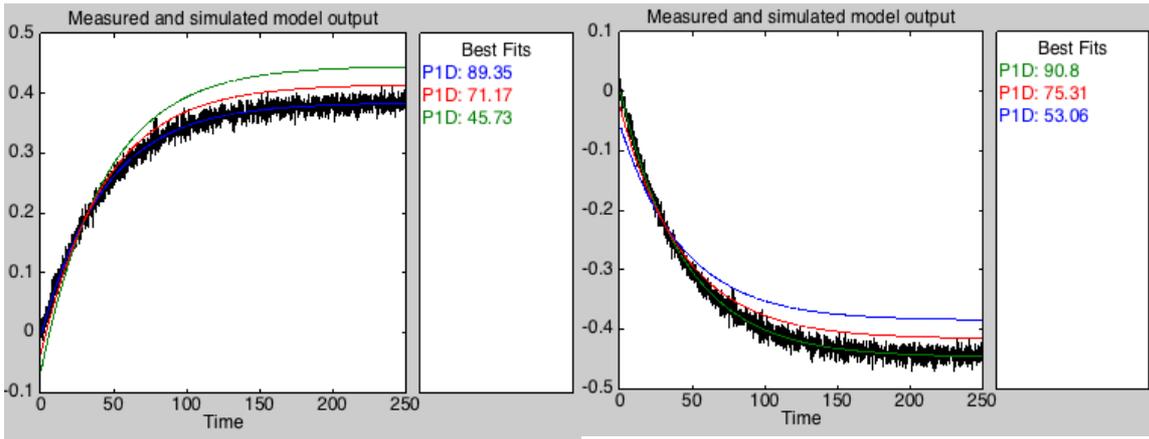


Figura 5.7 Respuesta sistema no lineal-lineal. X_2' frente a escalón amplitud $\pm 10\%$ en F_{200} , y ajuste

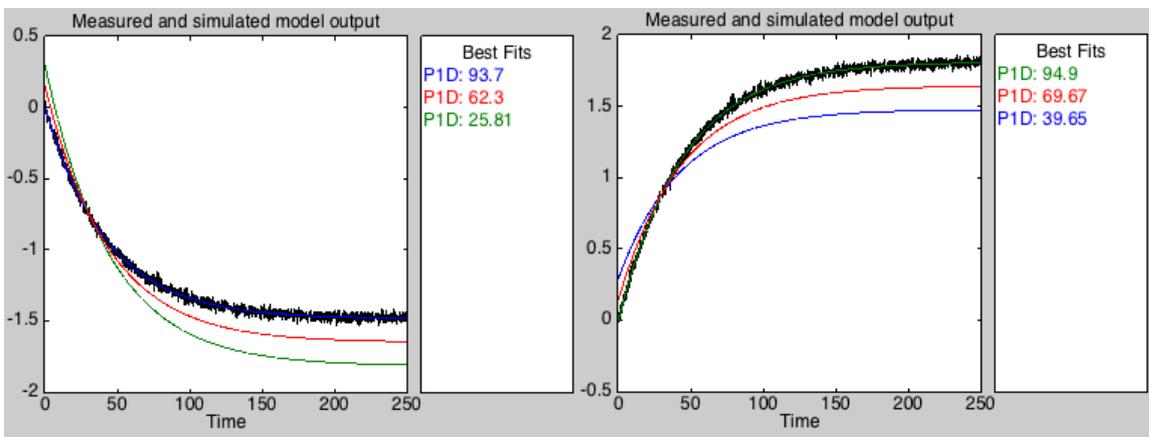


Figura 5.8 Respuesta sistema no lineal-lineal. P_2 frente a escalón amplitud $\pm 10\%$ en F_{200} , y ajuste

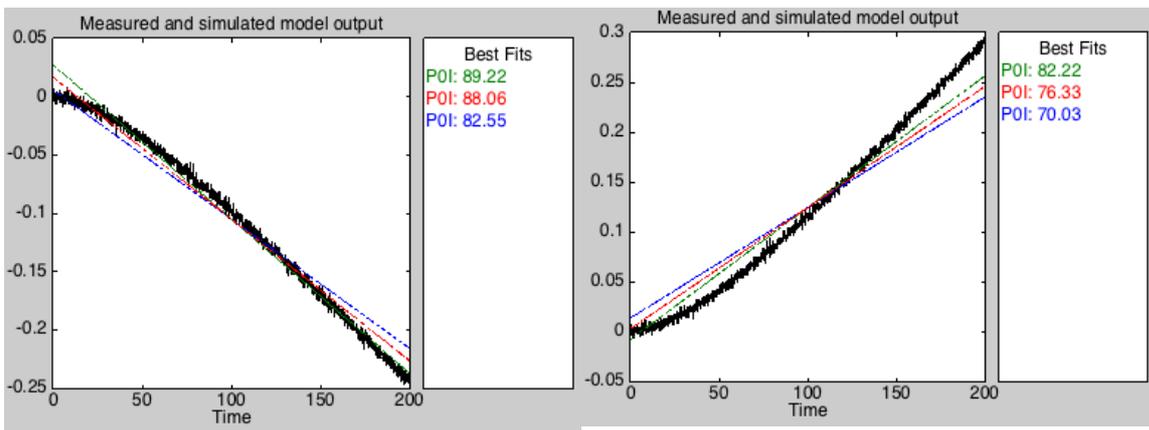


Figura 5.9 Respuesta sistema no lineal-lineal. L_2 frente a escalón amplitud $\pm 10\%$ en F_{200} , y ajuste

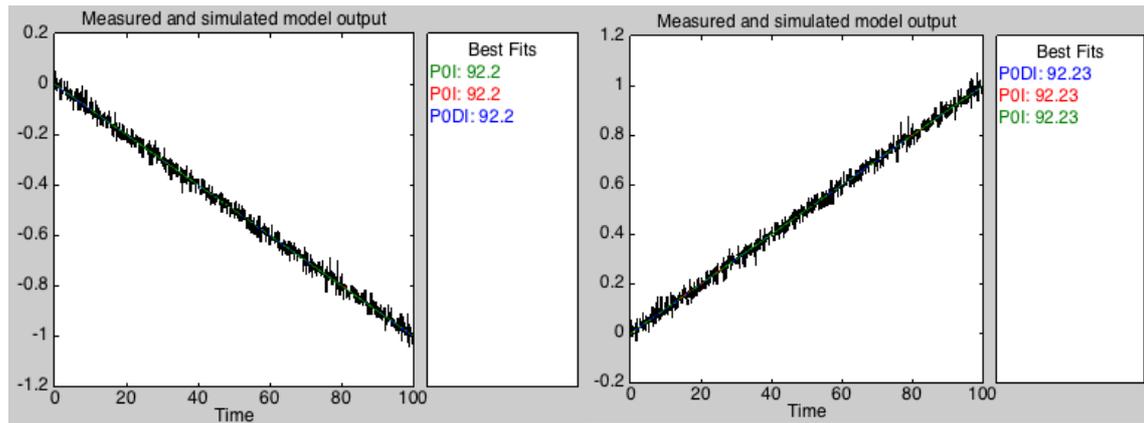


Figura 5.10 Respuesta sistema no lineal-lineal. L_2 frente a escalón amplitud $\pm 10\%$ en F_2 , y ajuste

En todas las simulaciones se repite el mismo patrón:

- Color Azul. Las respuestas lineales correspondientes a $G_+(s)$ se ajustan mucho mejor a la respuesta no lineal frente a escalones positivos en la variable manipulable del 10% de amplitud. Sin embargo, frente a entradas de -10% de amplitud, la bondad del ajuste es peor que las otras 2 funciones de transferencia. Esto se debe a cuestiones de diseño, pues $G_+(s)$ se diseña en función de entradas de +10% de amplitud con respecto al punto de funcionamiento.
- Color Verde. Las respuestas lineales correspondientes a $G_-(s)$ se ajustan mucho mejor a la respuesta no lineal frente a escalones en la variable manipulable del -10% de amplitud. Al contrario que en el caso anterior, es frente a escalones positivos en la variable manipulable donde las respuestas lineales de $G_-(s)$ tienen un peor ajuste. Se debe de nuevo al diseño de la función de transferencia.
- Color Rojo. Las respuestas lineales correspondientes a $G(s)$ ofrecen una bondad de ajuste intermedia entre $G_+(s)$ y $G_-(s)$. Utilizar esta función es la opción más adecuada frente a simulaciones en las que se cambien las variables manipulables tanto con escalones positivos como negativos con respecto a su punto de funcionamiento.

Cuanto mayor sea la amplitud de entrada con respecto al punto de funcionamiento de la variable manipulable, peor será la bondad del ajuste del sistema lineal. Esto se debe a la no linealidad del sistema. En este proyecto se ha considerado en las simulaciones escalones de hasta el 20% con respecto al punto nominal de las variables manipulables en las simulaciones, pues en estos casos los valores alcanzados en régimen permanente de las variables controladas lineales tienen un error con respecto a las no lineales menor siempre al 25%. La pérdida de bondad de simulación conforme mayor sean estas amplitudes se traduce en una peor efectividad de los controladores diseñados a la hora de seguir referencias y rechazar perturbaciones.

5.2 Identificación $G_d(s)$

Siguiendo el mismo procedimiento usado en el apartado anterior, se define la matriz $G_d(s)$:

$$\begin{bmatrix} X'_2(s) \\ P_2(s) \\ L_2(s) \end{bmatrix} = G_d(s) \cdot \begin{bmatrix} P_0(s) \\ F_1(s) \\ X_1(s) \\ T_1(s) \\ T_{200}(s) \end{bmatrix} \quad (5-6)$$

La matriz dada por la ecuación (5-7) representa las funciones de transferencias frente a perturbaciones de escalón positivo del 10% con respecto al punto nominal de funcionamiento.

$$G_{d+}(s) = \begin{bmatrix} \frac{23.275}{4.269s + 1} e^{-0.3s} & \frac{-6.729}{2.675s + 1} e^{-0.3s} & \frac{4.376}{4.489s + 1} e^{-0.9s} & \frac{0.21}{2.67s + 1} e^{-0.8s} & \frac{-0.37}{42.34s + 1} e^{-0.3s} \\ \frac{19.07}{34.84s + 1} e^{-0.02s} & \frac{-0.0263}{1.287s + 1} e^{-0.3s} & \frac{-0.579}{45.0s + 1} e^{-0.11s} & \frac{0.21}{43.45s + 1} & \frac{1.55}{44.0s + 1} \\ \frac{-0.0671}{s} & \frac{0.0504}{s} & \frac{0.00233}{s} e^{-s} & \frac{-0.000841}{s} & \frac{0.000883}{s} e^{-2s} \end{bmatrix} \quad (5-7)$$

Utilizando escalones de -10% de amplitud con respecto al punto de funcionamiento, se obtiene la matriz (5-8):

$$G_{d-}(s) = \begin{bmatrix} \frac{13.133}{2.785s + 1} e^{-0.3s} & \frac{-16.65}{5.706s + 1} e^{-0.3s} & \frac{4.343}{4.604s + 1} e^{-0.9s} & \frac{0.195}{2.79s + 1} e^{-0.5s} & \frac{-0.416}{45.63s + 1} e^{-0.4s} \\ \frac{9.093}{50.24s + 1} & \frac{150(1 + 0.017s)}{(1000s + 1)} e^{-0.02s} & \frac{-0.7}{56.1s + 1} e^{-0.3s} & \frac{0.198}{41.92s + 1} & \frac{1.53}{43.5s + 1} e^{-0.024s} \\ \frac{-0.0716}{s} & \frac{0.0478}{s} & \frac{0.00250}{s} e^{-s} & \frac{-0.000831}{s} & \frac{0.000911}{s} e^{-2s} \end{bmatrix} \quad (5-8)$$

Finalmente se toma $G_d(s)$ como media aritmética de $G_{d+}(s)$ y $G_{d-}(s)$:

$$G_d(s) = \begin{bmatrix} \frac{18.204}{3.527s + 1} e^{-0.3s} & \frac{-11.69}{4.188s + 1} e^{-0.3s} & \frac{4.4}{4.54s + 1} e^{-0.9s} & \frac{0.203}{2.73s + 1} e^{-0.65s} & \frac{-0.393}{44.0s + 1} e^{-0.35s} \\ \frac{14.08}{42.54s + 1} e^{-0.01s} & \frac{-0.0263}{1.287s + 1} e^{-0.3s} & \frac{-0.64}{50.6s + 1} e^{-0.21s} & \frac{0.204}{42.69s + 1} & \frac{1.54}{43.8s + 1} e^{-0.012s} \\ \frac{-0.0694}{s} & \frac{0.0491}{s} & \frac{0.00242}{s} e^{-s} & \frac{-0.000836}{s} & \frac{0.000897}{s} e^{-2s} \end{bmatrix} \quad (5-9)$$

Las funciones de transferencia anteriores se han calculado mediante la utilidad System Ident Toolbox de Matlab.

A continuación se muestran las comparaciones entre las respuestas de las variables de control del sistema no lineal y de las funciones lineales, así como la bondad de los ajustes, frente a variaciones en las variables de perturbación (a la izquierda frente a cambios del +10% y a la derecha frente a cambios del -10%). El color negro representa la respuesta de las variables de control del sistema no lineal frente a escalones en la perturbación de amplitud $\pm 10\%$. El color azul representa la respuesta de las variables de control del sistema lineal $G_{d+}(s)$ (5-7) frente a escalones en la perturbación de amplitud $\pm 10\%$, el color verde representa la respuesta de las variables de control del sistema lineal $G_{d-}(s)$ (5-4) frente a escalones en la perturbación de amplitud $\pm 10\%$, y el color rojo representa la respuesta de las variables de control del sistema lineal $G_d(s)$ (5-9) frente a escalones en la perturbación de amplitud $\pm 10\%$.

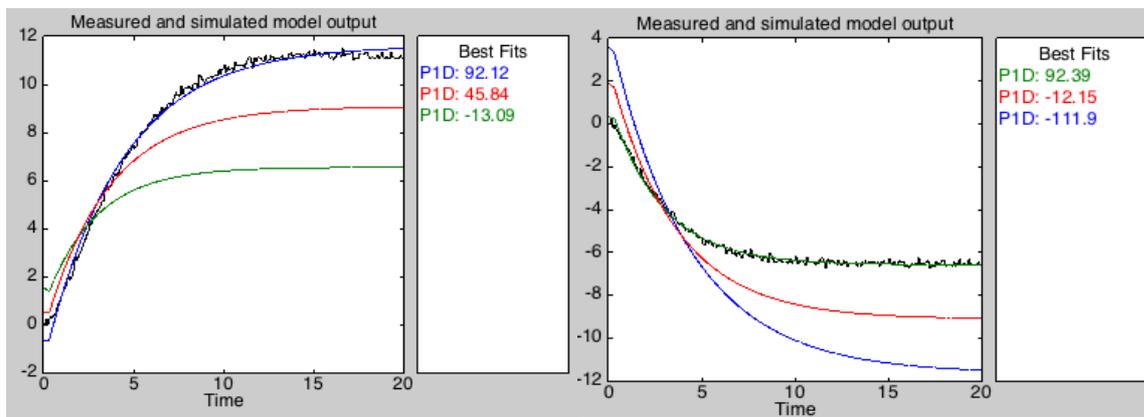


Figura 5.11 Respuesta sistema no lineal-lineal. X_2' frente a escalón amplitud $\pm 10\%$ en P_0 , y ajuste

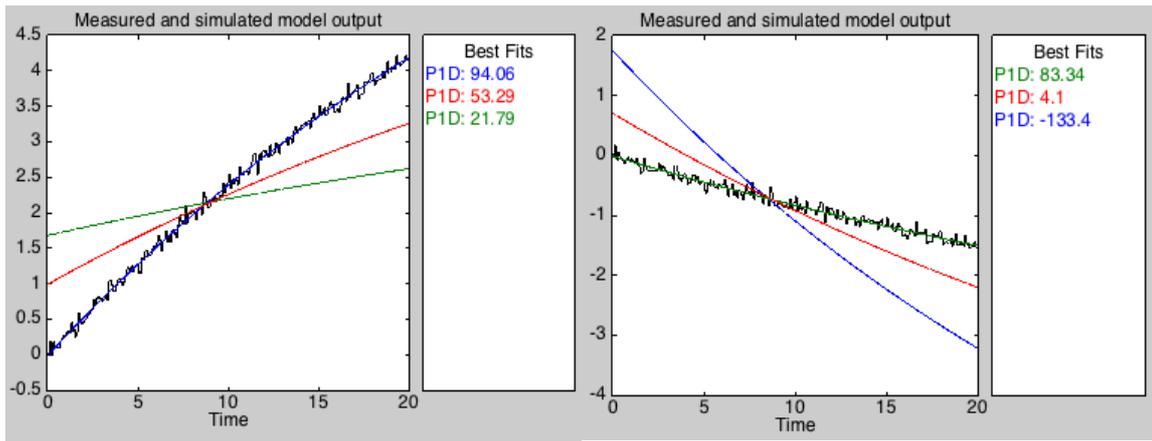


Figura 5.12 Respuesta sistema no lineal-lineal. P_2 frente a escalón amplitud $\pm 10\%$ en P_0 , y ajuste

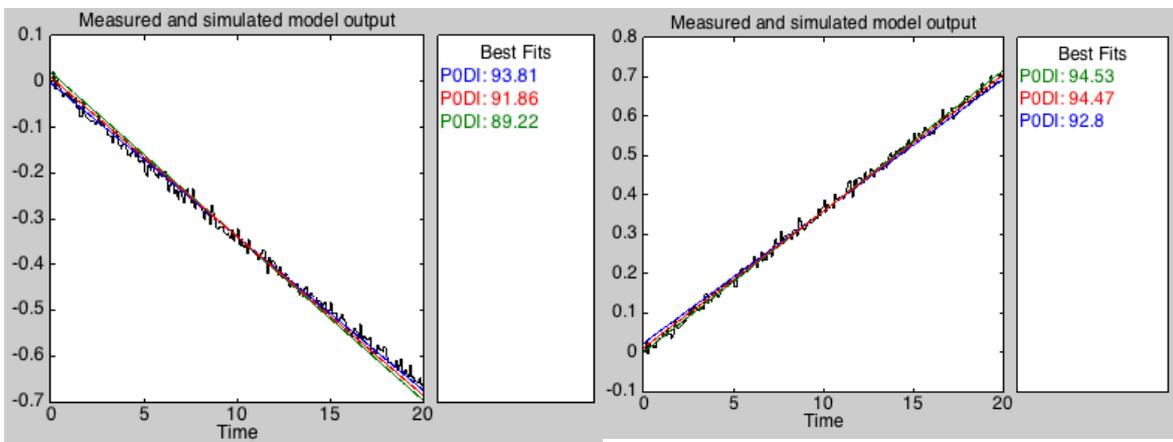


Figura 5.13 Respuesta sistema no lineal-lineal. L_2 frente a escalón amplitud $\pm 10\%$ en P_0 , y ajuste

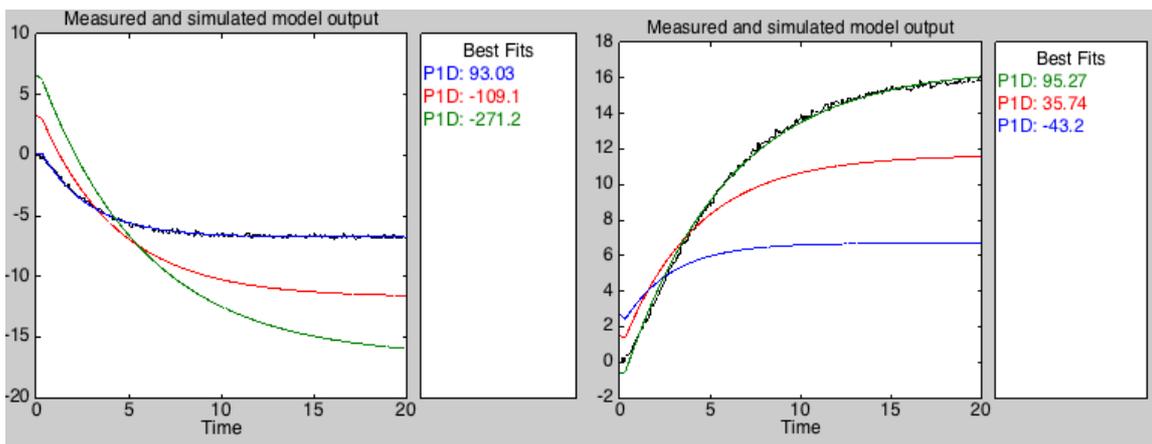


Figura 5.14 Respuesta sistema no lineal-lineal. X_2' frente a escalón amplitud $\pm 10\%$ en F_1 , y ajuste

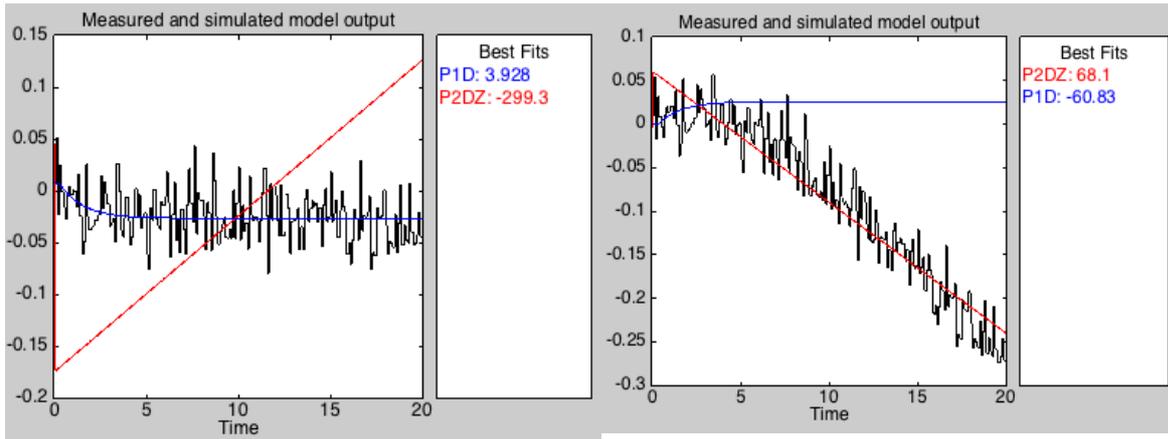


Figura 5.15 Respuesta sistema no lineal-lineal. P_2 frente a escalón amplitud $\pm 10\%$ en F_1 , y ajuste

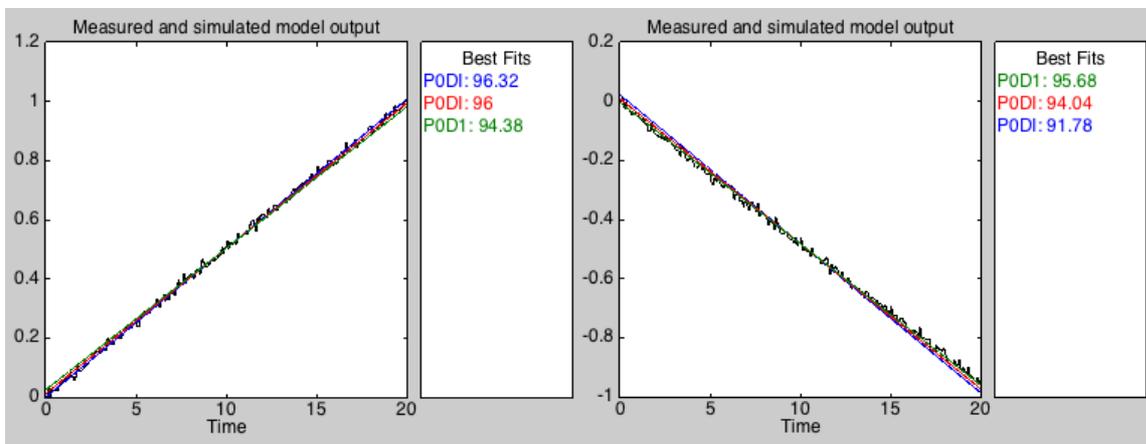


Figura 5.16 Respuesta sistema no lineal-lineal. L_2 frente a escalón amplitud $\pm 10\%$ en F_1 , y ajuste

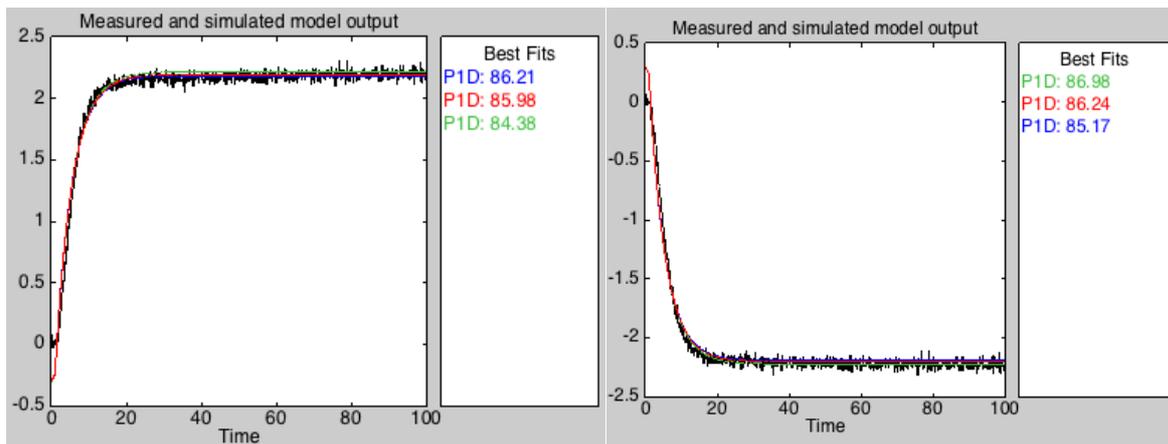


Figura 5.17 Respuesta sistema no lineal-lineal. X_2' frente a escalón amplitud $\pm 10\%$ en X_1 , y ajuste

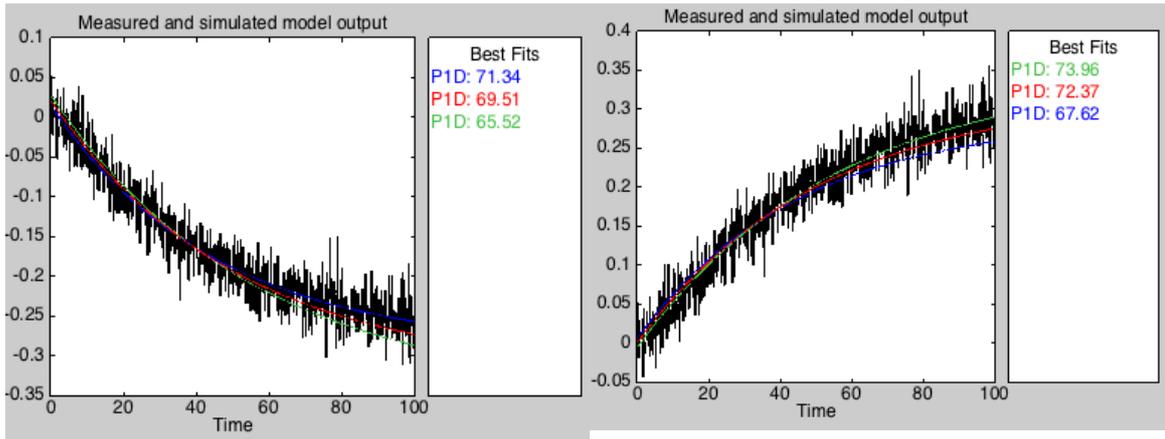


Figura 5.18 Respuesta sistema no lineal-lineal. P_2 frente a escalón amplitud $\pm 10\%$ en X_1 , y ajuste

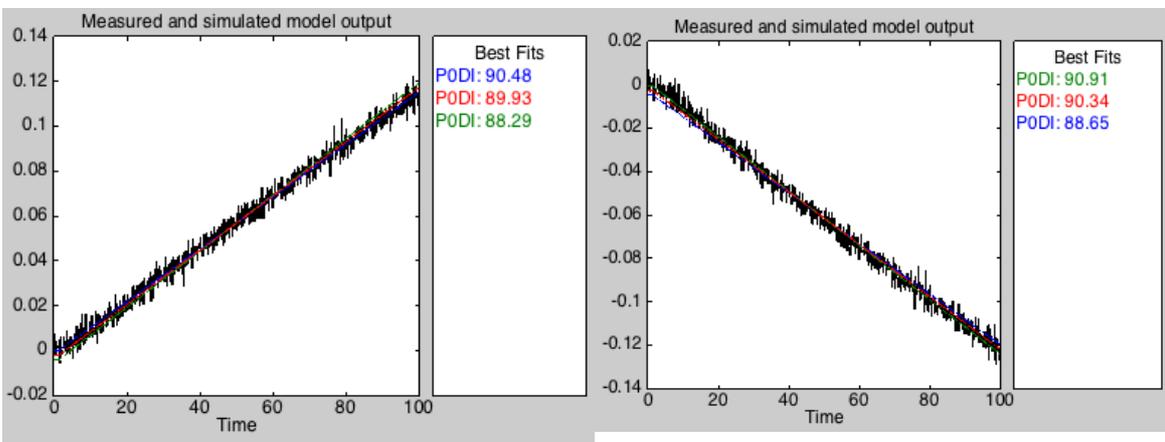


Figura 5.19 Respuesta sistema no lineal-lineal. L_2 frente a escalón amplitud $\pm 10\%$ en X_1 , y ajuste

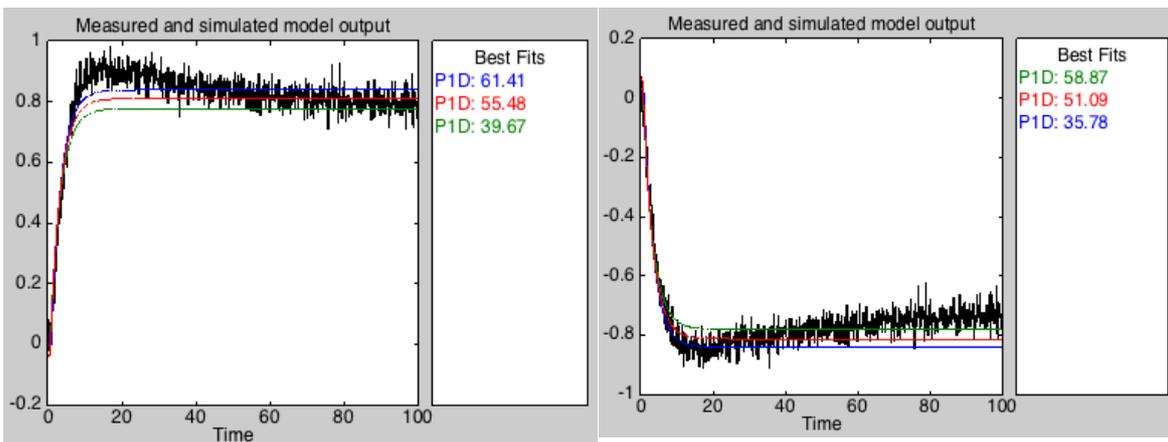


Figura 5.20 Respuesta sistema no lineal-lineal. X_2' frente a escalón amplitud $\pm 10\%$ en T_1 , y ajuste

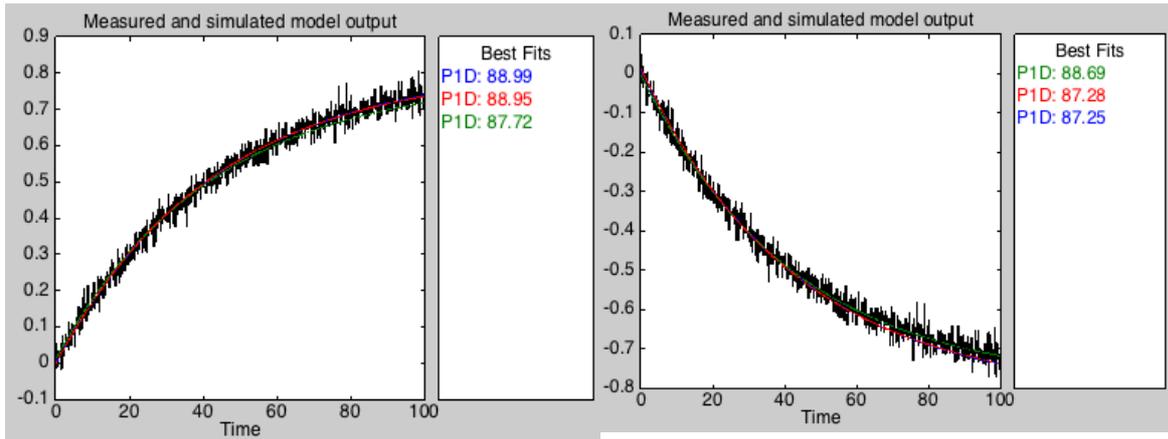


Figura 5.21 Respuesta sistema no lineal-lineal. P_2 frente a escalón amplitud $\pm 10\%$ en T_1 , y ajuste

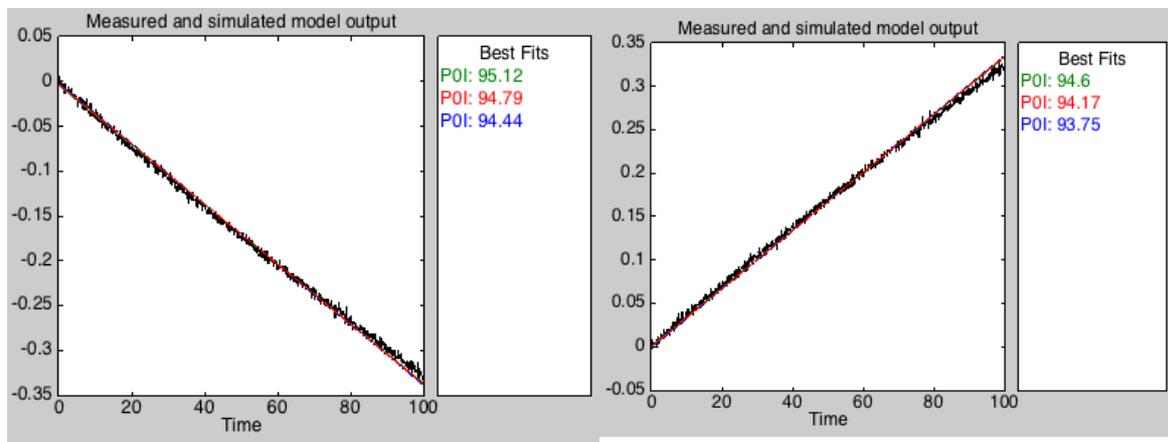


Figura 5.22 Respuesta sistema no lineal-lineal. L_2 frente a escalón amplitud $\pm 10\%$ en T_1 , y ajuste

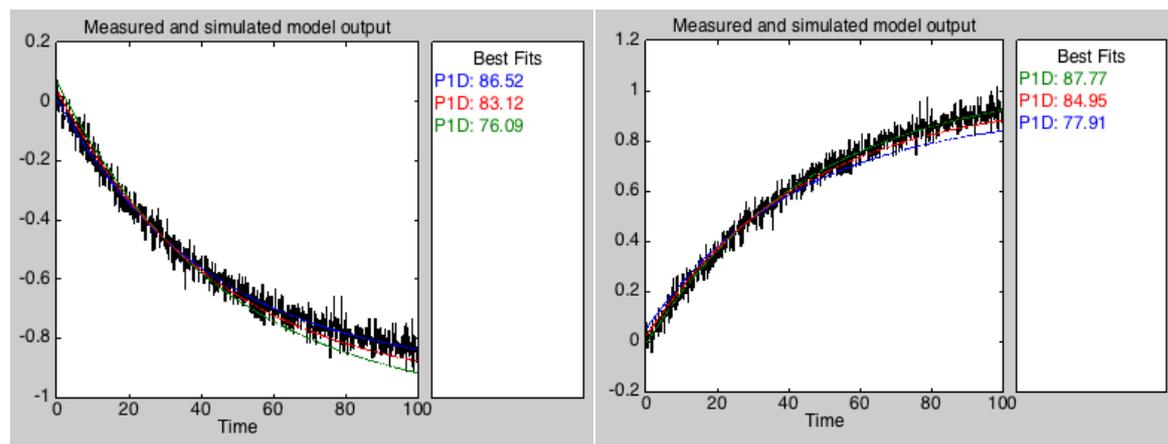


Figura 5.23 Respuesta sistema no lineal-lineal. X_2' frente a escalón amplitud $\pm 10\%$ en T_{200} , y ajuste

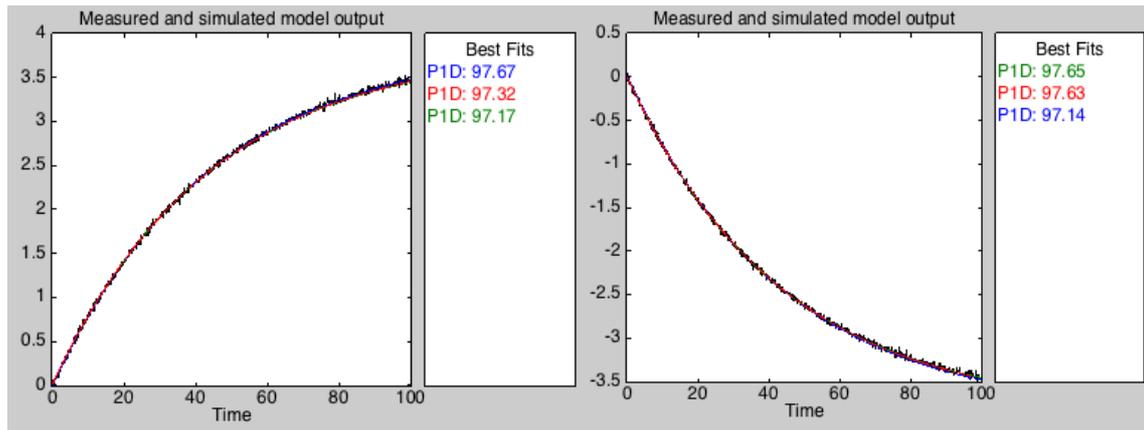


Figura 5.24 Respuesta sistema no lineal-lineal. P_2 frente a escalón amplitud $\pm 10\%$ en T_{200} , y ajuste

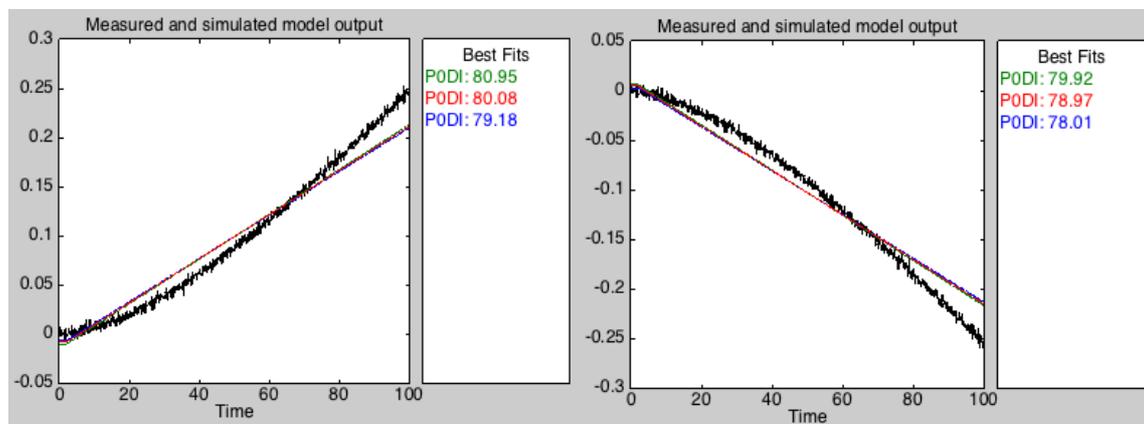


Figura 5.25 Respuesta sistema no lineal-lineal. L_2 frente a escalón amplitud $\pm 10\%$ en T_{200} , y ajuste

5.3 Elección emparejamientos SISO

Los emparejamientos de salida-entradas más favorables para un sistema SISO se pueden analizar con más profundidad usando la matriz DRGA (dynamic relative gain array) [10]. La matriz DRGA es similar a la RGA (matriz de ganancias relativas estáticas), con la particularidad de que para su cálculo se utilizan las funciones de transferencia completas, no solo las ganancias estáticas. El análisis de esta matriz permite interpretar qué emparejamientos son más favorables para realizar un control monovariable, información muy útil sobre las interacciones entre variables del sistema. Con estos datos se pueden escoger las parejas entrada-salida que más interactúan entre sí y menos con las demás.

Se define DRGA:

$$DRGA(G(s)) = \Lambda(s) \triangleq G(s) \times (G(s)^{-1})^T \quad (5-10)$$

Siendo $G(s)$ la función de transferencia de la variable de control frente a la variable manipulable del sistema. 'x' representa el producto elemento por elemento de las dos matrices. Cada elemento ' λ_{ij} ' de la DRGA representa:

$$\lambda_{ij} = \lim_{s \rightarrow 0} \left(\frac{G_{ij}(s)}{G_{ij}(s)} \right)_{\text{todos los lazos abiertos}} \quad (5-11)$$

Aunque la ecuación (5-10) es correcta, implica cálculo matricial cuyos elementos son funciones de Laplace. Operar este sistema matricial es complicado e induce a error incluso para programas informáticos. Sin embargo, una simplificación del cálculo fue introducida en el artículo publicado en 2010 por Wuhua Hu, Wen-Jian Cai y Gaoxi Xiao. [12].

Dicha simplificación consiste en cambiar la matriz $G(s)$ por la matriz de ganancias $G(0)$, pero tanto los integradores y diferenciadores puros que hubiere se sustituyen por la variable “ ε ”, que tiene un valor pequeño (típicamente un par de ordenes inferior a la ganancia más baja de $G(s)$). El resultado es el mismo que la RGA, pero con la ventaja de que se pueden considerar los integradores y diferenciadores en el cálculo, aspecto que no se podía desarrollar anteriormente. Aplicando al caso anterior:

$$G(0, \varepsilon) = \begin{bmatrix} 171 & 159.6 & 0 \\ 0.02 & -0.07923 & 0 \\ -0.6875 & -5.84 \cdot 10^{-5} & -0.0500 \\ \frac{\quad}{\varepsilon} & \frac{\quad}{\varepsilon} & \frac{\quad}{\varepsilon} \end{bmatrix} \quad (5-12)$$

$$\Lambda = RGA \left(\begin{bmatrix} 171 & 159.6 & 0 \\ 0.02 & -0.07923 & 0 \\ -0.6875 & -5.84 \cdot 10^{-5} & -0.0500 \\ \frac{\quad}{\varepsilon} & \frac{\quad}{\varepsilon} & \frac{\quad}{\varepsilon} \end{bmatrix} \right)_{\varepsilon \rightarrow 1 \cdot 10^{-7}} =$$

$$\begin{bmatrix} 171 & 159.6 & 0 \\ 0.02 & -0.07923 & 0 \\ -0.6875 & -5.84 \cdot 10^{-5} & -0.0500 \\ \frac{\quad}{\varepsilon} & \frac{\quad}{\varepsilon} & \frac{\quad}{\varepsilon} \end{bmatrix} X \left(\begin{bmatrix} 171 & 159.6 & 0 \\ 0.02 & -0.07923 & 0 \\ -0.6875 & -5.84 \cdot 10^{-5} & -0.0500 \\ \frac{\quad}{\varepsilon} & \frac{\quad}{\varepsilon} & \frac{\quad}{\varepsilon} \end{bmatrix}^{-1} \right)^T =$$

$$\begin{bmatrix} 171 & 159.6 & 0 \\ 0.02 & -0.07923 & 0 \\ -0.6875 & -5.84 \cdot 10^{-5} & -0.0500 \\ \frac{\quad}{\varepsilon} & \frac{\quad}{\varepsilon} & \frac{\quad}{\varepsilon} \end{bmatrix} X \begin{bmatrix} 0.004733 & 0.001195 & -0.06508 \\ 9.534 & -10.21 & -131.1 \\ 0 & 0 & -20\varepsilon \end{bmatrix} =$$

$$\begin{matrix} X_{F_{100}} & F_{200} & F'_2 \\ \begin{bmatrix} 0.809 & 0.191 & 0 \\ 0.191 & 0.809 & 0 \\ 0 & 0 & 1 \end{bmatrix} X'_2 \\ & & P_2 \\ & & L_2 \end{matrix}$$

$$(5-13)$$

Sea y_i la salida del sistema y m_j la entrada (conjunto proceso-válvula)[7] :

- $\lambda_{ij} = 1$. Significa que la ganancia estática de y_i respecto de m_j es independiente de si los lazos están abiertos o cerrados. Se trata de un caso in interacción. En este caso, el emparejamiento salida-entrada es el ideal para el lazo de control.
- $\lambda_{ij} = 0$. Este caso no indica si el correspondiente emparejamiento es factible o no. El control depende enteramente de los demás lazos.
- $0 < \lambda_{ij} < 1$. El emparejamiento y_i - m_j interactúa con los demás lazos. Esta interacción es mayor cuanto más se acerque el término a 0.5, y menor cuanto más se acerque a 1.
- $\lambda_{ij} < 0$. El signo de la ganancia estática de y_i respecto de m_j depende de si los lazos están abiertos o cerrados. La ganancia cambia de signo cuando los demás lazos están cerrados.
- $\lambda_{ij} > 1$. La ganancia estática de y_i respecto de m_j es menor con los restantes lazos cerrados que cuando están abiertos.

- $\lambda_{ij} = \infty$. La ganancia estática de y_i respecto de m_j con los otros lazos cerrados es 0.

Atendiendo a dichas propiedades de la DRGA, las parejas más adecuadas para el modelo equivalente SISO son las siguientes, puesto que son las que corresponden a los elementos de la matriz más cercanos a la unidad:

$X'_2 \rightarrow X_{F100}$
$P_2 \rightarrow F_{200}$
$L_2 \rightarrow F'_2$

Por otro lado, cabe destacar que aunque se elijan estos lazos, cualquier variación de las variables X_{F100} y F_{200} afectarán inevitablemente a las salidas P_2 , X'_2 , lo que es significativo de la naturaleza multivariable del sistema.

5.4 Control SISO equivalente con controles por realimentación

El sistema de control que se va a utilizar es la realimentación negativa. Mediante el uso de controladores que actúan sobre las entradas y que toman como referencia el error en la salida, se busca solventar los problemas de seguimiento de la referencia de salida y de rechazo de perturbaciones. El esquema típico de esta solución de control se puede observar en la **Figura 5.26**:

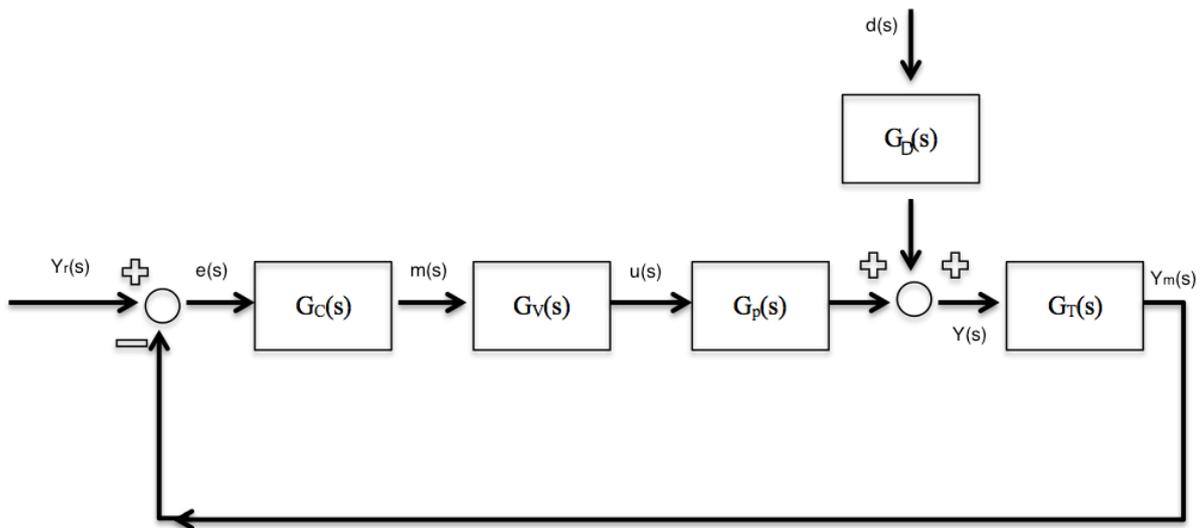


Figura 5.26 Modelo de control realimentado

El sistema realimentado consta de los siguientes elementos:

- $Y_r(s)$: Referencia. Valor de la salida que se quiere obtener.
- $e(s)$: Error medido. Diferencia entre la referencia y la variable controlada medida.
- $G_c(s)$: Controlador. Elemento de control que recibe el error y proporciona la salida para el elemento final de control (en este caso una válvula) para corregir dicho error.
- $m(s)$. Entrada del elemento final de control.
- $G_v(s)$: Función de transferencia de la válvula.
- $u(s)$: Entrada del proceso (salida de la válvula).
- $G_p(s)$: Sistema. Función que modela el comportamiento del sistema físico a controlar.
- $Y(s)$: Salida del sistema.

- $G_T(s)$: Función de transferencia del sensor-transmisor. Elemento electrónico que capta la señal de salida. Al medir la señal introduce típicamente ruido blanco.
- $Y_m(s)$: Salida del sistema medida por el sensor-transmisor.
- $d(s)$: perturbación externa.
- $G_d(s)$: Función de transferencia de la salida $Y(s)$ con respecto a la perturbación $d(s)$.

5.4.1 Adimensionalización

Puesto que la salida del sistema medida por el sensor ofrece un valor porcentual, así como la señal de control que proporciona el controlador, es necesario adimensionalizar las ganancias del proceso. La función de transferencia $G(s)$ definida anteriormente (5-5) no está adimensionalizada. La función de transferencia total del proceso se define como:

$$G_P(s) = G_V(s) \cdot G_P(s) \cdot G_T(s) \quad (5-14)$$

La adimensionalización de la matriz $G_P(s)$ depende de cada elemento, al variar la relación variable de control-variable manipulable, y por tanto, las unidades. A modo de ejemplo, se ilustra a continuación la adimensionalización del emparejamiento $X_{F100}-X_2$:

$$\begin{aligned} \frac{X_2' \%}{X_{F100} \%} &= (G_V \cdot G_P) \frac{\%C}{1} \cdot G_T \frac{\%}{\%C} = 171 \frac{\%C}{1} \cdot \frac{(100 - 0) \%}{(40 - 0) \%C} = 428 \frac{\%}{1} \\ \frac{X_2' \%}{X_{F100} \%} &= \frac{X_2' \%}{X_{F100} \%} \cdot \frac{X_{F100} \%}{X_{F100} \%} = 428 \frac{\%}{1} \cdot \frac{(1 - 0)}{(100 - 0)} \cdot \frac{1}{\%} = 4.28 \frac{\%}{\%} \end{aligned} \quad (5-15)$$

Utilizando los límites de las variables definidos en la **Tabla 1.1**, y siguiendo el mismo proceso anterior para cada elemento de la matriz (5-5), se obtienen los siguientes resultados, correspondientes a la matriz de transferencia del sistema total adimensionalizada:

$$G_P\%(s) = \begin{bmatrix} \frac{4.28}{2.475s + 1} e^{-0.93s} & \frac{0.2}{43.64s + 1} e^{-0.5s} & 0 \\ \frac{1.596}{42.74s + 1} & \frac{-0.317}{43.82s + 1} & 0 \\ \frac{-0.3438}{s} & \frac{-0.0117}{s} & \frac{-0.125}{s} \end{bmatrix} \quad (5-16)$$

Para el diseño de controladores, se pueden usar diversos métodos. En este proyecto, se van a usar controladores por realimentación con acciones PI y PID, sintonizando los parámetros mediante el método SIMC de Skogestad. Para ello, son necesarios los modelos simplificados FOPDT e integrador, antes obtenidos. La **Tabla 5.1** [7] ilustra los valores de dichos parámetros.

	G(s)	Kc	Ti	Td
PI(1)	$\frac{K}{1 + \tau s} e^{-\theta s}$	$\frac{1}{K} \frac{\tau}{\tau_c + \theta}$	$\min(\tau, 4(\tau_c + \theta))$	0
PID(2)	$\frac{K}{1 + \tau s} e^{-\theta s}$	$\frac{1}{K} \frac{2\tau + \theta}{2\tau_c + \theta}$	$\tau + \frac{\theta}{2}$	$\frac{\tau\theta}{2\tau + \theta}$
PI	$\frac{K'}{s} e^{-\theta s}$	$\frac{1}{K'} \frac{2\tau_c + \theta}{(\tau_c + \theta)^2}$	$2\tau_c + \theta$	0
PID	$\frac{K'}{s} e^{-\theta s}$	$\frac{1}{K'} \frac{2\tau_c + \theta}{(\tau_c + 0.5\theta)^2}$	$2\tau_c + \theta$	$\frac{0.25\theta^2 + \tau_c\theta}{2\tau_c + \theta}$

Tabla 5.1 Controladores SIMC Skogestad

(1): Parámetros PI según SIMC Skogestad

(2): Parámetros PID en paralelo según IMC Skogestad

(3): τ_c es un parámetro de diseño. Cuanto menor sea, mayor será la rapidez del sistema pero mayor su sobreoscilación. Es un parámetro que representa la robustez. Su valor dependerá de los valores límite que se requieran en la respuesta. Como valor estándar inicial, se toma $\tau_c = \tau/4$ [9].

Los parámetros mostrados en la anterior tabla reflejan datos de un PID en paralelo, que son los que se usan en este proyecto. A continuación se muestran las fórmulas de un PID en paralelo ideal y de un PID en serie ideal:

$$\text{Paralelo: } G_c(s) = K_c \left(1 + \frac{1}{t_i s} + t_d s \right) \quad (5-17)$$

$$\text{Serie: } G_c(s) = K'_c \left(\frac{t'_i s + 1}{t'_i s} \right) (t'_d s + 1)$$

En el PID paralelo, o no interactivo, las acciones integral y derivativa operan en paralelo independientemente una de otra. En el PID en serie sí interaccionan. Por otro lado, en los PI no existe dicha distinción al no haber acción derivativa.

Puesto que una acción derivativa pura es imposible de implementar, las formas físicamente realizables de los controladores PID son las siguientes:

$$\text{Paralelo: } G_c(s) = K_c \left(1 + \frac{1}{t_i s} + \frac{t_d s}{N t_d s + 1} \right) \quad (5-18)$$

$$\text{Serie: } G_c(s) = K'_c \left(\frac{t'_i s + 1}{t'_i s} \right) \left(\frac{t'_d s + 1}{N t'_d s + 1} \right)$$

Siendo N el factor de filtro derivativo.

Para cada pareja variable manipulable-variable de control calculada anteriormente, se va a desarrollar su correspondiente control por realimentación. El procedimiento de diseño es el que sigue:

- 1) Se calcula la función de transferencia lineal adimensional equivalente del emparejamiento entrada-salida (ya calculada).
- 2) Se sintonizan los parámetros del controlador correspondiente según la **Tabla 5.1**. Para comparar resultados y decidir que controlador es más efectivo, se especificarán y compararán diferentes tipos de controladores para cada lazo de control.
- 3) Se implementa el sistema de control en Simulink según la **Figura 5.26**. Para ello es necesario tener en cuenta que el controlador previamente calculado es lineal y frente al error produce una salida

lineal. Puesto que la simulación se lleva a cabo en el sistema real no lineal, es necesario añadir el punto nominal a la entrada ofrecida por el controlador. Para hacer esto se puede sumar dicho punto nominal a la entrada lineal o directamente establecer la condición inicial del integrador del controlador en el valor nominal.

- 4) Una vez creado el esquema completo controlado no lineal en Simulink, se observa el comportamiento de la salida frente a diferentes entradas. En función de los resultados que se obtengan, se termina de ajustar los parámetros del controlador manualmente.

Para el diseño control de cada emparejamiento variable controlada-variable manipulable se utilizan lazos de realimentación para la pareja seleccionada, mientras que para las otras parejas se dejan los lazos abiertos.

5.4.2 Control realimentado emparejamiento X_2' - X_{F100}

$$G_P\%(s) = \left(\frac{X_2'\%(s)}{X_{F100}\%(s)} \right) = \frac{4.275}{2.475s + 1} e^{-0.93s} \quad (5-19)$$

Aplicando las relaciones descritas en la **Tabla 5.1**, se calculan los parámetros de los controladores PI y PID:

$$PI \left\{ \begin{array}{l} k_c = \frac{1}{4.275} \frac{2.475}{2.475/4 + 0.93} = \mathbf{0.37} \\ t_i = \min \left(2.475; 4 \cdot \left(\frac{2.475}{4} + 0.93 \right) \right) = \mathbf{2.48} \end{array} \right. \quad (5-20)$$

$$PID \left\{ \begin{array}{l} k_c = \frac{1}{4.275} \left(\frac{2 \cdot 2.475 + 0.93}{2 \cdot \frac{2.475}{4} + 0.93} \right) = \mathbf{0.63} \\ t_i = 2.475 + \frac{0.93}{2} = \mathbf{2.94} \\ t_d = \frac{2.475 \cdot 0.93}{2 \cdot 2.475 + 0.93} = \mathbf{0.39} \end{array} \right. \quad (5-21)$$

Estos controladores son un ejemplo de acción inversa:

- Acción inversa: Frente a un aumento de la variable de control, el controlador reacciona proporcionando un menor valor de la variable manipulada, y viceversa. En este caso, si aumenta la concentración de producto, la acción del controlador se traducirá en proporcionar un menor valor de abertura de la válvula, y por tanto un menor caudal de vapor de entrada en el intercambiador. La acción inversa ocurre en controladores con ganancias positivas.
- Acción directa: Frente a un aumento de la variable de control, el controlador reacciona proporcionando un mayor valor de la variable manipulada, y viceversa. La acción directa ocurre en controladores con ganancias negativas.

La **Figura 5.27** muestra el modelo del sistema en bucle cerrado implementado en Simulink. Como se puede apreciar, el lazo de control se diseña con los demás lazos de control abiertos, es decir, con los emparejamientos P_2 - F_{200} y L_2 - F_2 sin controlar.

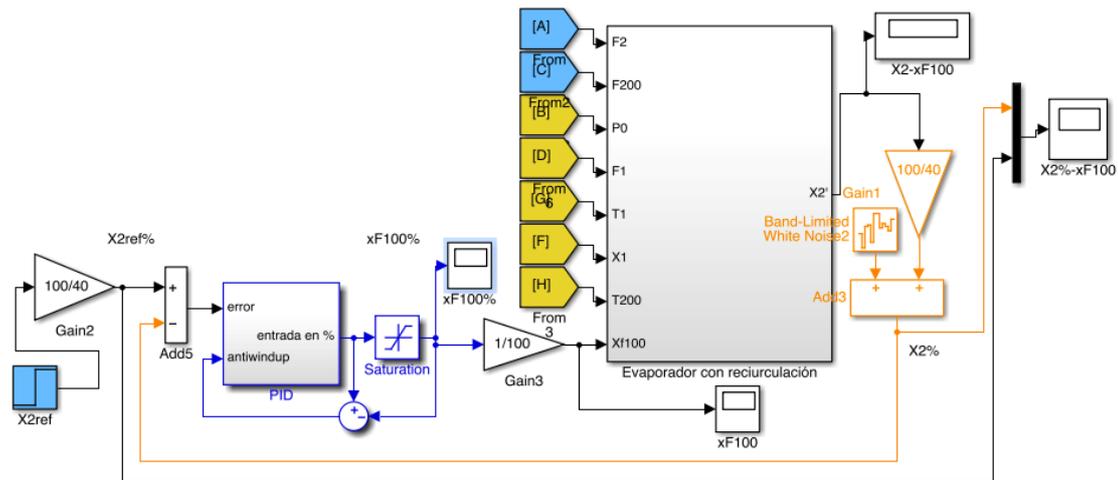


Figura 5.27 Diagrama de bloques Simulink X2-xF100

Entre los elementos que se representan en este diagrama cabe destacar:

- En naranja, aparece el sensor-transmisor que transforma la salida X_2' a $X_2\%$ y añade ruido blanco a la medida.

El ruido blanco es una señal aleatoria cuyos valores a lo largo del tiempo no presentan ninguna correlación estadística. Este ruido aparece en los sensores debido a la electrónica de los mismos. En concreto, se debe al movimiento aleatorio de los electrones en el interior de un conductor eléctrico.

- En el modelo hay diversas ganancias que ajustan los valores absolutos a valores porcentuales. Por ejemplo, “Gain2” ajusta la referencia a valor porcentual, y “Gain3, Gain1” ajustan, por un lado, la entrada porcentual a entrada absoluta, y por otro, la salida absoluta a salida porcentual (la función de proceso “Evaporador con recirculación” trabaja con valores absolutos). Estos ajustes se pueden observar en el cálculo realizado en (5–15).
- En azul, aparecen el controlador y la saturación. Puesto que las limitaciones físicas de una válvula son estar completamente cerrada o abierta (0%-100%), no se pueden proporcionar entradas que excedan estos valores. La función del “bloque saturación” es limitar los valores que exceden de dichos límites.

Entrando en detalle en el bloque del controlador, se puede observar lo siguiente:

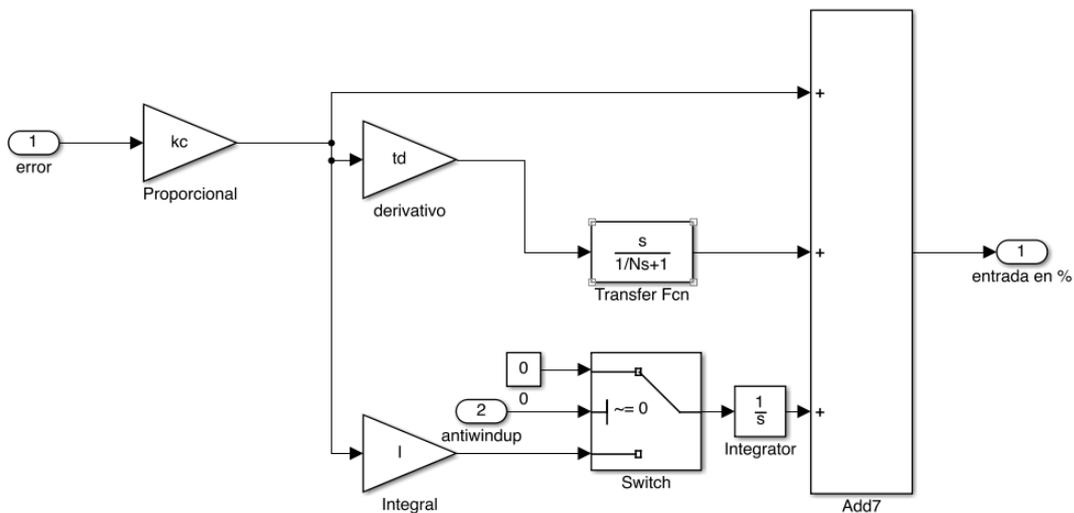


Figura 5.28 Esquema Simulink PID

La ecuación de un controlador PID ideal en paralelo es (5-17):

Sin embargo, como se puede observar, en el modelo implementado dos modificaciones importantes:

- Se incluye un filtro en la acción derivativa (factor $N=10$) para hacer un PID realizable, ya que una acción derivativa pura no se puede implementar en la realidad, pues correspondería a una función de transferencia con el grado del numerador mayor que el del denominador.

Como consecuencia del filtro, el efecto de la acción derivativa es algo menor, pero se evita la amplificación del ruido. Una acción derivativa pura amplifica inevitablemente el ruido, dando lugar a respuestas con sobreoscilaciones inesperadas, e incluso la inestabilidad.

- Se incluye un interruptor que desactiva o activa la saturación de la acción integral; es el llamado “anti-windup”.

Si la variable de entrada de proceso permanece en saturación, la acción integral acumula dicho error, que luego tiene que “deshacer” hasta alcanzar un nuevo punto de entrada. La acción “anti-windup” permite desactivar la acción integral en caso de saturación de la entrada, para que de esta manera no se almacene el error, mejorando la velocidad de respuesta.

La **Figura 5.29** muestra la respuesta del sistema con los controladores PI y PID antes diseñados, frente a un cambio inicial en la referencia X_2' del -5% y, posteriormente, a los 10 minutos un cambio del +5% con respecto al punto nominal.

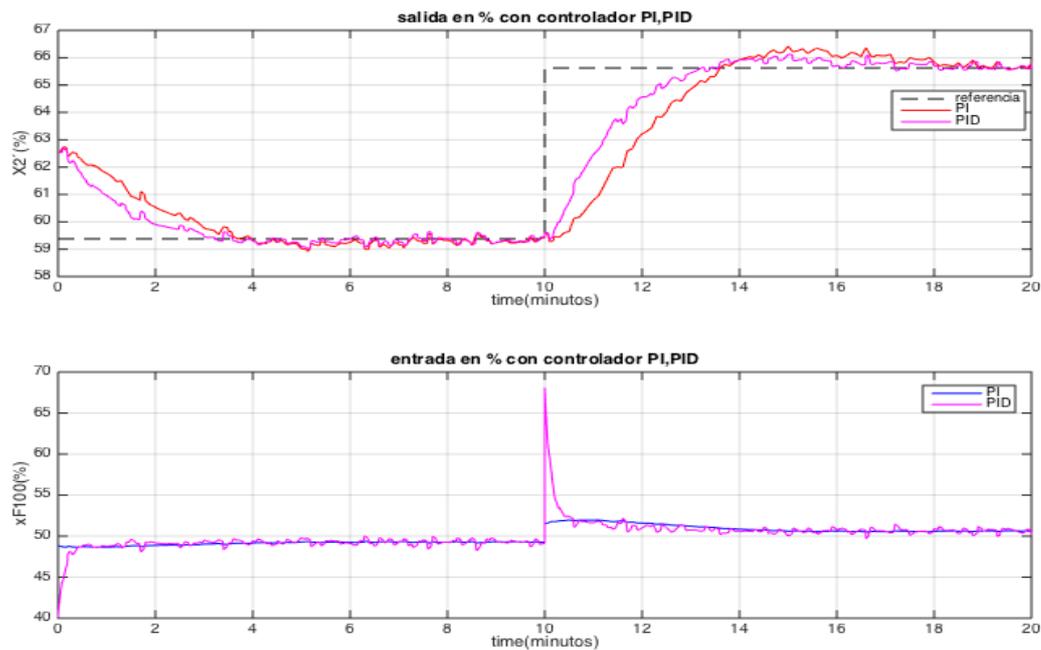


Figura 5.29 Respuesta de X_2' % y x_{F100} % frente a cambios en la referencia para PI y PID

Resultados:

- Ambos controladores ajustan la salida a la referencia con rapidez y con poca sobreoscilación (debido a el valor de τ_c de diseño). Sin embargo, la respuesta que ofrece el PID es mejor, pues permite aumentar la rapidez del sistema mediante K_c y contrarrestar la sobreoscilación que se genera con dicho aumento mediante el tiempo derivativo (t_d).
- Es también notorio la evolución de la variable manipulada. En el caso del PID los cambios son más rápidos y bruscos (pico derivativo), pero se mantiene mucho más tiempo en régimen estacionario fijo en un punto. De cara a la vida útil de la válvula, el hecho de que se esté variando la apertura durante menos tiempo es un aspecto positivo (PID alcanza más rápidamente el régimen estacionario), aunque un pico como el que produce el PID al abrir o cerrar la válvula puede inducir otros problemas por el brusco incremento de caudal. La elección depende de los requerimientos de control que se necesiten en el caso real. Para este proyecto y de cara a la simulación, la respuesta del PID es mejor pues aumenta la velocidad de reacción con menor sobreoscilación.
- Por otro lado, cabe destacar el por qué de los cambios instantáneos en la válvula. En el modelo dinámico se ha despreciado la dinámica de apertura-cierre de la válvula puesto que la constante de tiempo de ésta es del orden de segundos, frente a las constantes de tiempo del orden de minutos que presentan las variables de salida. Por ello, aunque realmente el cambio no sea instantáneo, al observar la respuesta de las salidas en el orden temporal de los minutos, es válida la simplificación.

Finalmente, tomando el PID como el controlador idóneo para este emparejamiento, se realiza un ajuste manual de los parámetros para mejorar la respuesta y se obtiene:

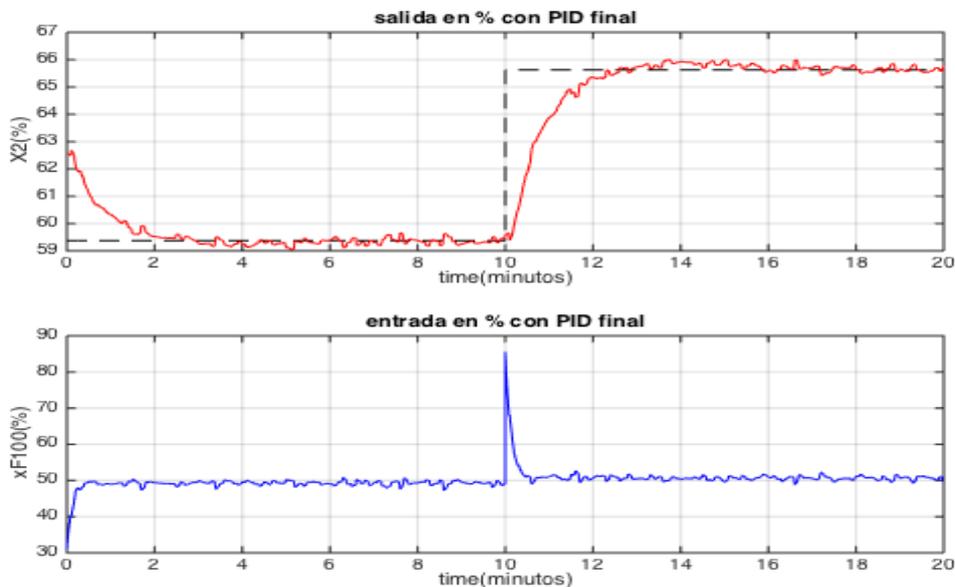


Figura 5.30 Respuesta de X_2' % y x_{F100} % frente a cambios en la referencia para PID final

$k_c = 1 \text{ \%}X_2' / \text{\%}x_{F100}$ $t_i = 3 \text{ min}$ $t_d = 0.5 \text{ min}$	S. O. $\sim 0\%$ $t_s = 1.65 \text{ min}$ $t_{e5\%} = 1.3 \text{ min}$
--	--

- S.O. Sobreoscilación: Representa en porcentaje la oscilación máxima de la curva. Viene dado por:

$$S. O. (\%) = \frac{y_{\text{pico}} - y_{\text{régimen permanente}}}{y_{\text{régimen permanente}} - y_{\text{nominal}}} \quad (5-22)$$

- T_s . Tiempo de subida. Es el tiempo que tarda la salida en alcanzar por primera vez el valor de la salida en régimen permanente.
- $Te_{5\%}$. Tiempo de establecimiento al 5%. Es el tiempo que tarda la salida en alcanzar el régimen estacionario (con una desviación del 5% con respecto al régimen permanente).

A continuación se muestra la respuesta de X_2' frente a una cadena de perturbaciones, manteniendo la referencia en el punto nominal. En concreto, en el minuto 20 se introduce un escalón de amplitud -10% en T_1 y en el minuto 60 un escalón de amplitud +10% en la perturbación P_0 .

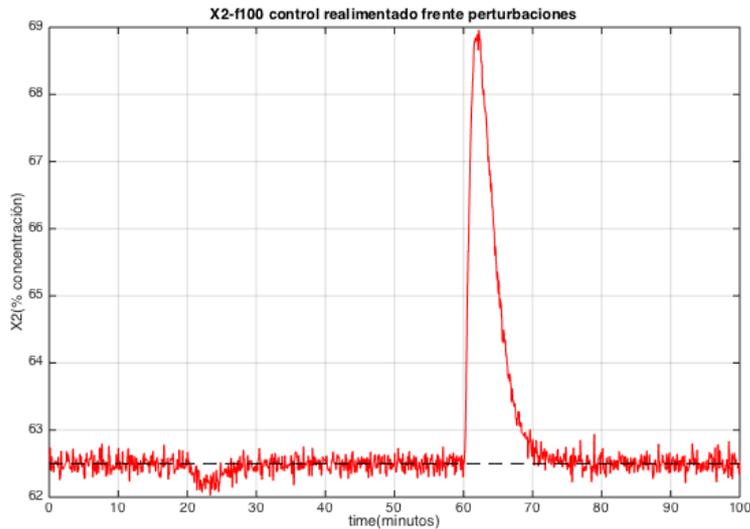


Figura 5.31 Control realimentado X_2' - X_{F100} frente a perturbaciones

Como se puede observar, la perturbación P_0 afecta mucho más a la sobreoscilación de X_2' que la perturbación T_1 . Esto cobra sentido físicamente pues una variación en la presión de la caldera provocará un cambio no deseado en el caudal de vapor de entrada que el controlador intentará corregir. Aunque en el régimen permanente la perturbación se termina rechazando al completo, la sobreoscilación provocada puede suponer un problema para los requerimientos de concentración de producto. Para ello, más adelante se introducirán otras técnicas para rechazar dichas perturbaciones con más efectividad.

Para verificar la importancia del filtro derivativo, en la **Figura 5.32** se ilustra el tipo de respuesta que se obtiene en la salida del sistema en caso de factor derivativo $N=0.1$ y $N=20$ frente a la misma referencia anterior, donde se puede observar la amplificación del ruido.

La respuesta que se obtiene muestra el efecto negativo sobre la variable manipulable de un factor derivativo bajo, pues existen muchos picos de difícil seguimiento para la válvula real. Por otro lado, la salida X_2' tiene efectos de amplificación de ruido, indeseables en todo caso.

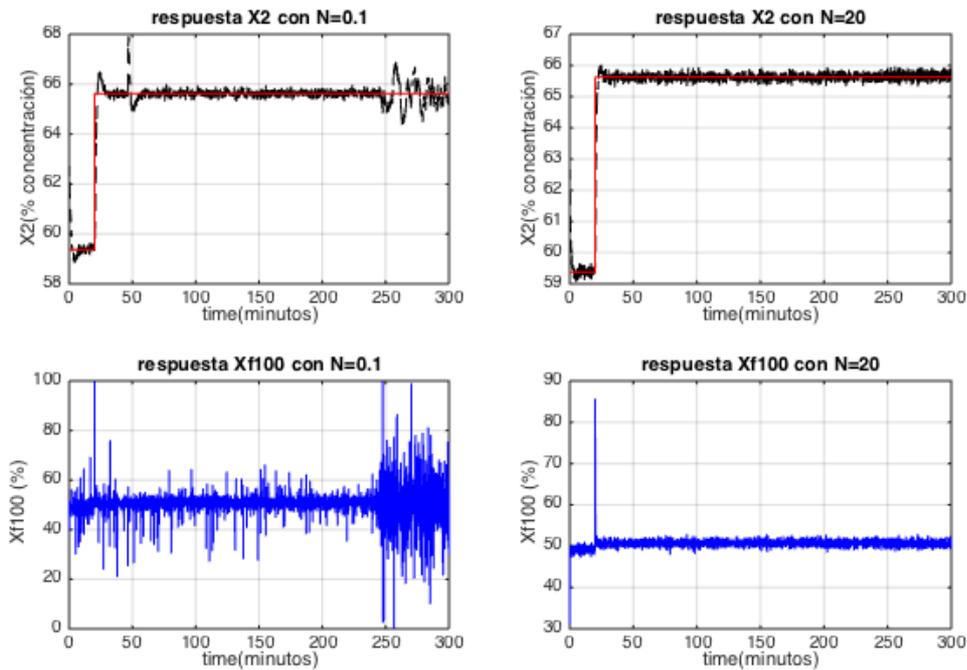


Figura 5.32 Respuesta de $X_2\%$ y $x_{F100}\%$ frente a cambios en la referencia con diferentes filtros PID

El código utilizado en Matlab para presentar las gráficas es el que aparece a continuación. El resto de códigos están recogidos en el Anexo A.

Código 3.1

```
figure
%a= introducir aquí la variable de Scope Simulink de la salida
%b= introducir aquí la variable de Scope Simulink de la entrada
ax1 = subplot(2,1,1); % top subplot
ax2 = subplot(2,1,2); % bottom subplot
plot(ax1,...
    a(:,1),a(:,3), 'black--',...
    a(:,1),a(:,2), 'r')
grid(ax1, 'on')
title(ax1, 'salida en % con controlador PID')
ylabel(ax1, 'X2(%)')
xlabel(ax1, 'time (minutos)')
plot(ax2,...
    b(:,1),b(:,2), 'b')
grid(ax2, 'on')
title(ax2, 'entrada en % con controlador PID')
ylabel(ax2, 'xF100 (%)')
xlabel(ax2, 'time (minutos)')
```

5.4.3 Control realimentado para el emparejamiento $P_2 - F_{200}$

$$G_P\%(s) = \left(\frac{P_2\%(s)}{F_{200}\%(s)} \right) = \frac{-0.3170}{43.82s + 1} \quad (5-23)$$

Aplicando las relaciones descritas en la **Tabla 5.1**, se calculan los parámetros de los controladores PI y PID ($\tau_c = \tau/4$):

$$\text{PI} \left\{ \begin{array}{l} \mathbf{k_c} = \frac{1}{-0.3170} \frac{43.82}{43.82/4 + 0} = \mathbf{-12.618} \\ \mathbf{t_i} = \min \left(43.82; 4 \cdot \left(\frac{4.823}{4} + 0 \right) \right) = \mathbf{43.82} \end{array} \right. \quad (5-24)$$

$$\text{PID} \left\{ \begin{array}{l} \mathbf{k_c} = \frac{1}{-0.3170} \frac{2 \cdot 43.82 + 0}{2 \cdot \frac{43.82}{4} + 0} = \mathbf{-12.618} \\ \mathbf{t_i} = 43.82 + \frac{0}{2} = \mathbf{43.82} \\ \mathbf{t_d} = \frac{43.82 \cdot 0}{2 \cdot 43.82 + 0} = \mathbf{0} \end{array} \right. \quad (5-25)$$

Destaca que tanto el PI como el PID sean iguales según el diseño SIMC, debido a que la acción derivativa es nula. Ésto tiene una explicación:

El sistema simplificado no tiene retardo, puesto que la respuesta no lineal original no presenta un crecimiento inicial suave (derivada $\neq 0$). Esto indica por tanto que la respuesta no lineal presenta un comportamiento típico de primer orden, lo que hace innecesario el uso de un PID (segundo orden) para anular los polos. Con el polo único del PI se consigue este objetivo.

Frente a esta situación, la elección es clara a favor del PI, pues la inclusión de la acción derivativa no sólo no mejoraría la respuesta, sino que incluiría problemas ya mencionados como la amplificación de ruido.

El esquema de control es exactamente el mismo que el utilizado en la **Figura 5.27**, sustituyendo los correspondientes parámetros del controlador, y la pareja entrada-salida a controlar, así como las ganancias de ajuste de adimensionalización.

La **Figura 5.33** muestra la respuesta del sistema, con el controlador PI implementado, ante un cambio inicial del -10% en la referencia P_2 y, posteriormente, a los 100 minutos, un cambio del +10% con respecto al punto nominal de funcionamiento.

Este lazo es un claro ejemplo de un controlador de acción directa (véase la ganancia negativa). Frente a un aumento de la P_2 , el controlador reacciona proporcionando un mayor valor F_{200} , y viceversa.

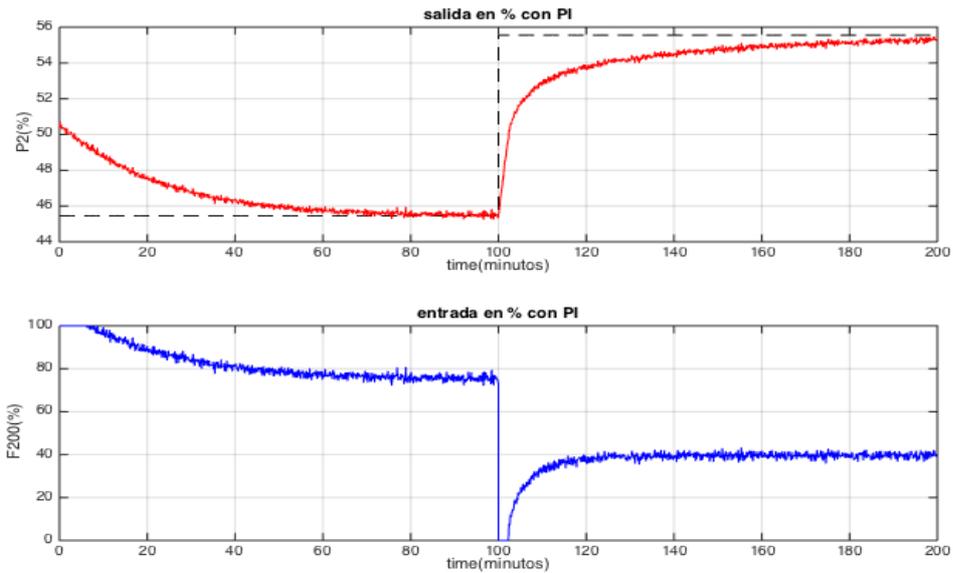


Figura 5.33 Respuesta de $P_2\%$ y $F_{200}\%$ frente a cambios en la referencia con PI

La respuesta es notoriamente más lenta que en el emparejamiento X_2 - x_{F100} , lo cual no es un problema puesto que P_2 no es una variable candidata a sufrir exigentes seguimientos de la referencia. Esto se debe a la diferencia en las constantes de tiempo. Aún así, se puede conseguir una mayor rapidez en el seguimiento de la referencia aumentando K_c , aunque con la siguiente limitación:

- Puesto que el ruido de salida se realimenta (la salida se compara con la referencia y posteriormente se introduce en el controlador), un valor alto de K_c significará un aumento de ruido significativo en la entrada, situación que se quiere evitar o reducir todo lo posible para que no se produzcan continuamente cambios bruscos en la válvula que manipula el caudal F_{200} .

La interacción con el lazo X_2 '- X_{F100} también influye en la rapidez de la respuesta. Dicho problema de interacción no se puede solucionar completamente con las técnicas de control monovariable.

Finalmente, se ajusta la sintonización del controlador manualmente a los siguientes parámetros:

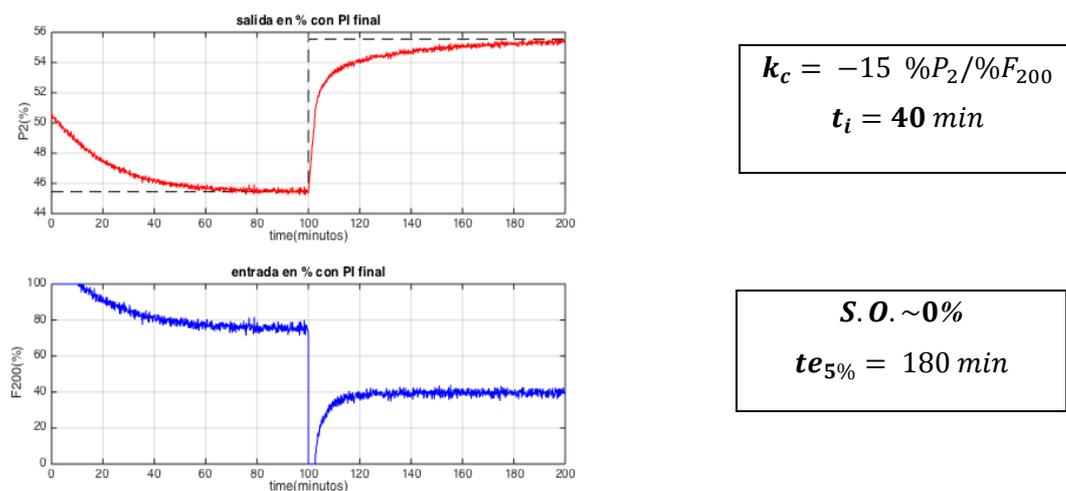


Figura 5.34 Respuesta de $P_2\%$ y $F_{200}\%$ frente a cambios en la referencia con PI final

Este modelo es un buen ejemplo también para observar cómo un aumento desmesurado de la ganancia del controlador afectaría negativamente a la entrada (pese a la mejora de la salida si la válvula física real

pudiese soportar tales variaciones en la entrada), y cómo mejora la salida frente grandes escalones en la referencia con la ayuda del sistema “anti-windup” incorporado [7].

La **Figura 5.35** muestra para $K_c=-100$ dicho efecto en la entrada y la salida para una referencia del 30% con respecto al punto nominal, así como una comparación con el efecto antiwindup desactivado:

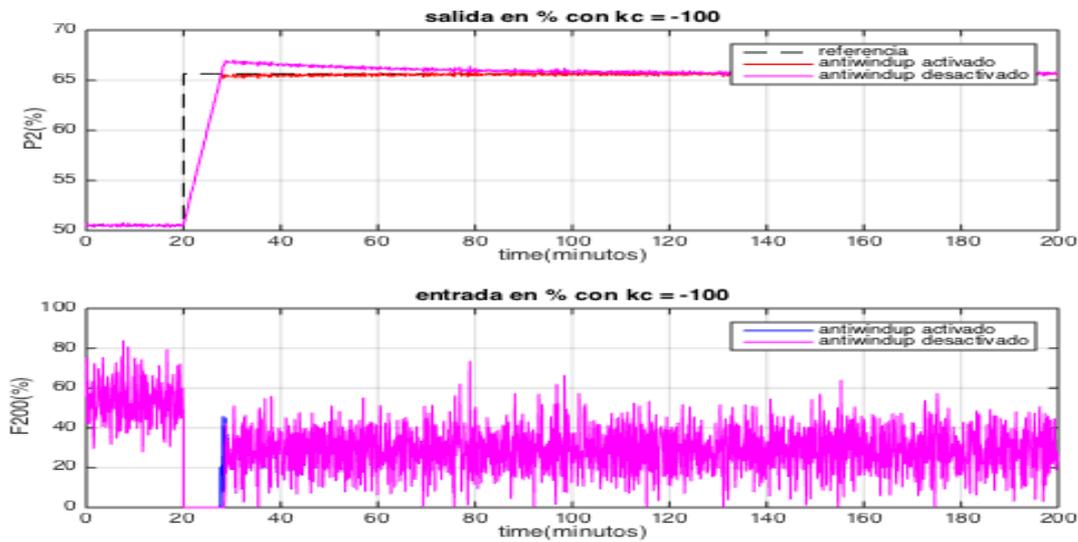


Figura 5.35 Respuesta de $P_2\%$ y $F_{200}\%$ con $k_c = -100$, con y sin antiwindup

Es preferible sin duda un valor de ganancia menor en este caso para que no haya tanto ruido en la entrada, pese a la pérdida de rapidez en el seguimiento de la referencia. Hay que tener en cuenta que P_2 es la variable menos importante de las 3 variables controladas, y no se requiere, por tanto, una excesiva precisión en dicho seguimiento.

A continuación se muestra la respuesta de L_2' frente a una cadena de perturbaciones, manteniendo la referencia en el punto nominal. En concreto, en el minuto 20 se introduce un escalón de amplitud +20% en T_{200} .

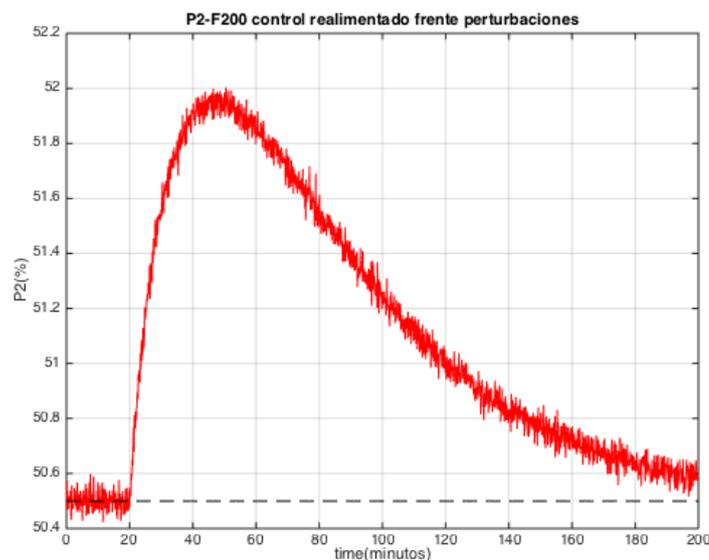


Figura 5.36 Control realimentado P_2 - F_{200} frente a perturbaciones

El rechazo a la perturbación T_{200} es poco eficaz. Aunque en régimen permanente el sistema consigue rechazar al completo la perturbación, la sobreoscilación es considerable y se mantiene durante largo tiempo. Más adelante se mejorará la respuesta de P_2 frente a T_{200} con otras técnicas de control.

5.4.4 Control realimentado emparejamiento $L_2 - F_2'$

$$G_P\%(s) = \left(\frac{L_2\%(s)}{F_2'\%(s)} \right) = \frac{-0.125}{s} \quad (5-26)$$

Aplicando las relaciones descritas para un sistema integrador de la **Tabla 5.1**, se calculan los parámetros de los controladores PI ($\tau_c = 1$):

$$\text{PI} \left\{ \begin{array}{l} \mathbf{k}_c = \frac{1}{-0.125} \cdot \frac{2}{1} = -16 \\ \mathbf{t}_i = 2 \cdot 1 = 2 \end{array} \right. \quad (5-27)$$

Como en el caso anterior, al no haber retardo no es necesario incluir acción derivativa para el correcto funcionamiento del control.

Este caso, puesto que es un sistema integrador, en realidad implementando un controlador que solo tuviese acción proporcional se conseguiría un error en régimen permanente nulo frente a cambios en la referencia:

$$G_{bc} = \frac{G_c \cdot G_P}{1 + G_c \cdot G_P} = \frac{K_c \cdot \frac{K_P}{s}}{1 + \frac{K_c K_P}{s}} = \frac{\frac{K_c K_P}{s}}{\frac{s + K_c K_P}{s}} = \frac{1}{\frac{1}{K_c K_P} s + 1} \rightarrow \text{r. p., } s = 0 \rightarrow G_{bc} = 1 \quad (5-28)$$

Sin embargo, el interés real de control en esta variable no es el seguimiento de una referencia, sino el rechazo de perturbaciones. Puesto que el evaporador presentará problemas graves de funcionamiento si se alcanzan los límites del separador, se mantendrá como referencia fija el nivel del evaporador en $L_2=1$ metro.

En cuanto a las perturbaciones, sí es necesario en este caso un término integrador para que en régimen permanente el error provocado por una perturbación sea igual a 0, y se implementa por tanto un PI con los parámetros obtenidos en la ecuación (5-27). Si hay cambios en una perturbación (F_1 en este caso, comportamiento de tipo integrador) y no en la referencia, la función de transferencia en bucle cerrado queda como sigue:

$$G_{bc} = \frac{G_d}{1 + G_c \cdot G_P} = \frac{G_d}{1 + \frac{T_i s + 1}{T_i s} K_c \cdot \frac{K_P}{s}} = \frac{G_d \cdot T_i s^2}{T_i s^2 + (T_i s + 1) k_c k} \rightarrow \text{r. p., } s = 0 \rightarrow G_{bc} = 0 \quad (5-29)$$

Lo cual indica que frente a un cambio en la perturbación F_1 , con un controlador PI la respuesta L_2 tiene un error en régimen permanente=0.

En la **Figura 5.37** se representa la salida L_2 y la variable manipulada F_2' frente a escalones del +10% y -10% en la perturbación F_1 , manteniendo una referencia L_2 fija en el punto nominal de funcionamiento, para el sistema realimentado con el controlador PI y con el controlador P.

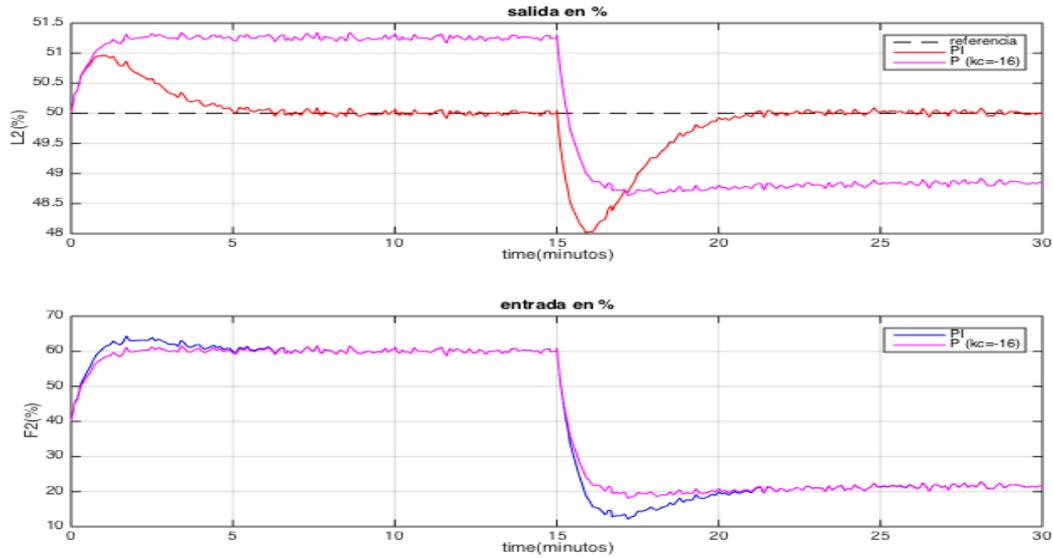


Figura 5.37 Respuesta de $L_2\%$ y $F_2\%$ frente a la perturbación F_1 con controlador PI y P

Queda clara por tanto la elección a favor del PI. La mejor estrategia para aumentar la rapidez de reacción ante la perturbación F_1 es aumentar ligeramente la ganancia del controlador, teniendo en cuenta el hecho de que, como en el caso anterior, el ruido en la entrada se puede amplificar indeseablemente. Teniendo esto en cuenta, se declaran los siguientes parámetros, obteniendo la respuesta de la **Figura 5.38** para el mismo escalón en la perturbación del caso anterior.

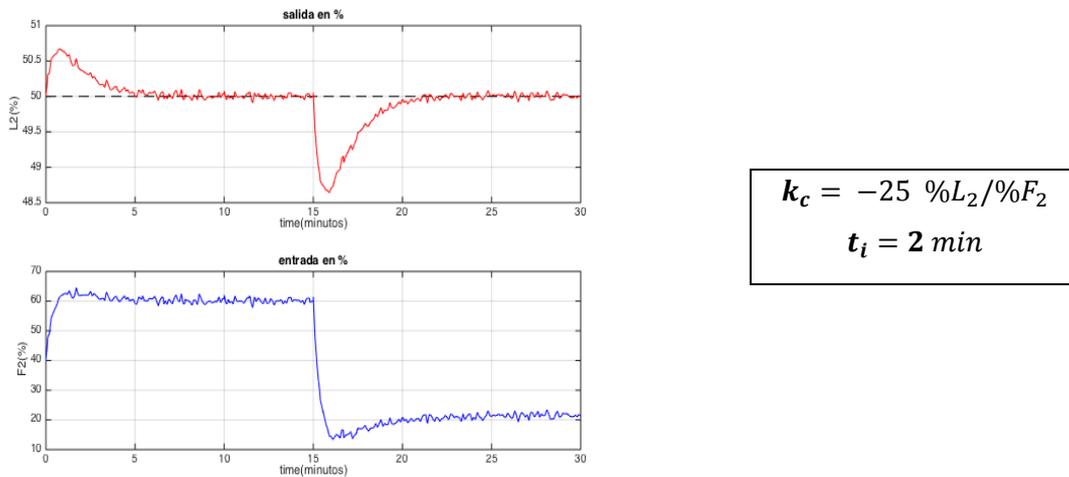


Figura 5.38 Respuesta de $L_2\%$ y $F_2\%$ frente a la perturbación F_1 con controlador PI final

5.5 Control avanzado para rechazo de perturbaciones: acción anticipativa y control en cascada

El control descrito hasta ahora ofrece buenas soluciones frente a cambios en la referencia, así como error nulo en régimen permanente para perturbaciones. Sin embargo, el efecto sobreoscilatorio que pueden provocar dichas perturbaciones puede ser significativo, y es de interés reducirlo lo máximo posible.

Para ello, se va a utilizar en este proyecto técnicas de control anticipativo y control en cascada [7].

Control Anticipativo (feed-forward)

Los sistemas de control por realimentación requieren que exista error en la variable de proceso a controlar para ejercer la acción correctora. Esta forma de actuar implica un cierto retraso en la acción de control y, como consecuencia, una corrección no del todo eficiente frente a perturbaciones externas.

La idea básica del control anticipativo es medir las perturbaciones y actuar sobre el proceso inmediatamente que se produzcan, sin tener que esperar a que afecten a la variable que se está controlando. Para ello se ha de disponer de un modelo de comportamiento del proceso frente a las perturbaciones ($G_d(s)$, calculado anteriormente).

La **Figura 5.39** muestra el diagrama de bloques de la acción anticipativa aplicada a un sistema realimentado.

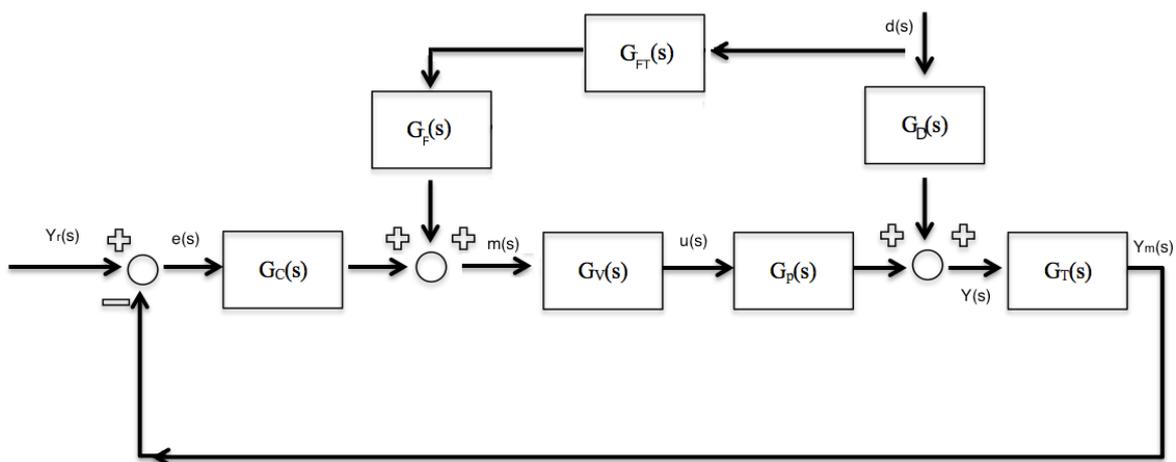


Figura 5.39 Control por acción anticipativa o feedforward

Se definen los siguientes elementos nuevos:

- $G_{FT}(s)$: Función de transferencia del sensor-transmisor que mide la perturbación.
- $G_F(s)$: Función de transferencia del control anticipativo. Idealmente, para que la variación en la variable de control sea nula frente a una perturbación, se debe cumplir:

$$G_D(s) + G_V(s) \cdot G_P(s) \cdot G_F(s) \cdot G_{FT}(s) = 0 \quad (5-30)$$

Por lo tanto, la función de transferencia ideal del control anticipativo es:

$$G_F(s) = -\frac{G_D(s)}{G_V(s)G_P(s)G_{FT}(s)} \quad (5-31)$$

Nótese que en la práctica dicho control no perfecto debido a la imposibilidad de medición de todas las perturbaciones, la inexactitud de la función de transferencia del proceso, errores en las medidas, etc.

Control en Cascada

En muchos procesos es muy sencillo detectar una perturbación antes de que tenga un efecto apreciable sobre la variable de proceso a controlar. Esta detección, que se lleva a cabo mediante la medida de alguna variable interna, permite actuar rápida e intensamente sobre el proceso, evitando desviaciones importantes en la variable a controlar. Esta idea es particularmente útil cuando existen perturbaciones que afectan a la propia variable de proceso que se manipula para controlarlo (perturbaciones a la entrada). [7]

Para estos casos, se utiliza el esquema de control en cascada, representado en la **Figura 5.40**. El control en cascada introduce un control por realimentación secundario (esclavo), que tiene como objetivo corregir la acción del elemento final de control (una válvula en este caso). Este control se produce anterioridad al lazo primario (maestro), frente a una perturbación que afecta directamente a dicho elemento final de control.

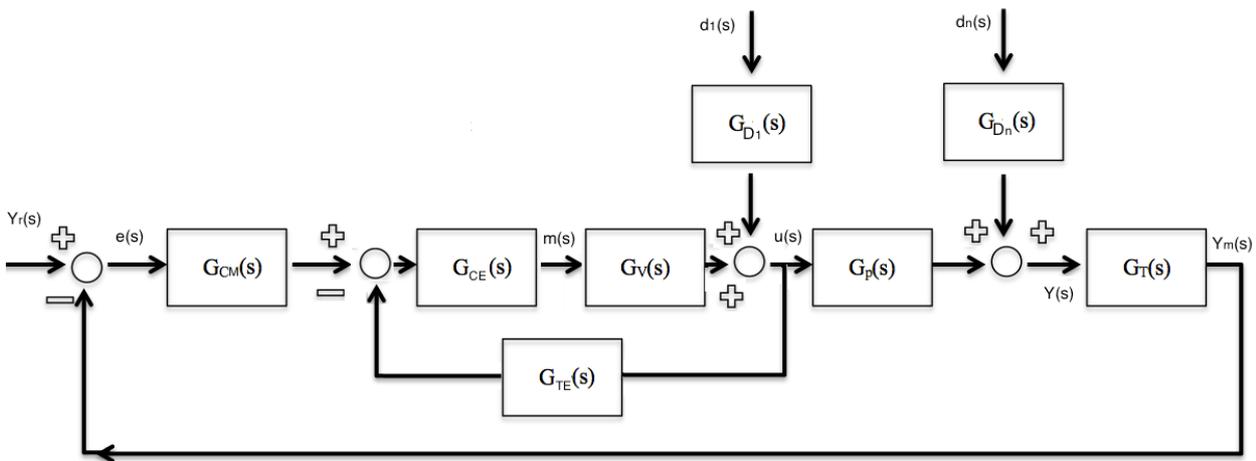


Figura 5.40 Control en cascada

Donde $G_{CM}(s)$, $G_{CE}(s)$ y $G_{TE}(s)$ representan respectivamente las funciones de transferencia del controlador maestro, el controlador esclavo y el sensor transmisor usado para medir la entrada $u(s)$.

Todas las pruebas de simulación desde este punto en adelante se realizan sobre el sistema SISO completo con los 3 bucles de realimentación (ver modelo completo en el Anexo B). De esta manera, esta última parte de diseño de control contempla seguimiento de referencias y rechazo de perturbaciones para las 3 variables de control con sus lazos de realimentación cerrados.

5.5.1 Acción anticipativa para F_1 en el emparejamiento X_2' - X_{F100}

Atendiendo a la **Figura 5.39**, el control anticipativo ofrece la siguiente solución de control:

$$G_F(s) = -\frac{G_D(s)}{G_V(s)G_P(s)G_{FT}(s)} = -\frac{\frac{-11.67}{4.165s + 1} e^{-0.3s} \left(\frac{\%C}{\text{kg/min}} \right)}{\frac{171}{2.475s + 1} e^{-0.93s} \left(\frac{\%C}{1} \right) \cdot \frac{100 - 0}{15 - 7} \left(\frac{\%}{\text{kg/min}} \right)} = \frac{2.311s + 0.934}{712.2s + 171} \frac{1}{\%} \quad (5-32)$$

Nota: el resultado de la división contiene un retardo positivo, eliminado por ser irrealizable. Un retardo positivo implicaría una acción de control que actuase antes de que se produjese la propia perturbación, lo cual no es posible.

los límites de las perturbaciones adoptados se muestran en la **Tabla 6.1**.

Teniendo en cuenta que el controlador ofrece un valor de m(s) porcentual, se añade un último término para adimensionalizar correctamente:

$$G_F\%(s) = \frac{X_{F_{100}}}{F_1} \left(\frac{1}{\%} \right) \cdot \frac{X_{F_{100}}}{X_{F_{100}}} \left(\frac{\%}{1} \right) = \frac{2.311s + 0.934}{712.2s + 171} \cdot \frac{100 - 0}{1 - 0} = \frac{231.1s + 93.4\%}{712.2s + 171\%} \quad (5-33)$$

Al introducir la función G_F en el modelo para $X_2^2 - xF_{100}$ en Simulink, se obtienen las siguientes respuestas para escalones de $\pm 10\%$ con respecto al punto nominal en la perturbación F_1 en los minutos 10 y 30, respectivamente:

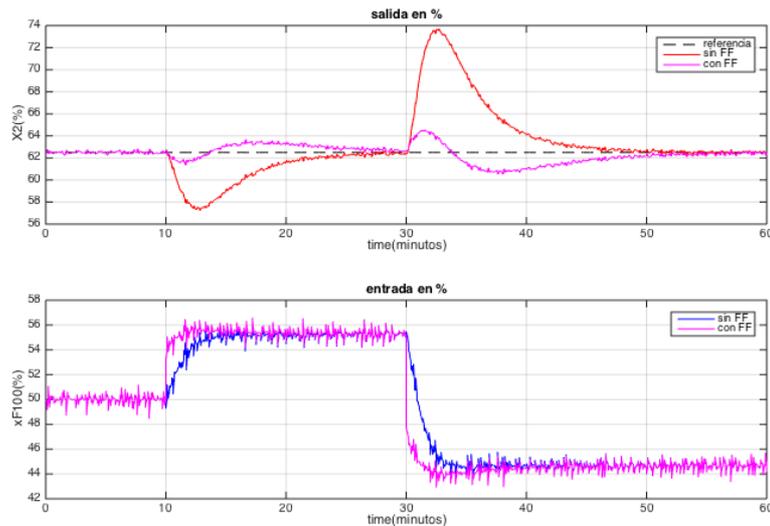


Figura 5.41 respuesta $X_2^2\%$ y $xF_{100}\%$ frente a escalón en F_1 , con y sin control anticipativo

Los resultados con el sistema controlado añadiendo la acción anticipativa de F_1 son significativamente más satisfactorios. Como se puede observar, la sobreoscilación provocada en la variable de control por la variación en F_1 es mucho menor en el sistema con acción anticipativa (valor de pico 64% frente a 74%). Esto se debe al control feed-forward, que modifica la salida que proporciona el controlador nada más se detecta la perturbación (ver minuto 10 y 30). En el modelo sin acción anticipativa, la salida del controlador no se ve afectada hasta que el control por realimentación evalúa la salida con la referencia, lo cual es un proceso mucho más lento.

El control no es perfecto porque variar F_1 implica variar X_2 , que implica variar $X_{F_{100}}$ y, por tanto, afecta también a L_2 (sistema multivariable).

5.5.2 Control prealimentado para T_{200} en el emparejamiento $P_2 - F_{200}$

La perturbación que afecta más claramente a la presión del sistema es la temperatura del agua de refrigeración a la entrada del condensador. Dicha temperatura influye directamente en la cantidad de vapor que se puede condensar y, por tanto, en la presión que existirá en el separador según la cantidad de vapor existente. El controlador anticipativo resulta como sigue:

$$G_F(s) = - \frac{G_D(s)}{G_V(s)G_p(s)G_{FT}(s)} = - \frac{\frac{1.543}{43.78s + 1} \left(\frac{\text{KPa}}{^\circ\text{C}} \right)}{\frac{-0.0792}{43.82s + 1} \left(\frac{\text{KPa}}{\text{kg/min}} \right) \cdot \frac{100 - 0}{37.5 - 12.5} \left(\frac{\%}{^\circ\text{C}} \right)} = \frac{16.91s + 0.3858}{3.469s + 0.0792} \frac{\text{kg/min}}{\%} \quad (5-34)$$

Teniendo en cuenta que el controlador ofrece un valor de m(s) porcentual, se añade un último término para adimensionalizar correctamente:

$$G_F\%(s) = \frac{F_{200}(\text{kg/min})}{T_{200}(\%)}. \frac{F_{200}(\%)}{F_{200}(\text{kg/min})} = \frac{2.311s + 0.934}{712.2s + 171} \cdot \frac{100 - 0}{400 - 0} = \frac{4.228s + 0.0965\%}{3.469s + 0.0792\%} \quad (5-35)$$

En la siguiente figura se puede observar la respuesta de la salida y entrada del bucle realimentado $P_2\%$ - $F_{200}\%$ sin y con prealimentación para una entrada de +10% T_{200} en el minuto 10 y -10% en el minuto 200 con respecto al punto nominal de funcionamiento:

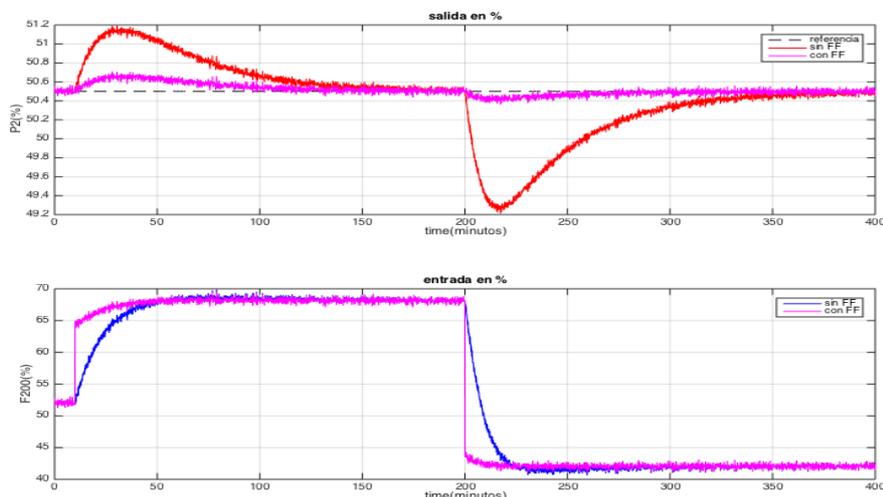


Figura 5.42 Salida $P_2\%$ y entrada $F_{200}\%$ frente a escalón en T_{200} , sin y con control anticipativo

La acción anticipativa suaviza la sobreoscilación considerablemente. Frente a un cambio en la perturbación T_{200} del -10%, la presión P_2 tiene un pico de 49.2 KPa en el control por realimentación, y un pico de tan solo 50.4 KPa en el caso de realimentación con feed-forward. Esto se debe al cambio inmediato en la entrada F_{200} que se produce tan pronto como aparece la perturbación.

La mejora añadida es evidente en este caso. Cabe destacar el tiempo de establecimiento prolongado que dispone el sistema sin feed-forward, que se debe a la constante de tiempo del bucle realimentado P_2 - F_{200} . En este caso no es algo preocupante (pero mejorable), pues P_2 no es una variable que requiera establecerse en régimen permanente rápidamente, como es el caso de X_2 . El interés a la hora de controlar P_2 reside en que no supere los límites de la variable, ya que puede generar problemas en el funcionamiento del sistema, pero el tiempo de establecimiento de la misma no es un problema a tener en cuenta. En todo caso, el tiempo de establecimiento se puede reducir considerablemente aumentando el valor de la ganancia del controlador PI e introduciendo control anticipativo.

5.5.3 Acción anticipativa para F_1 en el emparejamiento $L_2 - F_2'$

La variable de Control L_2 también es afectada considerablemente por la variable F_1 . Al introducir un mayor o menor caudal en el evaporador, el nivel del separador variará en consecuencia. Para evitar que ésta variación sea muy acusada, se introduce control feedforward:

$$G_F(s) = -\frac{G_D(s)}{G_V(s)G_p(s)G_{FT}(s)} = -\frac{\frac{0.05}{s} \left(\frac{\text{m}}{\text{kg/min}} \right)}{\frac{-0.05}{s} \left(\frac{\text{m}}{\text{kg/min}} \right) \cdot \frac{100 - 0}{15 - 7} \left(\frac{\%}{\text{kg/min}} \right)} = 0.08 \frac{\text{kg/min}}{\%} \quad (5-36)$$

Teniendo en cuenta que el controlador ofrece un valor de $m(s)$ porcentual, se añade un último término para adimensionalizar correctamente:

$$G_F\%(s) = \frac{F_2 \left(\frac{\text{kg/min}}{\%} \right)}{F_1 \left(\frac{\text{kg/min}}{\%} \right)} \cdot \frac{F_2 \left(\frac{\%}{\text{kg/min}} \right)}{F_2 \left(\frac{\text{kg/min}}{\%} \right)} = 0.08 \cdot \frac{100 - 0}{5 - 0} = 1.6 \frac{\%}{\%} \quad (5-37)$$

Usando la misma variación de F_1 usada en el caso anterior (variación de $\pm 10\%$ en los minutos 10 y 30), se obtienen los resultados de L_2 mostrados en la siguiente Figura:

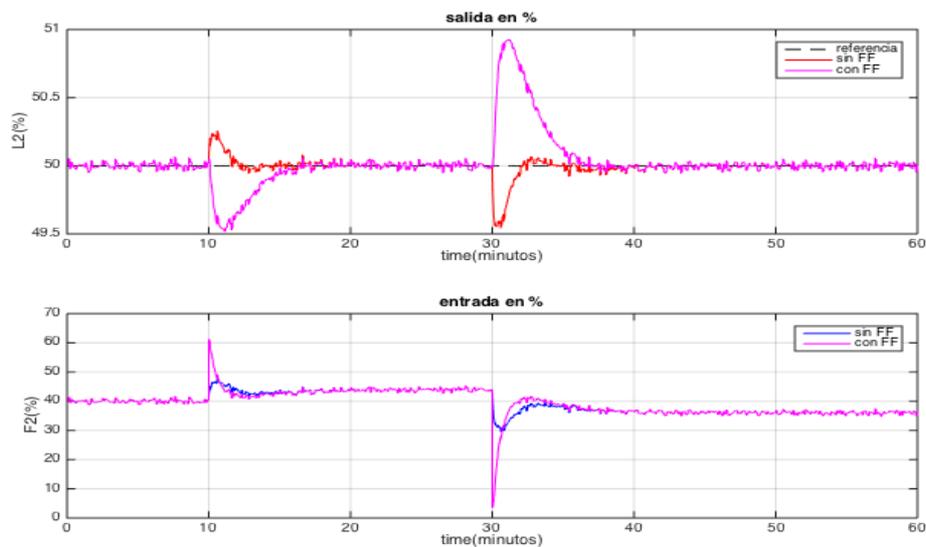


Figura 5.43 Salida $L_2\%$ y entrada $F_2\%$ frente a escalón en F_1 , sin y con acción anticipativa

En este caso el resultado del sistema con feed-forward es peor que la respuesta del sistema sin el control con acción anticipativa. La respuesta a esta situación se encuentra en la propia naturaleza multivariable del sistema:

A la hora de diseñar G_D , se evalúa la respuesta L_2 del sistema no lineal no realimentado frente a un escalón en la perturbación F_1 . Todas las demás variables manipulables y perturbaciones se mantienen en su punto nominal. Sin embargo, a la hora de implementar el control feedforward obtenido y provocar una variación en F_1 , no sólo cambia F_2 , sino el resto de las demás variables manipulables al estar controladas por realimentación (x_{F100} y F_{200}). Este cambio en las demás entradas manipulables es el que provoca la variación inesperada que se puede observar en la **Figura 5.43**.

De hecho, se puede comprobar que, aplicando el control feedforward calculado, variando F_1 y desactivando el bucle de realimentación de X_2 (es decir, se mantiene una x_{F100} constante), la variación en la respuesta de $L_2\%$ es prácticamente nula: rechaza casi a la perfección la perturbación F_1 . Esto es indicativo de que el diseño del controlador anticipativo es el adecuado, para un sistema monovariable (es decir, que no hubiese otras variables que interactuasen con $L_2\%$).

Esta situación es un claro ejemplo de las limitaciones del control SISO (single input- single output) frente al control MIMO (multiple inputs- multiple outputs).

A continuación se muestran las respuestas de $L_2\%$ y $F_2\%$ frente a los mismos cambios en F_1 anteriores, pero manteniendo el resto de entradas fijas en su punto de consigna.

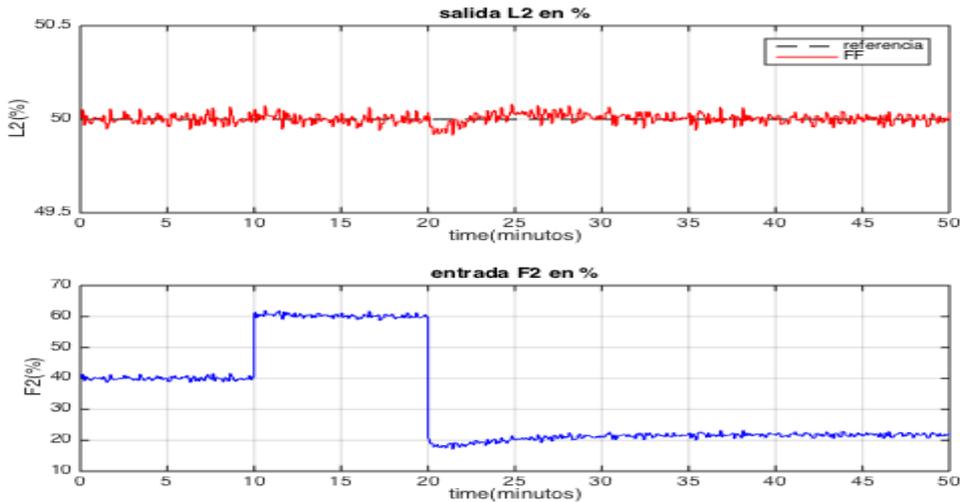


Figura 5.44 Salida $L_2\%$ y entrada $F_2\%$ frente a escalón en F_1 , con FF y resto de bucles abiertos

Para solucionar el problema a la hora de la implementación con todos los lazos de realimentación hay diversas opciones. La opción más completa, idealmente, supondría crear para cada uno de los bucles de realimentación, controles feedforward para cada una de las perturbaciones y el resto de entradas. Así, por ejemplo en este caso para el bucle realimentado $L_2\%-F_2\%$, habría que crear los controles realimentados para las 5 perturbaciones del sistema y para las otras dos variables manipulables: X_{F100} y F_{200} . De la misma manera, se tendría que hacer dicho proceso para los otros dos bucles realimentados, creando finalmente un total de $7 \cdot 3 = 21$ controles feedforward. Esta solución es evidentemente tediosa y compleja. Sólo sería recomendable como solución al problema de control SISO si se necesitase una alta capacidad de control frente a perturbaciones y se pudiesen medir todas estas.

La otra opción, que es la que se toma en este apartado, es modificar manualmente la ganancia de $G_F(s)$ para intentar minimizar lo máximo posible el efecto de la perturbación F_1 sobre L_2 .

Finalmente, para un valor de $G_F(s) = 0.5$, se obtiene una mejoría con respecto a la respuesta anterior. La siguiente Figura representa la respuesta del sistema al implementar dicho $G_F(s)$ frente a la misma entrada F_1 de la **Figura 5.43**:

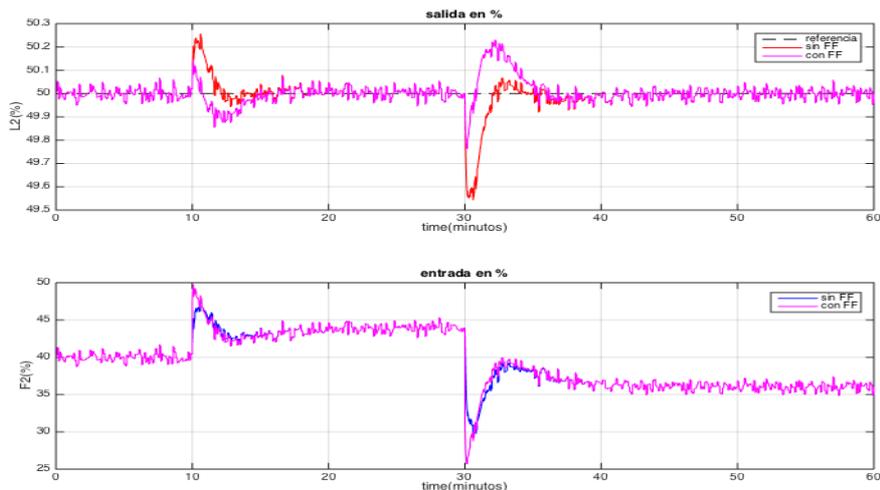


Figura 5.45 Salida $L_2\%$ y entrada $F_2\%$ frente a escalón en F_1 , sin y con control anticipativo 2

5.5.4 Control en cascada para P_0 en el emparejamiento $X_2' - xF_{100}$

Los sistemas de control en cascada están dotados de un controlador maestro, y uno o varios controladores esclavos. De esta manera, la salida de un controlador de realimentación maestro es el punto de consigna para otro (esclavo).

Este tipo de estructura es realizable a la hora de controlar un sistema dividido en varios subsistemas con entradas y salidas medibles. Esto permite controlar una variable secundaria con anterioridad a que sus efectos se hagan notar en la variable principal del sistema, lo que proporciona al sistema controlado de una mayor capacidad y rapidez de control de las variables de salida, frente a perturbaciones que afecten directamente a un subsistema realimentado con un control esclavo.

Es especialmente útil cuando uno de los subsistemas a controlar dispone de una dinámica mucho más rápida que la variable principal controlada. Es el caso en este sistema evaporador, en el que la dinámica entre la presión en el evaporador y la presión de vapor de caldera es mucho más rápida que la dinámica entre X_2' y F_{100} .

Se va a realizar, por tanto, un control en cascada para el emparejamiento $X_{F100}\% - X_2'\%$, donde el controlador maestro recibe la medida de $X_2'\%$, y da como salida el punto de consigna del controlador esclavo que controla el caudal de vapor.

El objetivo principal de esta implementación es reducir todo lo posible el impacto de una variación en la perturbación P_0 en la variable de salida X_2' , actuando previamente sobre la salida $F_{100}\%$ del subsistema de la válvula. Este tipo de control no tiene impacto en los cambios de referencia de X_2' .

La siguiente figura muestra el esquema de Simulink seguido a la hora de implementar el sistema de control en cascada. Es importante recalcar que el sistema de prealimentación para F_1 cambia ligeramente, pues la suma del resultado del control prealimentado se debe realizar en la salida del primer controlador maestro. Puesto que la salida de dicho controlador es una referencia porcentual de F_{100} , y la salida de $H_1(s)$ es xF_{100} , se han añadido 2 ganancias, la primera es la ganancia que ajusta xF_{100} a F_{100} , y la segunda F_{100} a $F_{100}\%$.

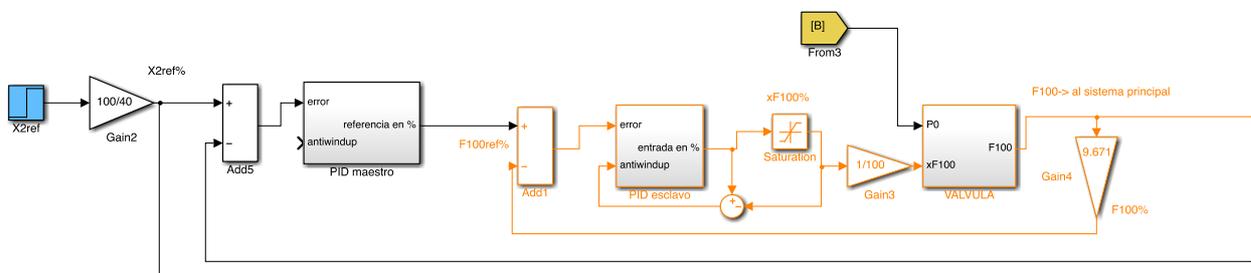


Figura 5.46 Esquema de Simulink para el control en cascada del emparejamiento $xF_{100}-X_2'$

El proceso a seguir para sintonizar los controladores en cascada es el siguiente [7]:

1. Sintonizar el controlador esclavo basándose en el esquema de la **Figura 5.40** y las siguientes consideraciones calculadas al inicio del capítulo 5 y capítulo 3:
 - $G_{D_1}(s) = \frac{F_{100}(s)}{P_0(s)}$. Función de transferencia de X_2' con respecto a la perturbación P_0 .
 - $G_V(s) = \frac{F_{100}(s) \%}{X_{F100}(s) \%}$. Función de transferencia del caudal de vapor con respecto a la salida del elemento de control (apertura de la válvula).

El objetivo de control es que el error a la entrada del controlador esclavo sea nula, es decir:

$$\begin{aligned}
e \cdot G_{CM} - u \cdot G_{TE} &= 0; \quad e \cdot G_{CM} = F_{100}^{ref} \%(s); \quad u \cdot G_{TE} = F_{100} \%(s); \\
\left((F_{100}^{ref} \% - F_{100} \%) \cdot G_{CE} \cdot G_V + G_{D_1} \cdot P_0 \right) \cdot G_{TE} &= F_{100} \% \\
F_{100} \% &= \frac{F_{100}^{ref} \% \cdot G_{CE} G_V G_{TE} + P_0 \cdot G_{D_1} G_{TE}}{1 + G_{CE} G_V G_{TE}}
\end{aligned} \tag{5-38}$$

Teniendo en cuenta que $G_V(s) = 1.8 \% / \%$, $G_{D_1}(s) = 0.93 \text{ kg}/(\text{min} \cdot \text{bar})$ y $G_{TE}(s) = 9.671 \% / (\text{kg}/\text{min})$:

$$F_{100} \% = \frac{F_{100}^{ref} \% \cdot G_{CE} \cdot 17.41 + P_0 \cdot 9}{1 + G_{CE} \cdot 17.41} \tag{5-39}$$

Se busca que $F_{100} \%$ sea igual a $F_{100}^{ref} \%$ (salida=referencia). Para ello, G_{CE} debe de ser lo suficientemente grande como para que $F_{100}^{ref} \% \approx F_{100} \%$. Así, para un controlador proporcional de ganancia $K_c=50$ y P_0 tomando el valor extremo 7.5 bar:

$$F_{100} \% = \frac{870.5}{871.5} F_{100}^{ref} \% + \frac{9 \cdot 2.5}{871.5} = 0.999 \cdot F_{100}^{ref} \% + 0.02 \approx F_{100}^{ref} \% \tag{5-40}$$

Se usará por tanto como controlador esclavo el controlador proporcional $K_{CE} = 50 \frac{\%}{\%}$.

- Una vez sintonizado el controlador esclavo, se procede a identificar la función de transferencia del sistema comprendido entre la entrada del bucle secundario (en este caso $F_{100}^{ref} \%$) y la salida $X_2' \%$. Para ello se mantiene el bucle secundario cerrado y se procede mediante los métodos de identificación (ident) vistos previamente, resultando:

$$G(s) = \left(\frac{X_2' \%(s)}{F_{100}^{ref} \%(s)} \right) = \frac{2.738}{3.519s + 1} e^{-0.6s} \frac{\%}{\%} \tag{5-41}$$

- Finalmente, se sintonizan los parámetros del controlador primario según las reglas SIMC de Skogestad:

$$\text{PID} \left\{ \begin{aligned}
\mathbf{k}_c &= \frac{1}{2.738} \frac{2 \cdot 3.519 + 0.6}{2 \cdot 0.880 + 0.6} = \mathbf{1.182} \\
\mathbf{t}_i &= 3.519 + \frac{0.6}{2} = \mathbf{3.819} \\
\mathbf{t}_d &= \frac{3.519 \cdot 0.6}{2 \cdot 3.519 + 0.6} = \mathbf{0.2764}
\end{aligned} \right. \tag{5-42}$$

Con los dos controladores diseñados, sólo falta implementar el esquema en Simulink tal como se vio en la **Figura 5.46**. Cabe destacar un aspecto importante en la implementación:

- El controlador primario no lleva asociado posteriormente un bloque de saturación, debido a que la salida que ofrece, $F_{100}^{ref} \%$, no es una variable manipulable y, por lo tanto, no debe ser limitada. Sin embargo, el controlador secundario, al controlar la entrada manipulable $x_{F_{100}} \%$, sí lleva asociado el bloque pertinente de saturación.

La **Figura 5.47** muestra las respuestas de las salidas $X_2' \%$ y las entradas $x_{F_{100}} \%$, sin y con control en cascada, frente a una variación en la presión de suministro de caldera (P_0) con respecto al punto nominal de

funcionamiento del +10% en el minuto 10, y del -10% en el minuto 30.

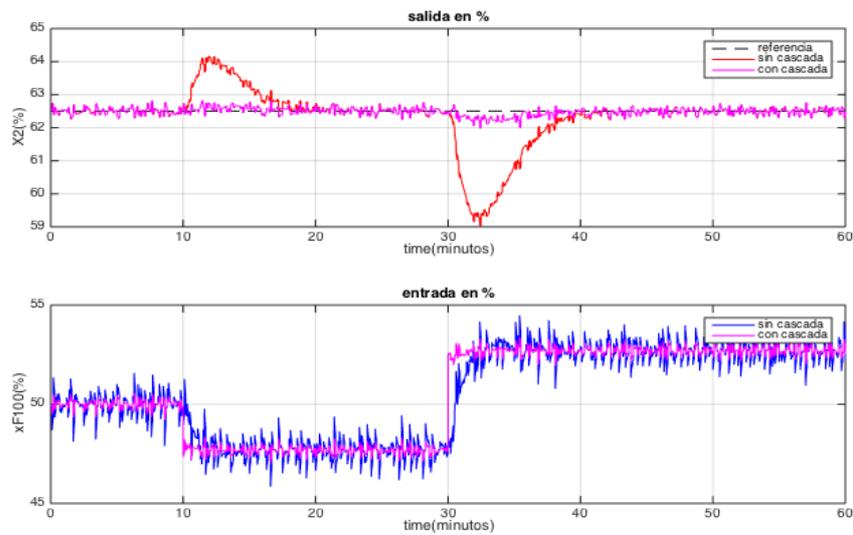


Figura 5.47 : salida X_2 '% y entrada x_{F100} % frente a perturbación P_0 sin y con control en cascada

En los resultados se observan dos claras mejorías:

1. La menor sobreoscilación provocada en X_2 '% frente a variaciones en P_0 (error máximo en transitorio). En concreto, el sistema sin control en cascada alcanza picos del 59% en X_2 ', mientras que con el control en cascada apenas alcanza el 62%.
2. Se comprueba el efecto del control en cascada en la entrada x_{F100} %. Inmediatamente después de aparecer la perturbación en el sistema, la entrada reacciona más rápidamente que el sistema sin control en cascada, para disminuir los efectos de la perturbación antes de evaluar la respuesta de la concentración. Esto se debe a la rapidez de la dinámica entre la presión de la caldera y la presión en el interior de la cámara del evaporador.
3. Es posible integrar el control anticipativo previamente calculado y el control en cascada. En la **Figura 5.48** se muestra la integración de las 2 acciones de control.

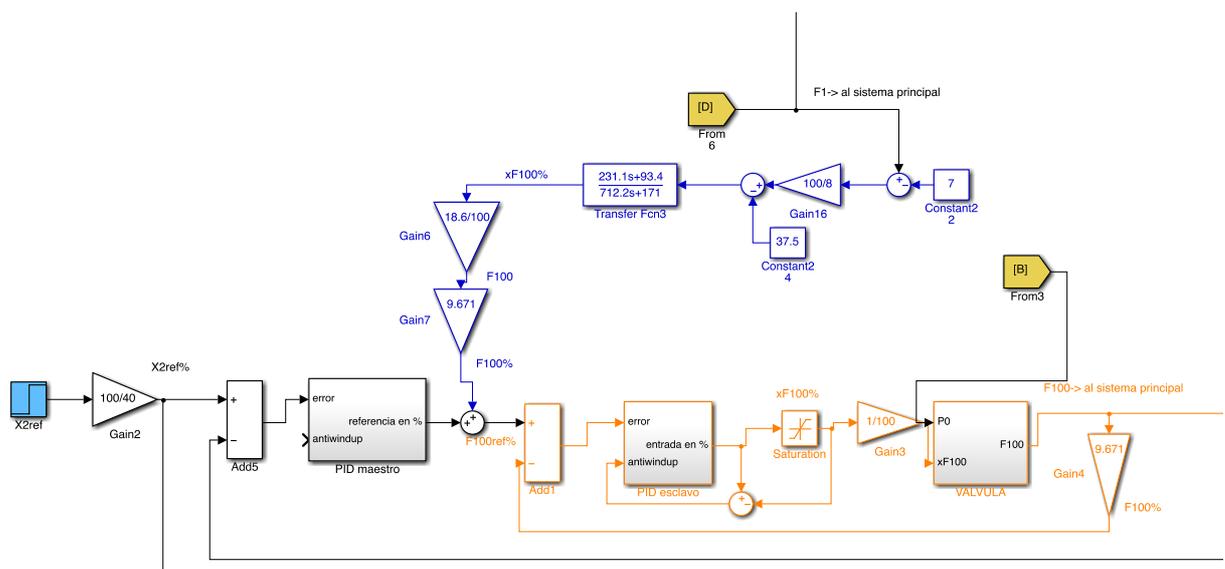


Figura 5.48 Control en cascada + acción anticipativa

5.6 Control SISO completo con todos los controladores

El diagrama de bloques utilizado en este apartado se muestra en el anexo B (diagrama control SISO completo). Incluye todas las técnicas de control desarrolladas en este capítulo, actuando en conjunto. Están incluidos, por tanto:

- 3 lazos de realimentación para los emparejamientos $X_{F100} - X_2'$, $F_{200} - P_2$ y $F_2' - L_2$.
- control feed-forward para $F_1 - X_2'$, $T_{200} - P_2$ y $F_1 - L_2'$.
- Control en cascada para $P_0 - X_2'$.

5.6.1 Control SISO completo frente a perturbaciones

A continuación se compara, por un lado, la respuesta de las variables de salida y variables manipulables del sistema SISO completo con todas las estrategias de control previamente desarrolladas implementadas, y por otro, la respuesta de dichas variables sin los controles feedforward y sin el control en cascada, frente a perturbaciones. La tabla siguiente muestra la secuencia de variación de las variables de perturbación:

Tiempo(min)	20	40	60	80	100	120	140	160	180	200
$P_0(\text{bar})$	+10							-5		
$F_1(\text{kg/min})$		-15			+5					
$X_1(\%)$			+10							
$T_1(^{\circ}\text{C})$						+20			+10	
$T_{200}(^{\circ}\text{C})$				+15			+5			

Tabla 5.2 Cadena de cambios en perturbaciones

La **Figura 5.49** muestra la respuesta de todas las variables manipulables y de control del sistema, frente a los cambios mostrados en la Tabla 5.2.

Resultados

- Clara mejoría en las variables de control en cuanto a sobreoscilaciones. El control anticipativo y en cascada permiten actuar rápidamente frente a las perturbaciones y evitar picos en el transitorio. Este efecto es especialmente notable en la respuesta de X_2' y L_2' ; los picos del transitorio que se producen en el sistema sin el control FF y cascada se reducen hasta en un 92%.
- En cuanto a las entradas, frente a las perturbaciones se producen cambios más rápidos en orden a rechazarlas en el sistema con control FF y cascada. Esto se debe a la reacción inmediata de la variable manipulable en cuanto se percibe la perturbación. Esta reacción produce un error más pequeño en la salida, pero se aumenta el esfuerzo de control, y por lo tanto el esfuerzo que tiene que hacer la válvula.
- Los controladores diseñados se basan en funciones de transferencia lineales y se implementan en el sistema no lineal. Esta implementación crea lógicamente que la respuesta de las variables de control no sea exactamente la deseada, pues cuanto más alejadas del punto de funcionamiento nominal se encuentren los valores de las perturbaciones, peor será el control al dejar de ser representativas las funciones de transferencia del sistema no lineal. En concreto, cuando se superan en este proyecto valores del $\pm 20\%$ con respecto al punto de funcionamiento de las perturbaciones, el control empieza a ser cada vez menos efectivo y las variables de control tienen un mayor tiempo de establecimiento en

régimen permanente y unos picos de transitorio más acusados.

- El diseño de cada controlador se realiza individualmente para cada pareja variable de control – variable manipulable, manteniendo el resto de variables manipulables y perturbaciones en su punto de funcionamiento. Gracias a la DRGA, los emparejamientos elegidos son los más óptimos, pero no dejan de ser un estudio SISO con limitaciones en las interacciones entre variables (aspecto mejorable con un sistema MIMO). Estas interacciones se pueden ver en la **Figura 5.49**. En concreto, en el minuto 100 de la variable de control X_2 %, se observa como existen 2 picos de transitorio al introducirse una variación en F_1 , en vez de establecerse directamente en la referencia tras el primer pico. Esto se debe a la interacción con otras variables.

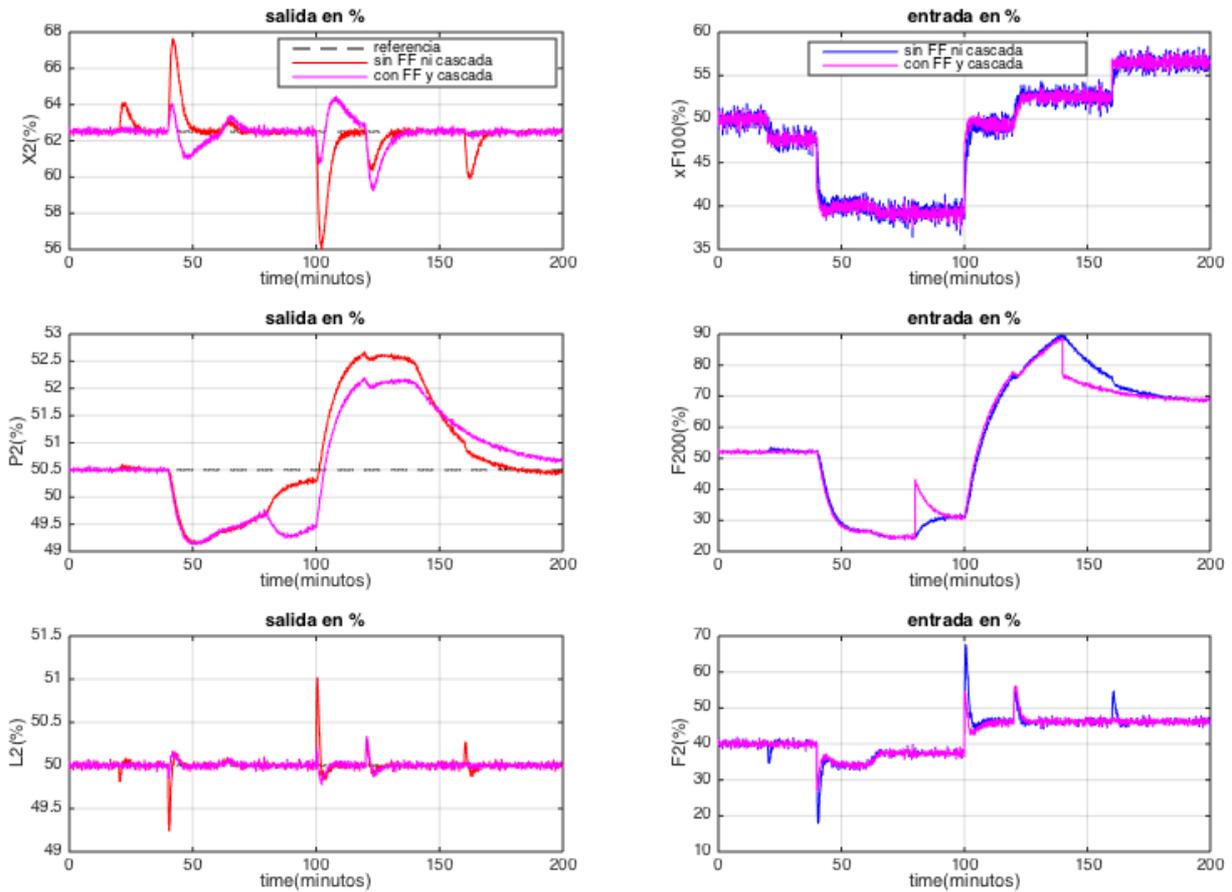


Figura 5.49 respuesta salidas y entradas del sistema SISO completo frente a cadena de perturbaciones sin control anticipativo-cascada y con él

5.6.2 Control SISO completo frente a seguimiento de referencias

A continuación se compara, por un lado, la respuesta de las variables de salida y variables manipulables del sistema SISO completo con todas las estrategias de control previamente desarrolladas implementadas, y por otro, la respuesta de dichas variables sin los controles feedforward y sin el control en cascada, frente a seguimiento de referencias. La tabla siguiente muestra la secuencia de variación de las referencias:

Tiempo(min)	20	40	60	80	100	120	140
X_2 'ref (%)	+10			-10			
P_2 ref (%)		-20			0		
L_2 ref (%)			-15			0	

Tabla 5.3 Cadena de cambios en referencias

Respuesta de todas las variables de entrada y de control del sistema:

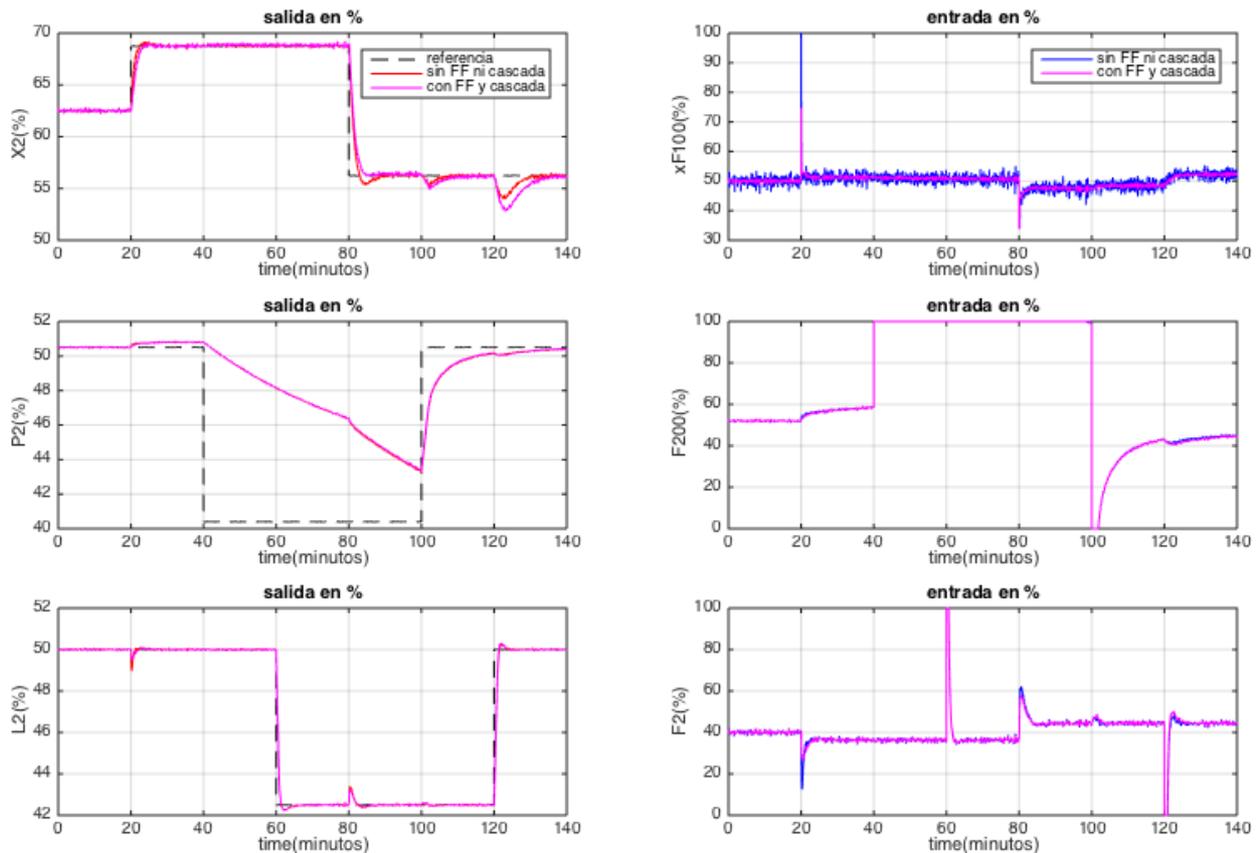


Figura 5.50 respuesta salidas y entradas del sistema SISO completo frente a cadena de referencias sin control anticipativo-cascada y con él

- La respuesta en este caso de las entradas y las salidas es prácticamente idéntica para ambos modelos (sin y con control avanzado integrado). De hecho, el único motivo por el cual se observa una ligera diferencia en X_2 '% es que el controlador PID diseñado para el modelo en cascada no es exactamente el mismo que el diseñado sin cascada. Esta similitud casi exacta radica en el hecho de que el diseño de controladores anticipativos y en cascada se basan en el rechazo de perturbaciones, pero no modifican la respuesta frente a cambios en las referencias.
- Cabe destacar la lenta velocidad de cambio de P_2 '%: la entrada se satura entre 0-400 kg/min por lo que una vez alcanzado dichos límites, como ocurre en este caso, la pendiente de la presión de salida es limitada en unos valores máximo y mínimo. La saturación es una no linealidad muy importante, pues en el diseño de controladores lineales no se tiene en cuenta, por ejemplo, que las válvulas estén limitadas a los valores 0%-100% (completamente cerrada – completamente abierta). Se pueden dar situaciones en el control lineal en las que el controlador ofrezca un valor de salida mayor o menor a esos límites, lo cual es imposible.

Para solventar este problema en el sistema no lineal se añaden bloques de saturación, que crean una simulación más realista y limitan la velocidad de respuesta de las variables de control. Se pueden observar también como F_2 ' y X_{F100} alcanzan límites de saturación.

- Son notables otras no linealidades en la respuesta. Bajar la presión del sistema es más complicado que subirla. Así, cuando se pide una referencia del -20% con respecto al punto de funcionamiento de P_2 , la

entrada F_{200} se satura en 400 kg/min y aun así la pendiente de bajada de P_2 es poco pronunciada. Sin embargo, al volver a requerir un incremento de la referencia, la entrada F_{200} se satura en 0 kg/min y la presión del evaporador alcanza el valor de referencia en mucho menos tiempo.

- Se comprueba la idoneidad de los emparejamientos variable de control – variable manipulable adoptados tras el análisis de la DRGA. Sin embargo, se pueden apreciar efectos típicos del sistema multivariable. L_2 presenta picos en los minutos 20 y 80 debido a los cambios de referencia en X_2' y por tanto en X_{F100} . P_2 también se ve afectada en el minuto 80 con un pequeño pico debido al cambio en X_{F100} . Por su parte, X_2' también presenta picos en el transitorio debido a los cambios de F_{200} en el minuto 100 y de F_2' en el minuto 120.

5.7 Comparación con otros autores

La mayoría de autores ([2],[3],[11]) que abordan el modelo del evaporador de tubos largos verticales con recirculación forzada toman como referencia el modelo dinámico de Newell and Lee [6]. Puesto que aquí se modifica este modelo, los resultados que se obtienen son diferentes a otros documentos, aunque sí se pueden distinguir las similitudes y diferencias siguientes:

- En cuanto a las entradas y perturbaciones, el modelo de Newell and Lee considera que F_3 es una perturbación (ver **Capítulo 2**), y que P_{100} es una entrada manipulable del sistema. Puesto que en realidad dichas variables no pueden modificarse directamente, en este proyecto F_3 no es una perturbación.
- Los emparejamientos escogidos para el control SISO sí coinciden con los escogidos por otros autores [2],[3],[11]. Debido a que en otras publicaciones una variable manipulable es P_{100} , los valores de la matriz RGA son diferentes.
- La solución de control realimentado que ofrecen consiste también en integrar controladores PI en los 3 bucles de realimentación. La ganancia de estos controladores es por lo general más alta, teniendo en cuenta que es calculado por el método de Ziegler and Nichols y que no se tiene en cuenta el ruido en las señales.
- Con respecto a control prealimentado, Newell and Lee elige también rechazar F_1 con control feedforward sobre el bucle de realimentación utilizado para controlar X_2 .

Este proyecto mejora los siguientes aspectos con respecto a publicaciones anteriores:

- El modelo dinámico ofrece resultados más acertados y coherentes con respecto al comportamiento real de las variables de control. En la **Figura 5.51** [6] se muestran las respuestas de las variables de control X_2 y P_2 frente a escalones en P_{100} y F_{200} , respectivamente, según el modelo de Newell and Lee. En dicho modelo, X_2 evoluciona siguiendo un comportamiento subamortiguado, y en la respuesta de P_2 existen cambios de pendiente abruptos, sin cambios en la variable manipulable. Estos comportamientos son singulares e impropios de un evaporador.
- Los controladores se han diseñado por teorías de control modernas (SIMC-Skogestad).
- Se desarrolla un modelo completo de simulación en Simulink y se incluyen sensores y ruido blanco en la implementación.

Los resultados obtenidos como consecuencia de estas variaciones son mejores y más realistas con respecto a otros estudios previos sobre la materia en cuestión.

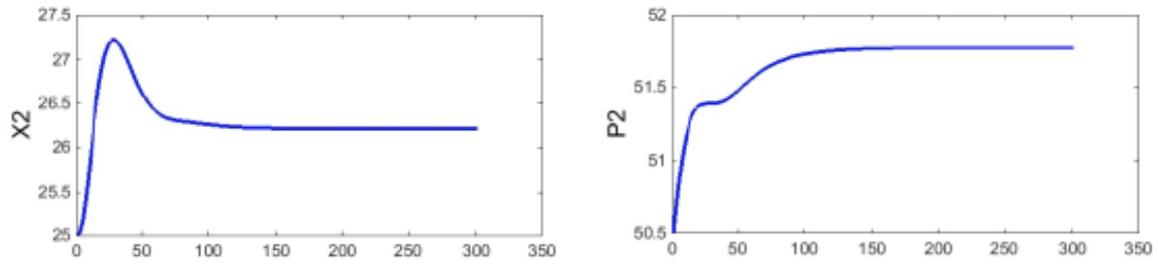


Figura 5.51 Respuestas X_2 frente a +10 KPa en P_{100} y P_2 frente a +10 kg/min en F_{200} . Modelo Newell and Lee

6 INTERFAZ USUARIO EN TIEMPO REAL

Partiendo como base del modelo de Simulink presentado en el Capítulo 4, se ha desarrollado una GUI (Graphical User Interface) que permite al usuario modificar los principales parámetros del sistema de control y observar los resultados en tiempo real sin necesidad de modificar nada directamente sobre el modelo de Simulink.

Para ejecutar la GUI y hacerla funcionar el usuario debe seguir los siguientes pasos:

1. Abrir la aplicación MATLAB® e introducir “Simulink” en la línea de comandos.
2. Abrir el archivo “CONTROLSISO_DEFINITIVO_REALTIME.slx” y pulsar sobre el “bloque Scope” en rojo nombrado como “SALIDAS %” y el bloque azul “ENTRADAS%”. Esto permitirá observar la respuesta de las salidas y las entradas en la gráficas emergentes.
3. Compilar y ejecutar (botón “RUN”) de los archivos “parámetros.m” y “miGUI.m”. Al final de este se desplegará la GUI, que se muestra en la Figura siguiente:

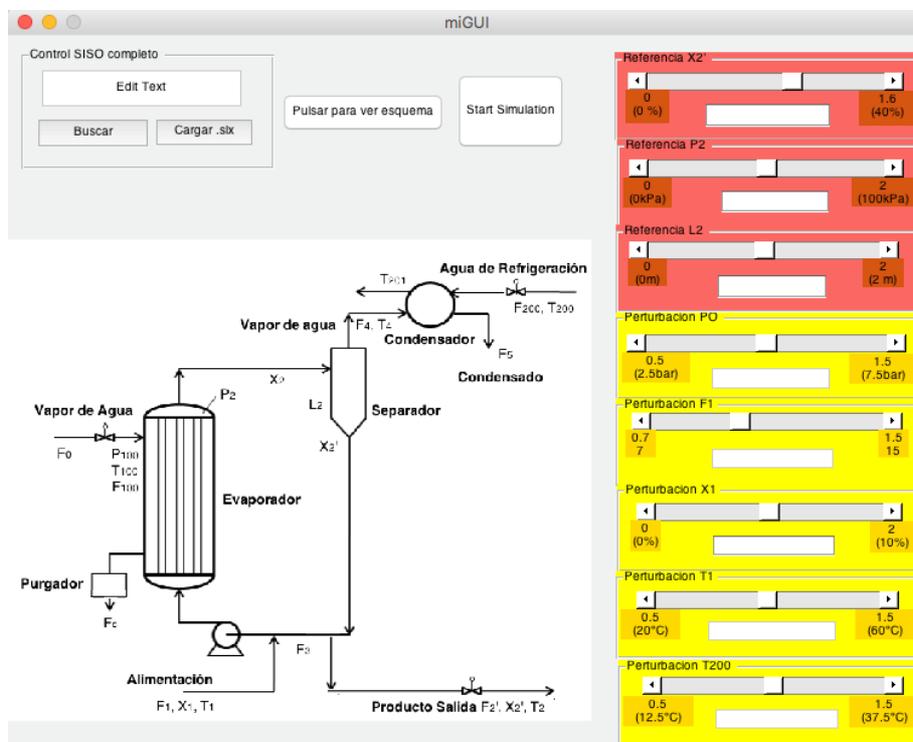


Figura 6.1 Pantalla GUI

En esta pantalla se puede interactuar con diversos elementos. Lo primero que se debe hacer para ello es pulsar “Buscar” y elegir el archivo “CONTROLSISO_DEFINITIVO_REALTIME.slx”. Posteriormente, al pulsar sobre “Cargar .slx” la GUI quedará enlazada con el sistema de Simulink y se podrá actuar sobre él.

En la pantalla se pueden observar 3 bloques en rojo y 5 bloques en amarillo. Corresponden a las referencias de las salidas y a las perturbaciones, respectivamente, en sus valores absolutos. Además, se puede observar una imagen que muestra el esquema del sistema evaporador.

Cada bloque tiene una barra “slider”, en la que se puede deslizar el bloque interior de un lado a otro para cambiar el valor de la ganancia con respecto al punto nominal de funcionamiento para modificar el valor de la referencia o perturbación. El valor de dicha ganancia se muestra en el bloque rectangular blanco inferior, donde también se puede introducir un valor numérico concreto. Además, en cada extremo de la barra hay 2 bloques con 2 números, que indican el valor mínimo y máximo que puede tomar la ganancia y valor absoluto de la variable.

A modo de ejemplo, el primer bloque rojo, (“Referencia X_2 ”) está limitado entre 0% y 40%, por lo que la

ganancia que podemos introducir está comprendida entre 0 y 1.6 ($25 \cdot 0 = 0$, $25 \cdot 1.6 = 40$).

El funcionamiento de la GUI consiste en trasladar los valores que se introducen en ella a las ganancias conectadas a los bloques “step” de las referencias y perturbaciones de Simulink (ver Anexo B), modificándolas a la par que lo hace el usuario.

Por otro lado, aunque los límites de las variables de salida se conocen (ver **Tabla 1.1**), los límites de las perturbaciones son un dato desconocido puesto que éstas son externas y no dependen del sistema de estudio. En este proyecto se estiman los siguientes límites:

Punto nominal	Límite Inferior (Ganancia)	Límite Inferior (Valor Absoluto)	Límite Superior (Ganancia)	Límite Superior (Valor Absoluto)
$P_0^0 = 5 \text{ bar}$	0.5	2.5 bar	1.5	7.5 bar
$F_1^0 = 10 \text{ kg/min}$	0.7	7 kg/min	1.5	15 kg/min
$X_1^0 = 5 \%$	0	0 %	2	10 %
$T_1^0 = 40 \text{ }^\circ\text{C}$	0.5	20 $^\circ\text{C}$	1.5	60 $^\circ\text{C}$
$T_{200}^0 = 25 \text{ }^\circ\text{C}$	0.5	12.5 $^\circ\text{C}$	1.5	37.5 $^\circ\text{C}$

Tabla 6.1 Límites perturbaciones

4. Pulsando “Start Simulation” se inicia la simulación, en la que se pueden ir variando las ganancias y ver sus consecuencias directas. Un ejemplo de ello se muestra en la **Figura 6.2**. Pulsando en “Stop Simulation” se puede parar la simulación y reiniciarla de nuevo con “Start Simulation”.

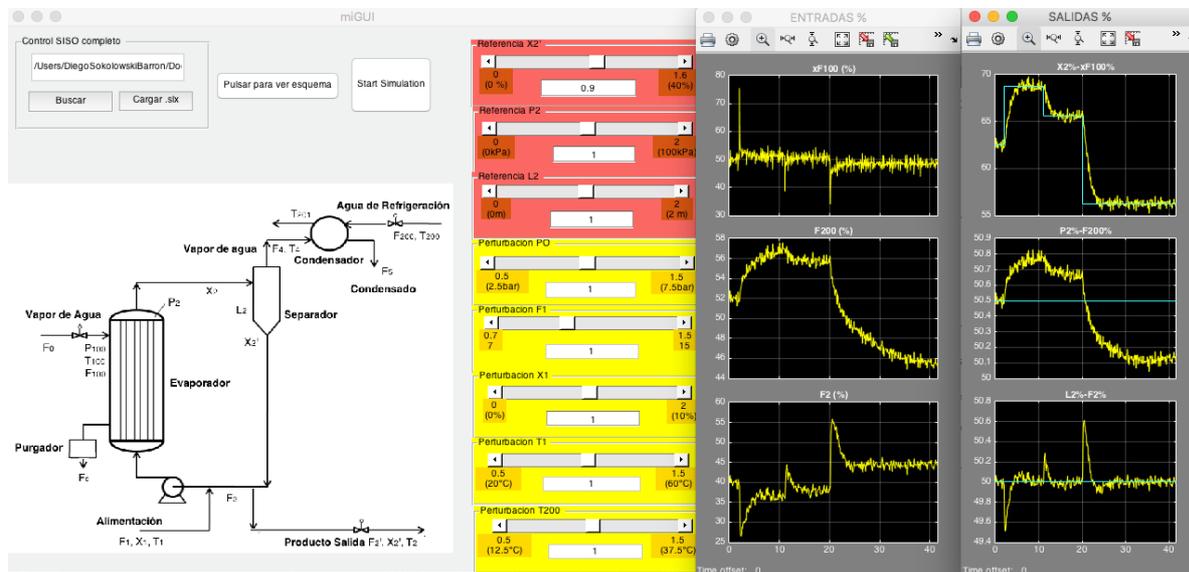


Figura 6.2 Simulación GUI

El código utilizado para la creación de la GUI se recoge en el Anexo A. Por otro lado, para conseguir que la simulación funcione ralentizada simulando el efecto del tiempo real, se utiliza el bloque de Simulink “Real-Time Pacer” [13].

Se ha creado otra GUI, denominada: “miGUI_nocontrol.m”, que se puede utilizar de la misma manera para la simulación del modelo no lineal no controlado: “modelo_no lineal_final_REALTIME.slx”.

7 CONCLUSIONES Y LÍNEAS FUTURAS

7.1 Conclusiones

En este proyecto se ha trabajado en el desarrollo del modelo dinámico de un evaporador de recirculación forzada y de su sistema e control, así como en su implementación en Simulink para poder analizar los resultados expresados gráficamente. Tomando como base los resultados obtenidos a lo largo de los capítulos anteriores, se destacan las siguientes conclusiones:

- El modelo dinámico ofrece un comportamiento admisible para la dinámica de las variables de salida. Sin embargo, el modelo sigue siendo simplificado y tiene limitaciones, aunque los resultados son satisfactorios bajo las consideraciones de funcionamiento tomadas.
- Se han mejorado modelos anteriores y se han incluido cambios importantes. En concreto, en cuanto al modelo dinámico, se ha incluido el modelo de tanques homogéneos en serie para el evaporador, se ha dotado de dinámica a determinadas variables que otros autores no consideran, se ha incluido un purgador de condensado, se ha considerado la presión de vapor saliente de una caldera como perturbación, se ha añadido el modelo de la válvula del caudal de vapor de entrada y se ha modificado la localización de la bomba de impulsión para aproximar el modelo a un evaporador real.

Por otro lado, en cuanto al control, se han añadido no linealidades en la simulación, ruido blanco, y se han adimensionalizado las variables de control y variables manipulables para una correcta implementación en controladores P, PI, PID. Además, se utilizan técnicas de diseño de parámetros de controladores más modernas (SIMC) y se hace un estudio de acciones anticipativas y control en cascada. Se han añadido sensores en el modelo de control.

- La utilización de Simulink es de gran ayuda en la visualización de los resultados. Con el modelo que se ha creado en este proyecto se pueden modificar fácilmente las variables del sistema y observar las diferentes respuestas en cuestión de segundos, todo ello en una interfaz gráfica intuitiva.

La división de diagramas en diferentes niveles permite observar con claridad la implementación de las diferentes ecuaciones. Así, el primer nivel muestra la configuración del evaporador en bloques y las entradas, perturbaciones y salidas principales, lo que convierte al archivo en una herramienta adecuada para la docencia y útil para usuarios que no dispongan de conocimientos avanzados del programa, ya que no se requieren apenas conocimientos del mismo para ejecutarlo y cambiar los principales parámetros. De manera adicional, el diagrama de Simulink sirve como base para futuras ampliaciones del proyecto.

- Se ha comprobado la utilidad de la RDGA a la hora de hacer la elección de los mejores emparejamientos SISO. La elección surgida de este análisis ha permitido obtener resultados muy favorables en un sistema que no incluye un control multivariable.
- El control realimentado del sistema no lineal con controladores sintonizados por el método SIMC ofrece buenos resultados en el seguimiento de referencias. No se producen errores graves como inestabilidad ni existen errores en régimen permanente ni grandes sobreoscilaciones. A su vez, se comprueba la utilidad de herramientas como “anti-windup” y el filtro derivativo.
- La utilización de control prealimentado y control en cascada permite mejorar considerablemente el rechazo frente a perturbaciones. Integrados estos controladores con los de realimentación, se consigue un modelo de control SISO capaz de disminuir el efecto de las perturbaciones y de realizar un seguimiento efectivo de las referencias.

7.2 Líneas futuras

1. Para mejorar la simulación y los resultados de futuros estudios, se puede profundizar más en el modelo dinámico planteado. Hay diversas variables cuyas dinámicas se han despreciado en este

proyecto, y que pueden ser objeto de desarrollo, como, por ejemplo, la dinámica del condensador.

2. El siguiente paso lógico con respecto al sistema de control consiste en implementar técnicas de control multivariable para mejorar los problemas que surgen de la interacción entre variables. Para ello, existen diversos campos de estudio como el regulador multivariable lineal (LMR), así como controles avanzados adaptativos y predictivos.

Apéndice A. Códigos Matlab

A.1 Parámetros y puntos nominales de funcionamiento

Código A.1 parametros.m

```
%parámetros sistema
pA= 20;
M= 20;
C= 4;
Cp= 0.07;
yr= 38.5;
ys= 36.6;
UA2= 6.84;
pi= 3.1416;
%puntos de equilibrio (puntos nominales de funcionamiento)
P0=5;
xF100E=0.5;
X1E=5;
X2E=25;
F1E=10;
F2E=2;
F3E=50;
F4E=8;
F5E=8;
T1E=40;
T2E=84.6;
T3E=80.6;
L2E=1;
P2E=50.5;
F100E=9.3;
T100E=119.9;
P100E=194.7;
Q100E=339;
F200E=208;
T200E=25;
T201E=46.1;
Q200E=307.9;
```

A.2 Cálculo de puntos nominales de funcionamiento, (evaporador N compartimentos)

Código A.2 pnominalesevap5.m

```
function [matsol]= pnominalesevap5 (F1, T1, X1, F100, P2, n)
Cp=0.08;
ys=36.6;
Cpv= 0.025;...
UA=100.11;
syms T21 X21 F41 F21 TW1;
eq1= F1*Cp*T1-F21*Cp*T21-F41*42.9+UA*(TW1-T21)/n;
```

```

eq2= F1*X1-F21*X21;
eq3= 0.5616*P2+0.3126*X21+48.43-T21;
eq4= F1-F41-F21;
eq5= 340.38/UA+(T21-TW1);

M= [eq1, eq2, eq3, eq4, eq5];
sol= solve (M, 'T21', 'X21', 'F41', 'F21', 'TW1');

T21E=sol.T21;
X21E=sol.X21;
F41E=sol.F41;
F21E=sol.F21;

T21Ereal= max (T21E);
X21Ereal= max (X21E);
double ([T21Ereal;X21Ereal]);

solbest= solve (subs(M, T21, T21Ereal), 'X21', 'F41', 'F21', 'TW1');

TW1E=solbest.TW1;
X21E=solbest.X21;
F41E=solbest.F41;
F21E=solbest.F21;

vect2N= [T21Ereal, zeros(1,n-1)];
vectWN= [TW1E, zeros(1,n-1)];
vectX2N= [X21E, zeros(1,n-1)];
vectF4N= [F41E, zeros(1,n-1)];
vectF2N= [F21E, zeros(1,n-1)];

for (i=2:n)
    syms T2N X2N F4N F2N TWN;
    eq1= (vectF2N(i-1)*Cp*vectT2N(i-1))-(F2N*Cp*T2N)+vectF4N(i-1)*Cpv*vectT2N...
        (i-1)-Cpv*F4N*T2N-(F4N-vectF4N(i-1))*(42.55+0.6*(i-1))+UA*(TWN-T2N)/n;
    eq2= vectF2N(i-1)*vectX2N(i-1)-F2N*X2N;
    eq3= 0.5616*P2+0.3126*X2N+48.43-T2N;
    eq4= vectF2N(i-1)-(F4N-vectF4N(i-1))-F2N;
    eq5= 340.38/UA+(T2N-TWN);

    M= [eq1, eq2, eq3, eq4, eq5];

    sol= solve (M, 'T2N', 'X2N', 'F4N', 'F2N', 'TWN');
    T2NE=sol.T2N;
    T2NEreal= max (T2NE);

    solbest= solve (subs(M, T2N, T2NEreal), 'X2N', 'F4N', 'F2N', 'TWN');
    vect2N(i)= double(T2NEreal);
    vectWN(i)= double(solbest.TWN);
    vectX2N(i)= double(solbest.X2N);
    vectF4N(i)= double(solbest.F4N);
    vectF2N(i)= double(solbest.F2N);
end
matsol= double([vect2N; vectWN; vectX2N; vectF4N; vectF2N]);
end

```

A.3 Representación gráfica de comparación de simulaciones

Código A.3 graphSISOCOMPLETO.m

```
% 3 entradas y 3 salidas con comparación entre 2 simulaciones
figure
%a= introducir aquí la variable de Scope Simulink de la salida
%b= introducir aquí la variable de Scope Simulink de la entrada
% nota: primero una simulación: guardar variables a,b,
% luego segunda simulación: guardar variables c,d
ax1 = subplot(3,2,1); % subplot 1
ax2 = subplot(3,2,2); % subplot 2
ax3 = subplot(3,2,3); % subplot 3
ax4 = subplot(3,2,4); % subplot 4
ax5 = subplot(3,2,5); % subplot 5
ax6 = subplot(3,2,6); % subplot 6
plot(ax1,...
     a.time(:,1),a.signals(1).values(:,2),'black--',...
     a.time(:,1),a.signals(1).values(:,1),'r')
     hold(ax1,'on')
     grid(ax1,'on')
title(ax1,'salida en %')
ylabel(ax1,'X2 (%)')
xlabel(ax1,'time (minutos)')
plot(ax2,...
     b.time(:,1),b.signals(1).values(:,1),'b')
     grid(ax2,'on')
     hold(ax2,'on')
title(ax2,'entrada en %')
ylabel(ax2,'xF100 (%)')
xlabel(ax2,'time (minutos)')
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
plot(ax3,...
     a.time(:,1),a.signals(2).values(:,2),'black--',...
     a.time(:,1),a.signals(2).values(:,1),'r')
     hold(ax3,'on')
     grid(ax3,'on')
title(ax3,'salida en %')
ylabel(ax3,'P2 (%)')
xlabel(ax3,'time (minutos)')
plot(ax4,...
     b.time(:,1),b.signals(2).values(:,1),'b')
     grid(ax4,'on')
     hold(ax4,'on')
title(ax4,'entrada en %')
ylabel(ax4,'F200 (%)')
xlabel(ax4,'time (minutos)')
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
plot(ax5,...
     a.time(:,1),a.signals(3).values(:,2),'black--',...
     a.time(:,1),a.signals(3).values(:,1),'r')
     hold(ax5,'on')
     grid(ax5,'on')
title(ax5,'salida en %')
ylabel(ax5,'L2 (%)')
xlabel(ax5,'time (minutos)')

plot(ax6,...
     b.time(:,1),b.signals(3).values(:,1),'b')...
     grid(ax6,'on')
     hold(ax6,'on')
title(ax6,'entrada en %')
```

```

ylabel(ax6, 'F2(%) ')
xlabel(ax6, 'time(minutos) ')
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
plot(ax1, ...
     c.time(:,1),c.signals(1).values(:,1), 'm')
grid(ax1, 'on')
title(ax1, 'salida en %')
ylabel(ax1, 'X2(%) ')
xlabel(ax1, 'time(minutos) ')
legend(ax1, 'referencia', 'sin prealimentacion ni cascada', 'con prealimentaci n
y cascada')
plot(ax2, ...
     d.time(:,1),d.signals(1).values(:,1), 'm')
grid(ax2, 'on')
title(ax2, 'entrada en %')
ylabel(ax2, 'xF100(%) ')
xlabel(ax2, 'time(minutos) ')
legend(ax2, 'sin prealimentacion ni cascada', 'con prealimentaci n y cascada')
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
plot(ax3, ...
     c.time(:,1),c.signals(2).values(:,1), 'm')
grid(ax3, 'on')
title(ax3, 'salida en %')
ylabel(ax3, 'P2(%) ')
xlabel(ax3, 'time(minutos) ')
plot(ax4, ...
     d.time(:,1),d.signals(2).values(:,1), 'm')
grid(ax4, 'on')
title(ax4, 'entrada en %')
ylabel(ax4, 'F200(%) ')
xlabel(ax4, 'time(minutos) ')
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
plot(ax5, ...
     c.time(:,1),c.signals(3).values(:,1), 'm')
grid(ax5, 'on')
title(ax5, 'salida en %')
ylabel(ax5, 'L2(%) ')
xlabel(ax5, 'time(minutos) ')
plot(ax6, ...
     d.time(:,1),d.signals(3).values(:,1), 'm')
grid(ax6, 'on')
title(ax6, 'entrada en %')
ylabel(ax6, 'F2(%) ')
xlabel(ax6, 'time(minutos) ')

```

A.4 GUI de sistema de control Simulink

Código A.4 miGUI.m

```
function varargout = miGUI(varargin)
%MIGUI M-file for miGUI.fig
%   MIGUI, by itself, creates a new MIGUI or raises the existing
%   singleton*.
%
%   H = MIGUI returns the handle to a new MIGUI or the handle to
%   the existing singleton*.
%
%   MIGUI('Property','Value',...) creates a new MIGUI using the
%   given property value pairs. Unrecognized properties are passed via
%   varargin to miGUI_OpeningFcn. This calling syntax produces a
%   warning when there is an existing singleton*.
%
%   MIGUI('CALLBACK') and MIGUI('CALLBACK',hObject,...) call the
%   local function named CALLBACK in MIGUI.M with the given input
%   arguments.
%
%   *See GUI Options on GUIDE's Tools menu. Choose "GUI allows only one
%   instance to run (singleton)".
%
% See also: GUIDE, GUIDATA, GUIHANDLES

% Edit the above text to modify the response to help miGUI

% Last Modified by GUIDE v2.5 22-Aug-2016 18:46:07

% Begin initialization code - DO NOT EDIT
gui_Singleton = 1;
gui_State = struct('gui_Name',       mfilename, ...
                  'gui_Singleton',  gui_Singleton, ...
                  'gui_OpeningFcn', @miGUI_OpeningFcn, ...
                  'gui_OutputFcn',  @miGUI_OutputFcn, ...
                  'gui_LayoutFcn',  [], ...
                  'gui_Callback',    []);
if nargin && ischar(varargin{1})
    gui_State.gui_Callback = str2func(varargin{1});
end

if nargout
    [varargout{1:nargout}] = gui_mainfcn(gui_State, varargin{:});
else
    gui_mainfcn(gui_State, varargin{:});
end
% End initialization code - DO NOT EDIT

% --- Executes just before miGUI is made visible.
function miGUI_OpeningFcn(hObject, eventdata, handles, varargin)
% This function has no output args, see OutputFcn.
% hObject    handle to figure
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
% varargin   unrecognized PropertyName/PropertyValue pairs from the
```

```

%          command line (see VARARGIN)

% Choose default command line output for miGUI
handles.output = hObject;

% Update handles structure
guidata(hObject, handles);

% UIWAIT makes miGUI wait for user response (see UIRESUME)
% uiwait(handles.figure1);

% --- Outputs from this function are returned to the command line.
function varargout = miGUI_OutputFcn(hObject, eventdata, handles)
% varargout  cell array for returning output args (see VARARGOUT);
% hObject    handle to figure
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Get default command line output from handles structure
varargout{1} = handles.output;

% --- Executes on button press in pushbutton_cargarslx.
function pushbutton_cargarslx_Callback(hObject, eventdata, handles)
% hObject    handle to pushbutton_cargarslx (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
modelname=get(handles.edit_slx,'string');
if isempty(modelname)
    errordlg('You have not input the model name');
end
checkload=~isempty(find_system('type','block_diagram','name',modelname));
if checkload==0
    try
        load_system(modelname);
    catch
    end
end
block_GainA=sprintf('%s/GainA',modelname);
block_GainB=sprintf('%s/GainB',modelname);
block_GainC=sprintf('%s/GainC',modelname);
block_GainD=sprintf('%s/GainD',modelname);
block_GainE=sprintf('%s/GainE',modelname);
block_GainF=sprintf('%s/GainF',modelname);
block_GainG=sprintf('%s/GainG',modelname);
block_GainH=sprintf('%s/GainH',modelname);

%se cogen los valores de los gain de Simulink
gananciaA=get_param([bdroot '/GainA'],'Gain');
gananciaB=get_param([bdroot '/GainB'],'Gain');
gananciaC=get_param([bdroot '/GainC'],'Gain');
gananciaD=get_param([bdroot '/GainD'],'Gain');
gananciaE=get_param([bdroot '/GainE'],'Gain');
gananciaF=get_param([bdroot '/GainF'],'Gain');
gananciaG=get_param([bdroot '/GainG'],'Gain');
gananciaH=get_param([bdroot '/GainH'],'Gain');

```

```

%Value obtained from Simulink slx model is char type
%damos al slider y al cuadro de texto el valor que tienen las ganancias
simul.
set(handles.slider_x2, 'Value', str2double(gananciaA));
set(handles.edit_x2, 'string', num2str(gananciaA));

set(handles.slider_p2, 'Value', str2double(gananciaB));
set(handles.edit_p2, 'string', num2str(gananciaB));

set(handles.slider_l2, 'Value', str2double(gananciaC));
set(handles.edit_l2, 'string', num2str(gananciaC));

set(handles.slider_p0, 'Value', str2double(gananciaD));
set(handles.edit_p0, 'string', num2str(gananciaD));

set(handles.slider_f1, 'Value', str2double(gananciaE));
set(handles.edit_f1, 'string', num2str(gananciaE));

set(handles.slider_x1, 'Value', str2double(gananciaF));
set(handles.edit_x1, 'string', num2str(gananciaF));

set(handles.slider_t1, 'Value', str2double(gananciaG));
set(handles.edit_t1, 'string', num2str(gananciaG));

set(handles.slider_t200, 'Value', str2double(gananciaH));
set(handles.edit_t200, 'string', num2str(gananciaH));

%Create handles so that they can be accessed across callback.
handles.modelname=modelname;
handles.block_GainA=block_GainA;
handles.block_GainB=block_GainB;
handles.block_GainC=block_GainC;
handles.block_GainD=block_GainD;
handles.block_GainE=block_GainE;
handles.block_GainF=block_GainF;
handles.block_GainG=block_GainG;
handles.block_GainH=block_GainH;
guidata(hObject, handles)

% --- Executes on button press in pushbutton_buscarslx.
function pushbutton_buscarslx_Callback(hObject, eventdata, handles)
% hObject      handle to pushbutton_buscarslx (see GCBO)
% eventdata    reserved - to be defined in a future version of MATLAB
% handles      structure with handles and user data (see GUIDATA)
[input_file,pathname] = uigetfile( ...
    {'*slx', 'mdl Files (*.slx)';...
    '*.*', 'All Files (*.*)'}, ...
    'Select files', ...
    'MultiSelect', 'on');
if pathname == 0
    return
end
%gets the current data file names inside the listbox
inputfile= fullfile(pathname,input_file);
current_folder=strcat(cd, '\\');
slxname=strrep(inputfile,current_folder, '');
slxname=strrep(slxname, '.slx', '');
%updates the gui to display all filenames in the listbox
set(handles.edit_slx, 'String', slxname);
guidata(hObject, handles);

```

```

% --- Executes on button press in pushbutton_quitarslx.
function pushbutton_quitarslx_Callback(hObject, eventdata, handles)
% hObject      handle to pushbutton_quitarslx (see GCBO)
% eventdata    reserved - to be defined in a future version of MATLAB
% handles      structure with handles and user data (see GUIDATA)

% --- Executes on button press in pushbutton_run.
function pushbutton_run_Callback(hObject, eventdata, handles)
% hObject      handle to pushbutton_run (see GCBO)
% eventdata    reserved - to be defined in a future version of MATLAB
% handles      structure with handles and user data (see GUIDATA)
mystring = get(hObject, 'String');
status = get_param(bdroot, 'simulationstatus');

if strcmp(mystring, 'Start Simulation')

    % Check the status of the simulation and start it if it's stopped
    if strcmp(status, 'stopped')
        set_param(bdroot, 'simulationcommand', 'start')
    end

    % Update the string on the pushbutton
    set(handles.pushbutton_run, 'String', 'Stop Simulation')

elseif strcmp(mystring, 'Stop Simulation')

    % Check the status of the simulation and stop it if it's running
    if strcmp(status, 'running')
        set_param(bdroot, 'SimulationCommand', 'Stop')
    end

    % Update the string on the pushbutton
    set(handles.pushbutton_run, 'String', 'Start Simulation')

else
    warning('Unrecognized string for pushbutton_run') %#ok<WNTAG>
end

% Assign handles and the startstop object to the base workspace
assignin('base', 'miGUI_handles', handles)
assignin('base', 'run_hObject', handles.pushbutton_run)

function edit_12_Callback(hObject, eventdata, handles)
% hObject      handle to edit_12 (see GCBO)
% eventdata    reserved - to be defined in a future version of MATLAB
% handles      structure with handles and user data (see GUIDATA)

% Hints: get(hObject, 'String') returns contents of edit_12 as text
%        str2double(get(hObject, 'String')) returns contents of edit_12 as a
double
value = get(hObject, 'String');

%valores l mites 2-0
if (str2double(value)>2.0)

```

```

        value=num2str(2.0);
    end

    if (str2double(value)< 0)
        value=num2str(0);
    end

    % Update the model's gain value
    set_param([bdroot '/GainC'],'Gain',value)
    set(handles.edit_l2,'String',value);

    slider_position =str2double(value);
    set(handles.slider_l2,'Value',slider_position);

    % Update simulation if the model is running
    status = get_param(bdroot,'simulationstatus');
    if strcmp(status,'running')
        set_param(bdroot, 'SimulationCommand', 'Update')
    end
    guidata(hObject,handles);

% --- Executes during object creation, after setting all properties.
function edit_l2_CreateFcn(hObject, eventdata, handles)
% hObject    handle to edit_l2 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
%         See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

% --- Executes on slider movement.
function slider_l2_Callback(hObject, eventdata, handles)
% hObject    handle to slider_l2 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'Value') returns position of slider
%         get(hObject,'Min') and get(hObject,'Max') to determine range of
slider
slider_position = get(hObject,'Value');
value = num2str(slider_position);

% Update the model's gain value
set_param([bdroot '/GainC'],'Gain',value)

% Set the value of the gain edit box
set(handles.edit_l2,'String',value);

% Update simulation if the model is running
status = get_param(bdroot,'simulationstatus');
if strcmp(status,'running')
    set_param(bdroot, 'SimulationCommand', 'Update')
end
guidata(hObject,handles);

```

```

% --- Executes during object creation, after setting all properties.
function slider_l2_CreateFcn(hObject, eventdata, handles)
% hObject      handle to slider_l2 (see GCBO)
% eventdata    reserved - to be defined in a future version of MATLAB
% handles      empty - handles not created until after all CreateFcns called

% Hint: slider controls usually have a light gray background.
if isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor',[.9 .9 .9]);
end

function edit_p2_Callback(hObject, eventdata, handles)
% hObject      handle to edit_p2 (see GCBO)
% eventdata    reserved - to be defined in a future version of MATLAB
% handles      structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of edit_p2 as text
%        str2double(get(hObject,'String')) returns contents of edit_p2 as a
double
value = get(hObject,'String');

%valores l mites 2-0
if (str2double(value)>2.0)
    value=num2str(2.0);
end

if (str2double(value)< 0)
    value=num2str(0);
end

% Update the model's gain value
set_param([bdroot '/GainB'],'Gain',value)
set(handles.edit_p2,'String',value);

slider_position =str2double(value);
set(handles.slider_p2,'Value',slider_position);

% Update simulation if the model is running
status = get_param(bdroot,'simulationstatus');
if strcmp(status,'running')
    set_param(bdroot, 'SimulationCommand', 'Update')
end
guidata(hObject,handles);
% --- Executes during object creation, after setting all properties.
function edit_p2_CreateFcn(hObject, eventdata, handles)
% hObject      handle to edit_p2 (see GCBO)
% eventdata    reserved - to be defined in a future version of MATLAB
% handles      empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
%        See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))

```

```

        set(hObject, 'BackgroundColor', 'white');
    end

% --- Executes on slider movement.
function slider_p2_Callback(hObject, eventdata, handles)
% hObject    handle to slider_p2 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject, 'Value') returns position of slider
%         get(hObject, 'Min') and get(hObject, 'Max') to determine range of
slider
slider_position = get(hObject, 'Value');
value = num2str(slider_position);

% Update the model's gain value
set_param([bdroot '/GainB'], 'Gain', value)

% Set the value of the gain edit box
set(handles.edit_p2, 'String', value);

% Update simulation if the model is running
status = get_param(bdroot, 'simulationstatus');
if strcmp(status, 'running')
    set_param(bdroot, 'SimulationCommand', 'Update')
end
guidata(hObject, handles);

% --- Executes during object creation, after setting all properties.
function slider_p2_CreateFcn(hObject, eventdata, handles)
% hObject    handle to slider_p2 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

% Hint: slider controls usually have a light gray background.
if isequal(get(hObject, 'BackgroundColor'),
get(0, 'defaultUiControlBackgroundColor'))
    set(hObject, 'BackgroundColor', [.9 .9 .9]);
end

function edit_t200_Callback(hObject, eventdata, handles)
% hObject    handle to edit_t200 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject, 'String') returns contents of edit_t200 as text
%         str2double(get(hObject, 'String')) returns contents of edit_t200 as a
double
value = get(hObject, 'String');

%valores l mites 1.5-0.5
if (str2double(value)>1.5)
    value=num2str(1.5);
end

if (str2double(value)< 0.5)
    value=num2str(0.5);
end

% Update the model's gain value

```

```

set_param([bdroot '/GainH'],'Gain',value)
set(handles.edit_t200,'String',value);

slider_position =str2double(value);
set(handles.slider_t200,'Value',slider_position);

% Update simulation if the model is running
status = get_param(bdroot,'simulationstatus');
if strcmp(status,'running')
    set_param(bdroot, 'SimulationCommand', 'Update')
end
guidata(hObject,handles);

% --- Executes during object creation, after setting all properties.
function edit_t200_CreateFcn(hObject, eventdata, handles)
% hObject    handle to edit_t200 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
%         See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

% --- Executes on slider movement.
function slider_t200_Callback(hObject, eventdata, handles)
% hObject    handle to slider_t200 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'Value') returns position of slider
%         get(hObject,'Min') and get(hObject,'Max') to determine range of
slider
slider_position = get(hObject,'Value');
value = num2str(slider_position);

% Update the model's gain value
set_param([bdroot '/GainH'],'Gain',value)

% Set the value of the gain edit box
set(handles.edit_t200,'String',value);

% Update simulation if the model is running
status = get_param(bdroot,'simulationstatus');
if strcmp(status,'running')
    set_param(bdroot, 'SimulationCommand', 'Update')
end
guidata(hObject,handles);

% --- Executes during object creation, after setting all properties.
function slider_t200_CreateFcn(hObject, eventdata, handles)
% hObject    handle to slider_t200 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB

```

```

% handles    empty - handles not created until after all CreateFcns called

% Hint: slider controls usually have a light gray background.
if isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor',[.9 .9 .9]);
end

function edit_t1_Callback(hObject, eventdata, handles)
% hObject    handle to edit_t1 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of edit_t1 as text
%         str2double(get(hObject,'String')) returns contents of edit_t1 as a
double
value = get(hObject,'String');

%valores l mites 1.5-0.5
if (str2double(value)>1.5)
    value=num2str(1.5);
end

if (str2double(value)< 0.5)
    value=num2str(0.5);
end

% Update the model's gain value
set_param([bdroot '/GainG'],'Gain',value)
set(handles.edit_t1,'String',value);

slider_position =str2double(value);
set(handles.slider_t1,'Value',slider_position);

% Update simulation if the model is running
status = get_param(bdroot,'simulationstatus');
if strcmp(status,'running')
    set_param(bdroot, 'SimulationCommand', 'Update')
end
guidata(hObject,handles);

% --- Executes during object creation, after setting all properties.
function edit_t1_CreateFcn(hObject, eventdata, handles)
% hObject    handle to edit_t1 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
%         See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

% --- Executes on slider movement.
function slider_t1_Callback(hObject, eventdata, handles)
% hObject    handle to slider_t1 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
% Hints: get(hObject,'Value') returns position of slider

```

```

%         get(hObject,'Min') and get(hObject,'Max') to determine range of
slider
slider_position = get(hObject,'Value');
value = num2str(slider_position);

% Update the model's gain value
set_param([bdroot '/GainG'],'Gain',value)

% Set the value of the gain edit box
set(handles.edit_t1,'String',value);

% Update simulation if the model is running
status = get_param(bdroot,'simulationstatus');
if strcmp(status,'running')
    set_param(bdroot,'SimulationCommand','Update')
end
guidata(hObject,handles);

% --- Executes during object creation, after setting all properties.
function slider_t1_CreateFcn(hObject, eventdata, handles)
% hObject    handle to slider_t1 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

% Hint: slider controls usually have a light gray background.
if isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor',[.9 .9 .9]);
end

function edit_x1_Callback(hObject, eventdata, handles)
% hObject    handle to edit_x1 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of edit_x1 as text
%        str2double(get(hObject,'String')) returns contents of edit_x1 as a
double
value = get(hObject,'String');

%valores lîmites 2-0
if (str2double(value)>2)
    value=num2str(2);
end

if (str2double(value)< 0)
    value=num2str(0);
end

% Update the model's gain value
set_param([bdroot '/GainF'],'Gain',value)
set(handles.edit_x1,'String',value);

slider_position =str2double(value);
set(handles.slider_x1,'Value',slider_position);

```

```

% Update simulation if the model is running
status = get_param(bdroot, 'simulationstatus');
if strcmp(status, 'running')
    set_param(bdroot, 'SimulationCommand', 'Update')
end
guidata(hObject, handles);

% --- Executes during object creation, after setting all properties.
function edit_x1_CreateFcn(hObject, eventdata, handles)
% hObject    handle to edit_x1 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
%         See ISPC and COMPUTER.
if ispc && isequal(get(hObject, 'BackgroundColor'),
get(0, 'defaultUicontrolBackgroundColor'))
    set(hObject, 'BackgroundColor', 'white');
end

% --- Executes on slider movement.
function slider_x1_Callback(hObject, eventdata, handles)
% hObject    handle to slider_x1 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject, 'Value') returns position of slider
%         get(hObject, 'Min') and get(hObject, 'Max') to determine range of
slider
slider_position = get(hObject, 'Value');
value = num2str(slider_position);

% Update the model's gain value
set_param([bdroot '/GainF'], 'Gain', value)

% Set the value of the gain edit box
set(handles.edit_x1, 'String', value);

% Update simulation if the model is running
status = get_param(bdroot, 'simulationstatus');
if strcmp(status, 'running')
    set_param(bdroot, 'SimulationCommand', 'Update')
end
guidata(hObject, handles);

% --- Executes during object creation, after setting all properties.
function slider_x1_CreateFcn(hObject, eventdata, handles)
% hObject    handle to slider_x1 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

% Hint: slider controls usually have a light gray background.
if isequal(get(hObject, 'BackgroundColor'),
get(0, 'defaultUicontrolBackgroundColor'))
    set(hObject, 'BackgroundColor', [.9 .9 .9]);
end

function edit_f1_Callback(hObject, eventdata, handles)
% hObject    handle to edit_f1 (see GCBO)

```

```

% eventdata reserved - to be defined in a future version of MATLAB
% handles structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of edit_f1 as text
% str2double(get(hObject,'String')) returns contents of edit_f1 as a
double
value = get(hObject,'String');

%valores límites 1.5-0.7
if (str2double(value)>1.5)
    value=num2str(1.5);
end

if (str2double(value)< 0.7)
    value=num2str(0.7);
end

% Update the model's gain value
set_param([bdroot '/GainE'],'Gain',value)
set(handles.edit_f1,'String',value);

slider_position =str2double(value);
set(handles.slider_f1,'Value',slider_position);

% Update simulation if the model is running
status = get_param(bdroot,'simulationstatus');
if strcmp(status,'running')
    set_param(bdroot, 'SimulationCommand', 'Update')
end
guidata(hObject,handles);

% --- Executes during object creation, after setting all properties.
function edit_f1_CreateFcn(hObject, eventdata, handles)
% hObject handle to edit_f1 (see GCBO)
% eventdata reserved - to be defined in a future version of MATLAB
% handles empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
% See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

% --- Executes on slider movement.
function slider_f1_Callback(hObject, eventdata, handles)
% hObject handle to slider_f1 (see GCBO)
% eventdata reserved - to be defined in a future version of MATLAB
% handles structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'Value') returns position of slider
% get(hObject,'Min') and get(hObject,'Max') to determine range of
slider
slider_position = get(hObject,'Value');
value = num2str(slider_position);

```

```

% Update the model's gain value
set_param([bdroot '/GainE'],'Gain',value)

% Set the value of the gain edit box
set(handles.edit_f1,'String',value);

% Update simulation if the model is running
status = get_param(bdroot,'simulationstatus');
if strcmp(status,'running')
    set_param(bdroot, 'SimulationCommand', 'Update')
end
guidata(hObject,handles);

% --- Executes during object creation, after setting all properties.
function slider_f1_CreateFcn(hObject, eventdata, handles)
% hObject    handle to slider_f1 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

% Hint: slider controls usually have a light gray background.
if isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor',[.9 .9 .9]);
end

function edit_p0_Callback(hObject, eventdata, handles)
% hObject    handle to edit_p0 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of edit_p0 as text
%        str2double(get(hObject,'String')) returns contents of edit_p0 as a
double
value = get(hObject,'String');

%valores l mites 1.5-0.5
if (str2double(value)>1.5)
    value=num2str(1.5);
end

if (str2double(value)< 0.5)
    value=num2str(0.5);
end

% Update the model's gain value
set_param([bdroot '/GainD'],'Gain',value)
set(handles.edit_p0,'String',value);

slider_position =str2double(value);
set(handles.slider_p0,'Value',slider_position);

% Update simulation if the model is running
status = get_param(bdroot,'simulationstatus');
if strcmp(status,'running')
    set_param(bdroot, 'SimulationCommand', 'Update')
end
guidata(hObject,handles);

% --- Executes during object creation, after setting all properties.

```

```

function edit_p0_CreateFcn(hObject, eventdata, handles)
% hObject      handle to edit_p0 (see GCBO)
% eventdata    reserved - to be defined in a future version of MATLAB
% handles      empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
%         See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

% --- Executes on slider movement.
function slider_p0_Callback(hObject, eventdata, handles)
% hObject      handle to slider_p0 (see GCBO)
% eventdata    reserved - to be defined in a future version of MATLAB
% handles      structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'Value') returns position of slider
%         get(hObject,'Min') and get(hObject,'Max') to determine range of
slider
slider_position = get(hObject,'Value');
value = num2str(slider_position);

% Update the model's gain value
set_param([bdroot '/GainD'],'Gain',value)

% Set the value of the gain edit box
set(handles.edit_p0,'String',value);

% Update simulation if the model is running
status = get_param(bdroot,'simulationstatus');
if strcmp(status,'running')
    set_param(bdroot, 'SimulationCommand', 'Update')
end
guidata(hObject,handles);

% --- Executes during object creation, after setting all properties.
function slider_p0_CreateFcn(hObject, eventdata, handles)
% hObject      handle to slider_p0 (see GCBO)
% eventdata    reserved - to be defined in a future version of MATLAB
% handles      empty - handles not created until after all CreateFcns called

% Hint: slider controls usually have a light gray background.
if isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor',[.9 .9 .9]);
end

function edit_x2_Callback(hObject, eventdata, handles)
% hObject      handle to edit_Gain (see GCBO)
% eventdata    reserved - to be defined in a future version of MATLAB
% handles      structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of edit_x2 as text

```

```

%      str2double(get(hObject,'String')) returns contents of edit_x2 as a
double

value = get(hObject,'String');

%valores límites 1.6-0
if (str2double(value)>1.6)
    value=num2str(1.6);
end

if (str2double(value)< 0)
    value=num2str(0);
end

% Update the model's gain value
set_param([bdroot '/GainA'],'Gain',value)
set(handles.edit_x2,'String',value);

slider_position =str2double(value);
set(handles.slider_x2,'Value',slider_position);

% Update simulation if the model is running
status = get_param(bdroot,'simulationstatus');
if strcmp(status,'running')
    set_param(bdroot, 'SimulationCommand', 'Update')
end
guidata(hObject,handles);

% --- Executes during object creation, after setting all properties.
function edit_x2_CreateFcn(hObject, eventdata, handles)
% hObject    handle to edit_x2 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
%       See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

% --- Executes on slider movement.
function slider_x2_Callback(hObject, eventdata, handles)
% hObject    handle to slider_x2 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'Value') returns position of slider
%       get(hObject,'Min') and get(hObject,'Max') to determine range of
slider
slider_position = get(hObject,'Value');
value = num2str(slider_position);

% Update the model's gain value
set_param([bdroot '/GainA'],'Gain',value)
% Set the value of the gain edit box
set(handles.edit_x2,'String',value);
% Update simulation if the model is running
status = get_param(bdroot,'simulationstatus');
if strcmp(status,'running')

```

```

    set_param(bdroot, 'SimulationCommand', 'Update')
end
guidata(hObject,handles);
% --- Executes during object creation, after setting all properties.
function slider_x2_CreateFcn(hObject, eventdata, handles)
% hObject    handle to slider_x2 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

% Hint: slider controls usually have a light gray background.
if isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor',[.9 .9 .9]);
end
function edit_slx_Callback(hObject, eventdata, handles)
% hObject    handle to edit_slx (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

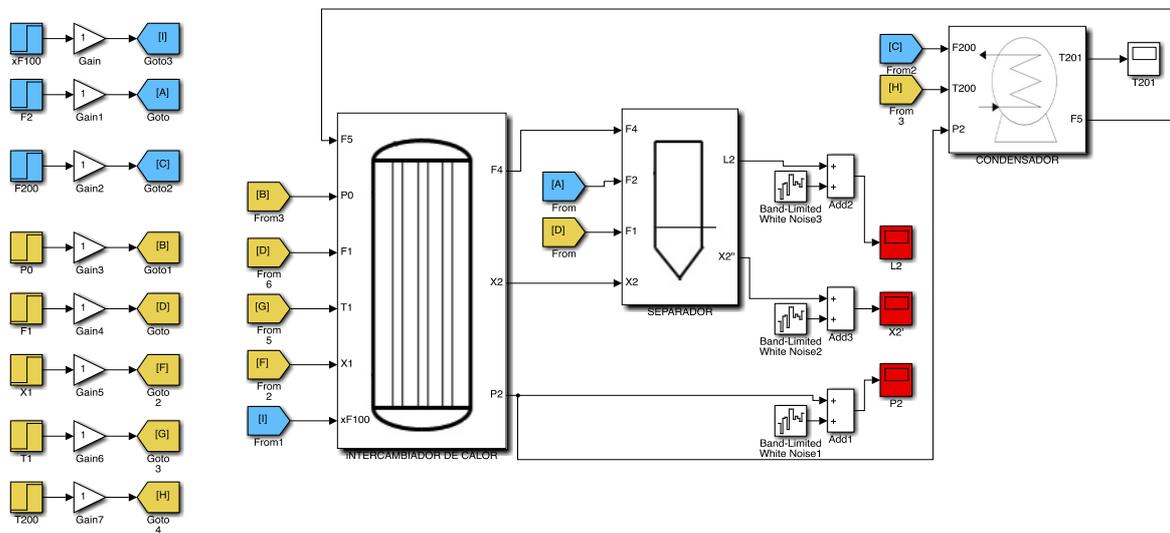
% --- Executes during object creation, after setting all properties.
function edit_slx_CreateFcn(hObject, eventdata, handles)
% hObject    handle to edit_slx (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
% See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

```

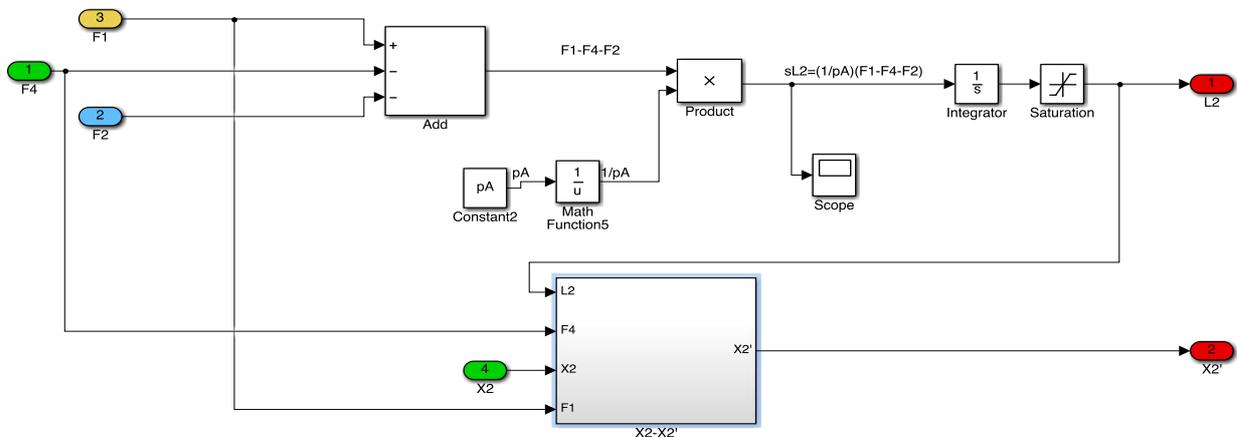
Apéndice B. Diagramas Simulink

B.1 MODELO NO LINEAL

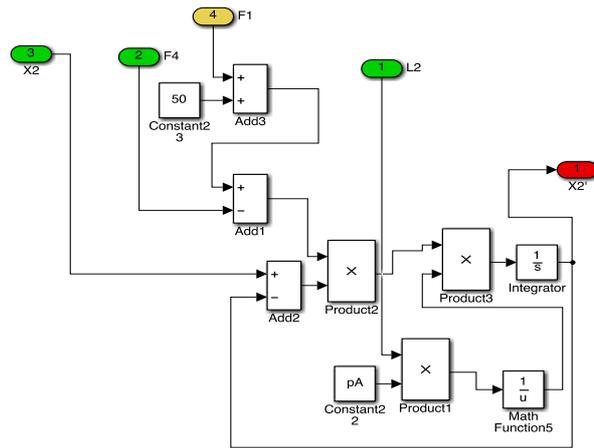


Vista general. Nivel 0

B.1.1 Separador

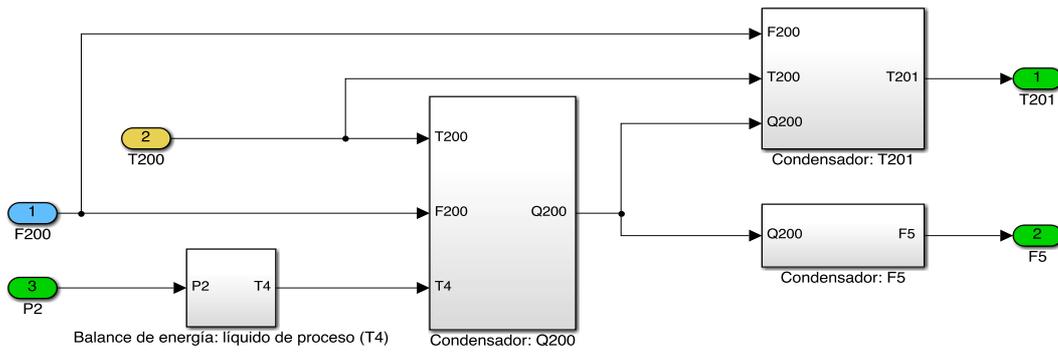


Separador. Nivel 1

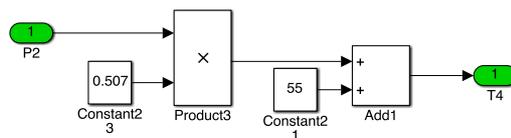


Separador. Nivel 2 (X_2-X_2')

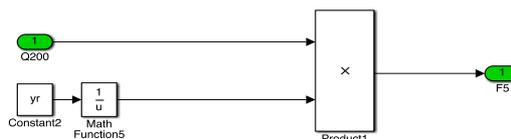
B.1.2 Condensador



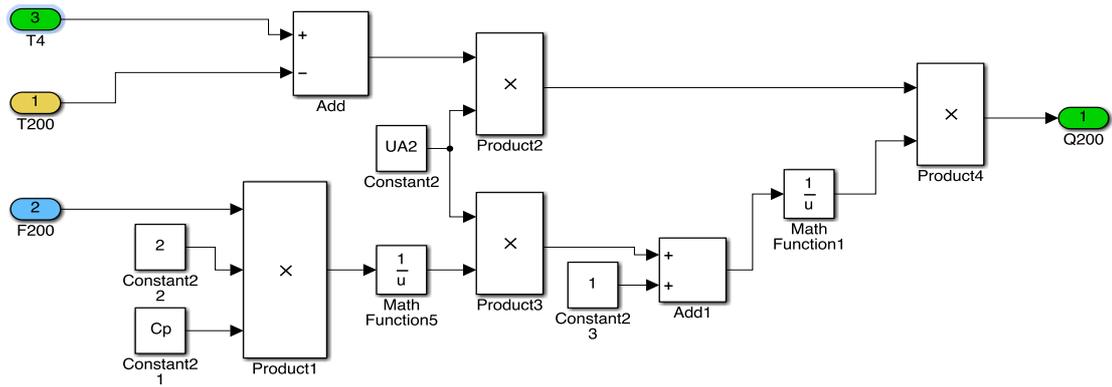
Condensador. Nivel 1



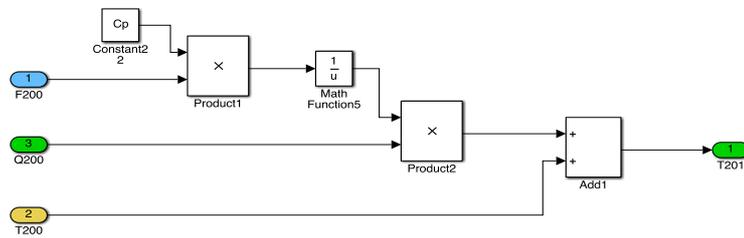
Condensador. Nivel 2 (Balance de energía: líquido de proceso-T4)



Condensador. Nivel 2 (F_5)

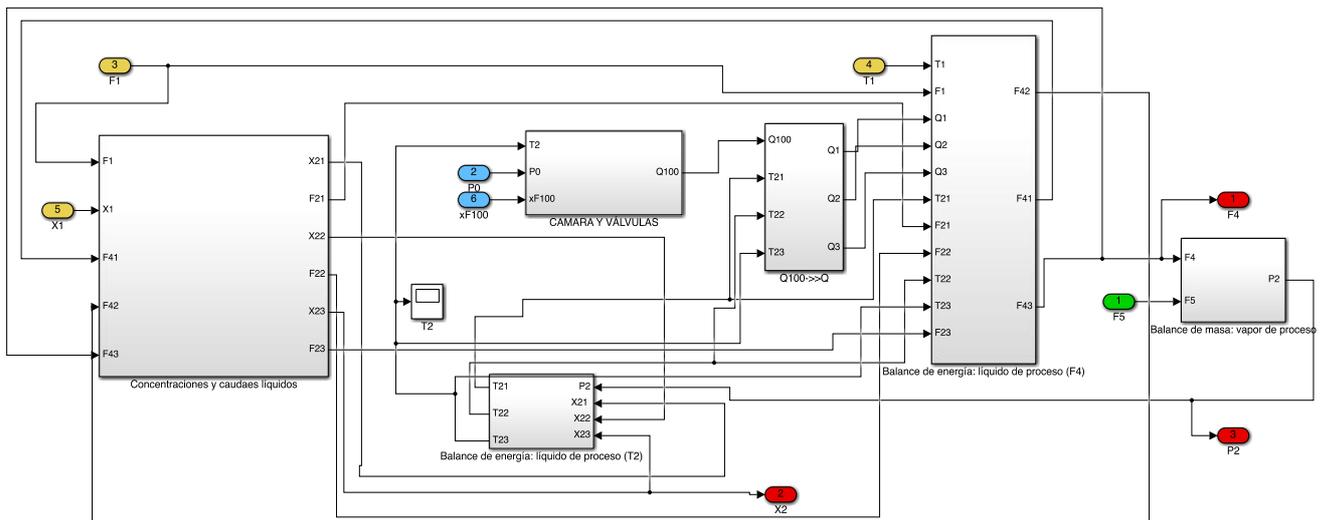


Condensador. Nivel 2 (Q₂₀₀)

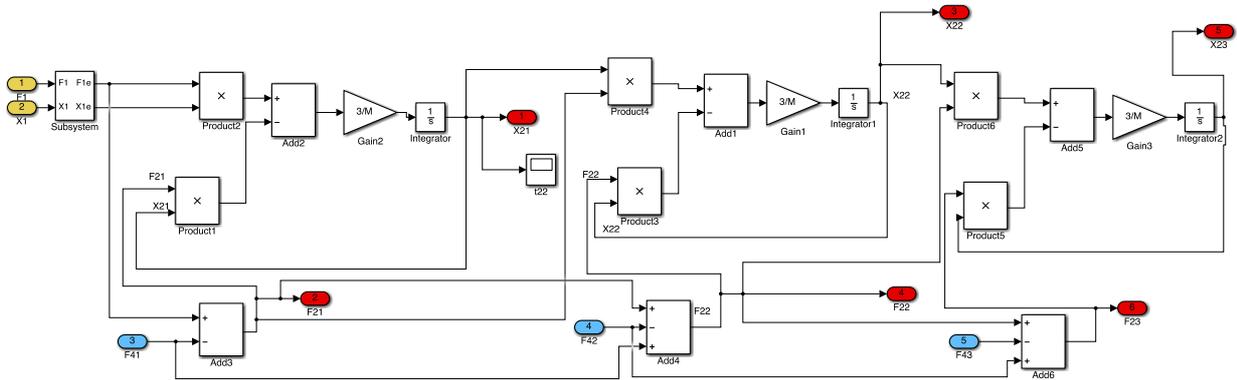


Condensador. Nivel 2 (T₂₀₁)

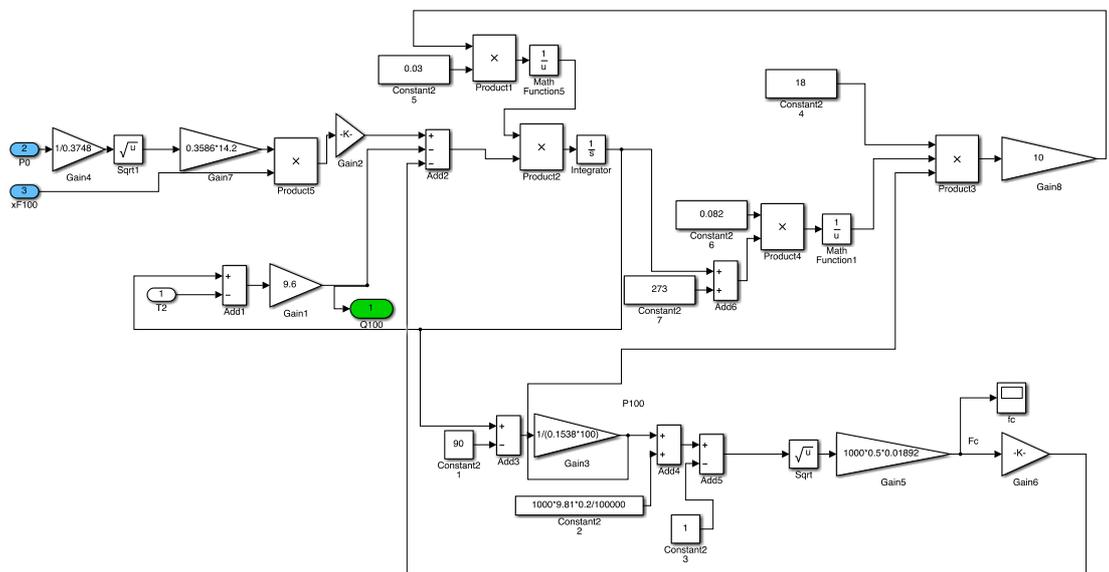
B.1.3 Evaporador



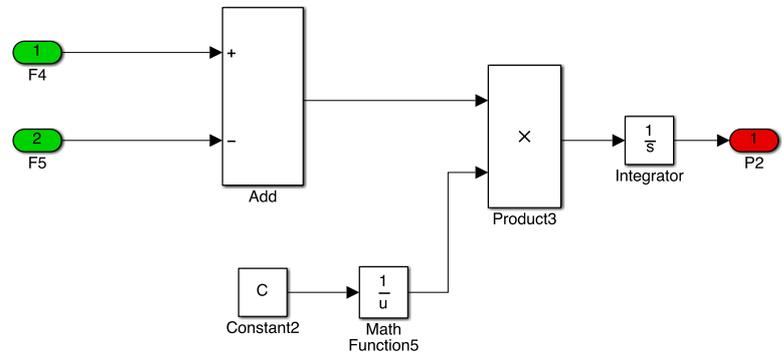
Evaporador. Nivel 1



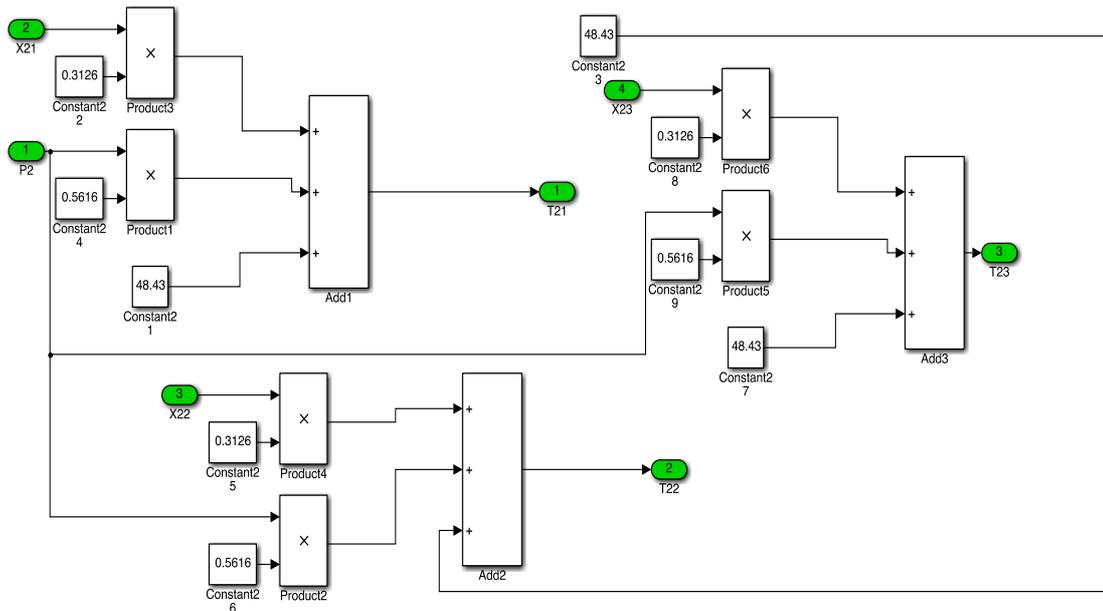
Evaporador. Nivel 2 (Concentraciones y caudales líquidos)



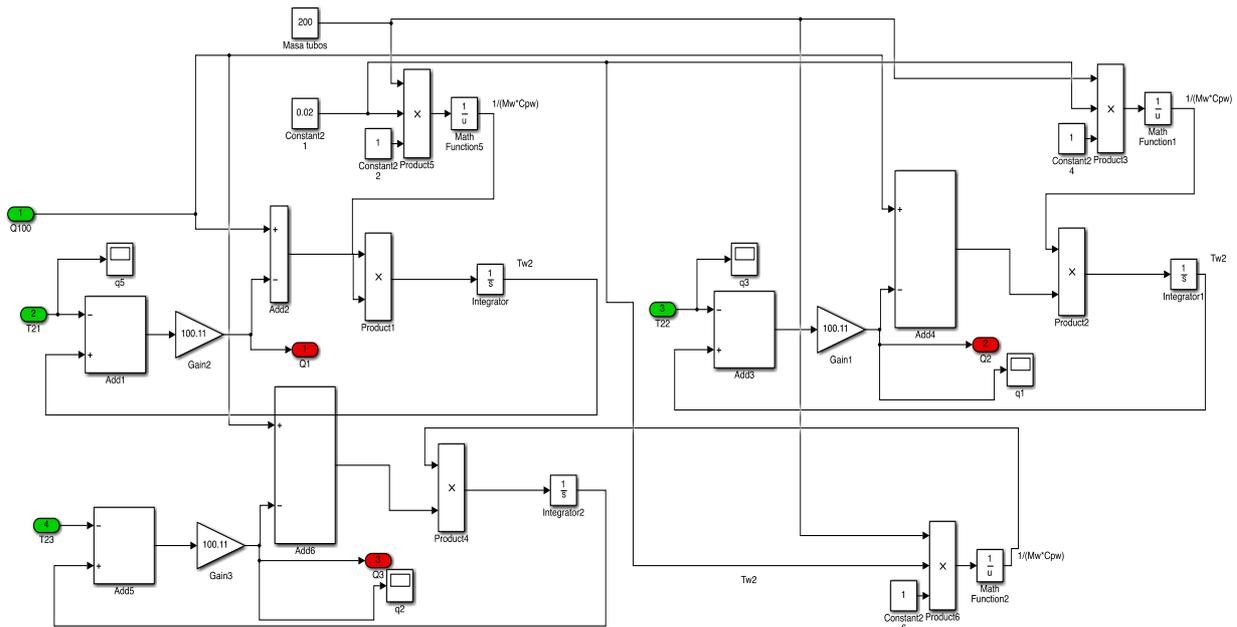
Intercambiador de Calor. Nivel 2 (Cámara y válvulas)



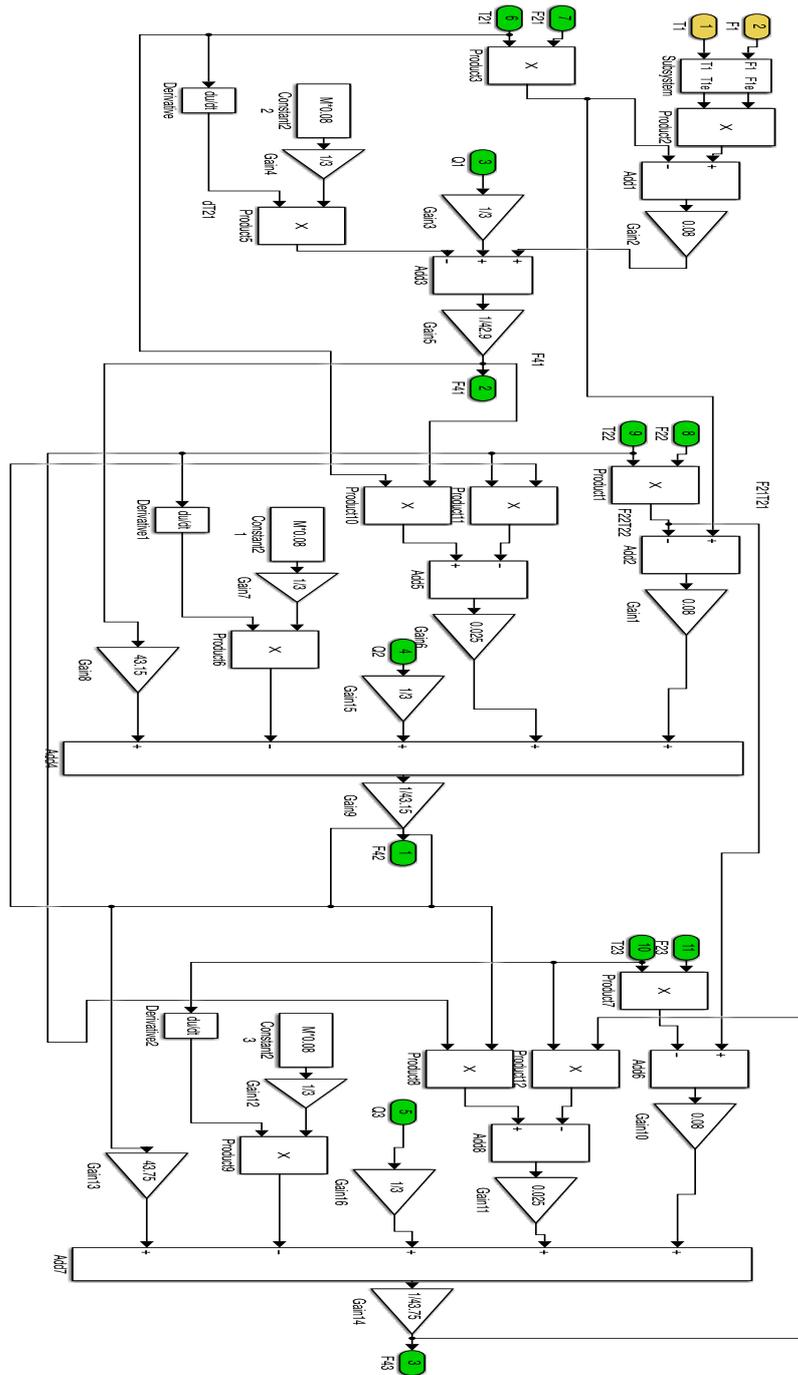
Evaporador. Nivel 2 (Balance de masa, vapor de proceso)



Evaporador. Nivel 2 (Líquido de proceso-temperatura T₂)



Evaporador. Nivel 2 (Q₁₀₀ -> Q)



Evaporador. Nivel 2 (Balance de energía, líquido de proceso F₄)

B.2 MODELO NO LINEAL. CONTROL SISO COMPLETO

La siguiente imagen muestra el control completo del sistema. Para ello, se crea un subsistema de todo el “Nivel 0” visto al inicio de este apéndice, y se añaden los controles prealimentados, realimentados y en cascada.

Índice de figuras

Figura 1.1 Esquema de funcionamiento Evaporado	2
Figura 2.1 Esquema subdivisión el evaporador	6
Figura 2.2 Sección longitudinal interior de una de las mitades simétricas del intercambiador de calor	7
Figura 2.3 Variables separador, bomba y producto	9
Figura 4.1 Pantalla principal modelo no lineal Simulink	19
Figura 4.2 Ejemplo de respuesta de simulación	20
Figura 4.3 Respuestas variables de control frente a variación en X_{F100}	21
Figura 4.4 Respuestas variables de control frente a variación en F_{200}	21
Figura 4.5 Respuestas variables de control frente a variación en F_2	22
Figura 4.6 Respuestas variables de control frente a variación en P_0	22
Figura 4.7 Respuestas variables de control frente a variación en F_1	23
Figura 4.8 Respuestas variables de control frente a variación en X_1	23
Figura 4.9 Respuestas variables de control frente a variación en T_1	24
Figura 4.10 Respuestas variables de control frente a variación en T_{200}	25
Figura 5.1 respuesta en scope-Simulink	28
Figura 5.2 Pantalla de estimación de modelo Ident	29
Figura 5.3 test ajuste respuesta	29
Figura 5.4 Respuesta sistema no lineal-lineal. X_2' frente a escalón amplitud $\pm 10\%$ en X_{F100} , y ajuste	31
Figura 5.5 Respuesta sistema no lineal-lineal. P_2 frente a escalón amplitud $\pm 10\%$ en X_{F100} , y ajuste	31
Figura 5.6 Respuesta sistema no lineal-lineal. L_2 frente a escalón amplitud $\pm 10\%$ en X_{F100} , y ajuste	31
Figura 5.7 Respuesta sistema no lineal-lineal. X_2' frente a escalón amplitud $\pm 10\%$ en F_{200} , y ajuste	32
Figura 5.8 Respuesta sistema no lineal-lineal. P_2 frente a escalón amplitud $\pm 10\%$ en F_{200} , y ajuste	32
Figura 5.9 Respuesta sistema no lineal-lineal. L_2 frente a escalón amplitud $\pm 10\%$ en F_{200} , y ajuste	32
Figura 5.10 Respuesta sistema no lineal-lineal. L_2 frente a escalón amplitud $\pm 10\%$ en F_2 , y ajuste	33
Figura 5.11 Respuesta sistema no lineal-lineal. X_2' frente a escalón amplitud $\pm 10\%$ en P_0 , y ajuste	34
Figura 5.12 Respuesta sistema no lineal-lineal. P_2 frente a escalón amplitud $\pm 10\%$ en P_0 , y ajuste	35
Figura 5.13 Respuesta sistema no lineal-lineal. L_2 frente a escalón amplitud $\pm 10\%$ en P_0 , y ajuste	35
Figura 5.14 Respuesta sistema no lineal-lineal. X_2' frente a escalón amplitud $\pm 10\%$ en F_1 , y ajuste	35
Figura 5.15 Respuesta sistema no lineal-lineal. P_2 frente a escalón amplitud $\pm 10\%$ en F_1 , y ajuste	36
Figura 5.16 Respuesta sistema no lineal-lineal. L_2 frente a escalón amplitud $\pm 10\%$ en F_1 , y ajuste	36
Figura 5.17 Respuesta sistema no lineal-lineal. X_2' frente a escalón amplitud $\pm 10\%$ en X_1 , y ajuste	36

Figura 5.18 Respuesta sistema no lineal-lineal. P_2 frente a escalón amplitud $\pm 10\%$ en X_1 , y ajuste	37
Figura 5.19 Respuesta sistema no lineal-lineal. L_2 frente a escalón amplitud $\pm 10\%$ en X_1 , y ajuste	37
Figura 5.20 Respuesta sistema no lineal-lineal. X_2' frente a escalón amplitud $\pm 10\%$ en T_1 , y ajuste	37
Figura 5.21 Respuesta sistema no lineal-lineal. P_2 frente a escalón amplitud $\pm 10\%$ en T_1 , y ajuste	38
Figura 5.22 Respuesta sistema no lineal-lineal. L_2 frente a escalón amplitud $\pm 10\%$ en T_1 , y ajuste	38
Figura 5.23 Respuesta sistema no lineal-lineal. X_2' frente a escalón amplitud $\pm 10\%$ en T_{200} , y ajuste	38
Figura 5.24 Respuesta sistema no lineal-lineal. P_2 frente a escalón amplitud $\pm 10\%$ en T_{200} , y ajuste	39
Figura 5.25 Respuesta sistema no lineal-lineal. L_2 frente a escalón amplitud $\pm 10\%$ en T_{200} , y ajuste	39
Figura 5.26 Modelo de control realimentado	41
Figura 5.27 Diagrama de bloques Simulink X2-xF100	45
Figura 5.28 Esquema Simulink PID	45
Figura 5.29 Respuesta de X_2' y $xF_{100}\%$ frente a cambios en la referencia para PI y PID	46
Figura 5.30 Respuesta de X_2' y $xF_{100}\%$ frente a cambios en la referencia para PID final	47
Figura 5.31 Control realimentado X_2' - X_{F100} frente a perturbaciones	48
Figura 5.32 Respuesta de X_2' y $xF_{100}\%$ frente a cambios en la referencia con diferentes filtros PID	49
Figura 5.33 Respuesta de $P_2\%$ y $F_{200}\%$ frente a cambios en la referencia con PI	51
Figura 5.35 Respuesta de $P_2\%$ y $F_{200}\%$ con $k_c = -100$, con y sin antiwindup	52
Figura 5.36 Control realimentado P_2 - F_{200} frente a perturbaciones	52
Figura 5.37 Respuesta de $L_2\%$ y $F_2\%$ frente a la perturbación F_1 con controlador PI y P	54
Figura 5.39 Control por acción anticipativa o feedforward	55
Figura 5.40 Control en cascada	56
Figura 5.41 respuesta X_2' y $xF_{100}\%$ frente a escalón en F_1 , con y sin control anticipativo	57
Figura 5.42 Salida $P_2\%$ y entrada $F_{200}\%$ frente a escalón en T_{200} , sin y con control anticipativo	58
Figura 5.43 Salida $L_2\%$ y entrada F_2' frente a escalón en F_1 , sin y con acción anticipativa	59
Figura 5.44 Salida $L_2\%$ y entrada F_2' frente a escalón en F_1 , con FF y resto de bucles abiertos	60
Figura 5.45 Salida $L_2\%$ y entrada F_2' frente a escalón en F_1 , sin y con control anticipativo 2	60
Figura 5.46 Esquema de Simulink para el control en cascada del emparejamiento xF_{100} - X_2'	61
Figura 5.47 : salida X_2' y entrada $xF_{100}\%$ frente a perturbación P_0 sin y con control en cascada	63
Figura 5.48 Control en cascada + acción anticipativa	63
Figura 5.49 respuesta salidas y entradas del sistema SISO completo frente a cadena de perturbaciones sin control anticipativo-cascada y con él	65
Figura 5.50 respuesta salidas y entradas del sistema SISO completo frente a cadena de referencias sin control anticipativo-cascada y con él	66
Figura 5.51 Respuestas X_2 frente a +10 KPa en P_{100} y P_2 frente a +10 kg/min en F_{200} . Modelo Newell and Lee	68
Figura 6.1 Pantalla GUI	69

Bibliografía

- [1] *Anibal Alberto Bizama Soto Blog*. (2012). Recuperado el 13 de Julio de 2016, de <http://anibalbizama.blogspot.com.es/2012/11/9-sistema-de-control-de-procesos.html>
- [2] Cao, Y. (2010). *CONSTRAINED SELF-OPTIMIZING CONTROL VIA DIFFERENTIATION*. Cranfield, UK.
- [3] Dittmar, R. (2015). Decentralized SISO Active Disturbance Rejection Control of the Newell-Lee forced circulation evaporator. Heide, Alemania.
- [4] Fernández Benítez, J., & Corrochano Sánchez, C. (2012). *Cuadernos de Transmisión de Calor*. Madrid: Sección de Publicaciones ETSII-UPM.
- [5] *imm*. (s.f.). Recuperado el 13 de Julio de 2016, de <http://www.iim.unsj.edu.ar/control/>
- [6] Newell, R., & Lee, P. (1989). *Applied Process Control: A Case Study*. Brisbane, Australia: Prentice Hall.
- [7] Ollero de Castro, P., & Fernández Camacho, E. (2012). *Instrumentación y control de plantas químicas*. Sevilla: Síntesis.
- [8] Samson. (2012). *Application Notes for Valves Sizing. Sizing examples*. Frankfurt Am main.
- [9] Sastrón, F. (2013). *Apuntes Ingeniería de Control, ETSII-UPM*. Madrid.
- [10] Skogestad, S., & Postlethwaite, I. (2001). *Multivariable Feedback Control, Analysis and Design*. John Wiley and Sons.
- [11] Switalski, S. (1999). Control system with the specific performance index for an evaporator. Gliwice, Polonia.
- [12] Wuhua Hu, Wen-Jian Cai, & Gaoxi Xiao. (2010). *Relative Gain Array for MIMO Processes Containing Integrators and/or Differentiators*. Singapore.
- [13] *Gautam Vallabha*. Real-Time pacer for Simulink. Mathworks. Recuperado el 10 de Julio de 2016, de www.mathworks.com