

DEPARTAMENTO DE INGENIERÍA DE
SISTEMAS Y AUTOMÁTICA

UNIVERSIDAD DE VALLADOLID

TRABAJO DE MÁSTER

**Programación de controlador predictivo en
EcosimPro® e implementación en planta de
distribución de oxígeno**

MÁSTER EN INVESTIGACIÓN EN
INGENIERÍA DE PROCESOS Y SISTEMAS

Directores:

César de Prada Moraga

Rubén Martí Martínez

Colaborador:

Daniel Navia López

JUAN DAVID TEJERINA GUTIÉRREZ

JULIO 2013

CÉSAR DE PRADA MORAGA, RUBÉN MARTÍ MARTÍNEZ y DANIEL NAVIA LÓPEZ, del Departamento de Ingeniería de Sistemas y Automática de la Universidad de Valladolid, CERTIFICAN:

Que D. JUAN DAVID TEJERINA GUTIÉRREZ ha realizado bajo nuestra dirección el Trabajo Fin de Máster titulado *“Programación de controlador predictivo en EcosimPro® e implementación en planta de distribución de oxígeno”*.

Valladolid, 9 de julio de 2013

Fdo.: César de Prada Moraga

Fdo.: Rubén Martí Martínez

Reunido el Tribunal designado por el Departamento de Ingeniería de Sistemas y Automática para la evaluación del Trabajo de Investigación del Máster Oficial en Investigación en Ingeniería de Procesos y Sistemas, y después de estudiar la Memoria y atender a la defensa del trabajo *“Programación de controlador predictivo en EcosimPro® e implementación en planta de distribución de oxígeno”*, presentado por el alumno D. JUAN DAVID TEJERINA GUTIÉRREZ, decidió otorgarle la calificación de _____.

Valladolid, 9 de julio de 2013

El presidente

El secretario

Fdo.:

Fdo.:

Vocal

Fdo.:

ÍNDICE

1. RESUMEN	3
2. OBJETIVOS	4
3. INTRODUCCIÓN	5
3.1. DISTRIBUCIÓN DE HIDRÓGENO EN LAS REFINERÍAS	6
4. DESCRIPCIÓN DE LA RED DE DISTRIBUCIÓN DE OXÍGENO	8
4.1. CONDICIONES DE OPERACIÓN	12
5. CONCEPTOS BÁSICOS SOBRE TRANSFERENCIA DE MATERIA G-L	14
6. IDENTIFICACIÓN BASADA EN UN MODELO DE CONOCIMIENTO	18
6.1. IDENTIFICACIÓN DE $k_L a$	22
6.2. VALIDACIÓN DEL MODELO	26
6.3. SIMULACIÓN DEL MODELO EN ECOSIM	29
7. CONTROLADOR PREDICTIVO	34
7.1. PROGRAMACIÓN DEL CONTROLADOR PREDICTIVO	37
7.2. SIMULACIÓN DEL CONTROLADOR EN ECOSIM	49
7.3. MOVING HORIZON ESTIMATION	55
7.4. SINTONÍA DEL CONTROLADOR	64
8. SIMULACIÓN EN EL SCADA EDUSCA	67
8.1. CREACIÓN DEL SERVIDOR OPC DE LA SIMULACIÓN DE LA PLANTA	67
8.2. CONFIGURACIÓN DE LA SIMULACIÓN EN EL SACADA	68
8.3. CREACIÓN DEL SERVIDOR OPC DEL CONTROL DE LA PLANTA	71
8.4. CONFIGURACIÓN DEL CONTROLADOR DE LA PLANTA EN EL SCADA	72
8.5. ESQUEMA DE COMUNICACIONES	77
9. APLICACIÓN DEL CONTROLADOR PREDICTIVO A LA PLANTA	78
9.1. CONFIGURACIÓN DEL SCADA	78
9.2. ESQUEMA DE COMUNICACIONES	79
9.3. EXPERIMENTOS DE CONTROL APLICADO A LA PLANTA	80
10. CONCLUSIONES Y TRABAJO FUTURO	87
11. ANEXO I - CÓDIGOS DE ECOSIMPRO	88
12. BIBLIOGRAFÍA	114

1. RESUMEN

El presente proyecto presenta la aplicación de un controlador predictivo no lineal (NMPC), programado en EcosimPro®, a la planta de distribución de oxígeno instalada en el Laboratorio del Departamento de Ingeniería de Sistemas y Automática (El Sede Dr. Mergelina, Universidad de Valladolid), que simula el comportamiento de una red de distribución de hidrógeno en una refinería.

La planta, cuya descripción detallada se muestra en el apartado 4, consta de dos generadores de oxígeno que aportan flujos de gas con distinta pureza. Se dispone de dos colectores para la circulación de sendos flujos con tres ramas de aporte por cada colector a tres unidades consumidoras de oxígeno. Dichas unidades son depósitos de tanque agitado por los que circula un flujo continuo de agua impulsado por bombas peristálticas, de tal modo que el consumo de oxígeno está basado en el fenómeno de difusión gas-líquido.

Las variables medidas en cada unidad son la concentración de oxígeno disuelto (variable controlada) y la temperatura (perturbación medible), siendo las referencias de los controladores PID de flujo de gas en las ramas de aporte a cada unidad las variables manipuladas en el problema de control.

Inicialmente, se crea una librería en EcosimPro con la simulación del modelo, previamente identificado y validado, y la programación del controlador predictivo, tomando como suposición inicial que dicho modelo es idéntico al proceso, y posteriormente introduciendo desviaciones en los parámetros para estudiar la corrección de los errores de modelado.

Para ello, se plantea un problema adicional de minimización en el que la función objetivo es el sumatorio del cuadrado de las diferencias entre los valores de la variable predicha por el modelo y los valores reales medidos del proceso, a lo largo de un horizonte móvil, teniendo como variable de decisión un parámetro de perturbación no medible que modifica el modelo para ajustarlo a los valores reales (*Moving Horizon Estimation, MHE*).

Una vez programado el control predictivo, se establece el esquema de comunicaciones necesario para su implantación a la planta en tiempo real vía OPC, gracias a la posibilidad de crear servidores OPC a partir de los códigos de EcosimPro y configurando el intercambio de variables en el SCADA EDUSCA (*Educational Scada*), que permite la interacción del usuario con el proceso.

En resumen, este proyecto presenta una nueva funcionalidad de EcosimPro, que además de simular el comportamiento de un modelo, es capaz de llevar a cabo el control y la supervisión de un proceso real mediante la creación de servidores OPC a partir de los códigos programados.

2. OBJETIVOS

El objetivo del proyecto es la implantación del control predictivo a la planta de distribución de oxígeno, que simula el comportamiento de una red de distribución de hidrógeno en una refinería. De este modo, las unidades consumidoras de oxígeno de la planta simulan reactores de hidrosulfuración donde el hidrógeno reacciona con el azufre en presencia de un catalizador, obteniéndose como producto final un hidrocarburo desulfurado (en la planta piloto en lugar de hidrógeno circula oxígeno, cuyo consumo está basado en la difusión de oxígeno en agua).

Para aplicar el control predictivo, en primer lugar se identifica un modelo basado en primeros principios. Una vez validado el modelo, se programa el controlador predictivo en el entorno de EcosimPro® 4.8 [1]. Dicho entorno dispone de un lenguaje moderno de modelado orientado a objetos (OOML) llamado EL (EcosimPro Language).

Finalmente, es necesario establecer el sistema de comunicación con la planta para enviar y recibir datos en tiempo real.

De forma secuencial, los objetivos a cumplir para el desarrollo del proyecto son:

- ✓ Identificación y validación del modelo.
- ✓ Programación del controlador predictivo en EcosimPro.
- ✓ Selección de los parámetros de sintonía y simulación del MPC.
- ✓ Definición del esquema de comunicaciones con la planta.
- ✓ Configuración del SCADA y simulación del MPC a partir de los servidores OPC de la planta y del controlador, creados a partir de los códigos de Ecosim.
- ✓ Aplicación del MPC a la planta reemplazando el servidor del modelo por el servidor de la planta real.

3. INTRODUCCIÓN

El presente proyecto se enmarca dentro de un proyecto de investigación más amplio, cuyo objetivo es desarrollar nuevos conocimientos en el campo del control y optimización de sistemas de gran escala en la industria de procesos y demostrar su aplicación a escala industrial [2].

El interés por este tipo de sistemas ha ido creciendo progresivamente por su impacto en el funcionamiento global de un sistema o factoría, tanto económico como técnico, al igual que por el papel estratégico que a menudo desempeña [3]. Entre ellos se encuentran las redes de distribución: de agua, electricidad, gas, telecomunicaciones, etc., de cuya gestión eficaz pueden depender tanto la satisfacción de las demandas de los usuarios como la rentabilidad de su operación.

En particular, las redes de distribución de vapor, gas, tratamiento de aguas, etc. de muchas industrias de procesos enlazan los centros de producción o almacenaje con las plantas donde se consumen dichos recursos a través de un sistema de distribución, cuyas características habituales son:

- Gran escala
- Interacción entre los distintos elementos
- Variabilidad de las demandas
- Presencia de diversas restricciones de operación
- Control del funcionamiento a través de decisiones de diferente naturaleza. Estas decisiones pueden ser:
 - De tipo continuo (valor de apertura de las válvulas)
 - De tipo discreto (encendido o apagado de compresores)

Toda esta complejidad hace que el control y la operación global tengan lugar muy a menudo en condiciones no óptimas tanto desde el punto de vista económico como de servicio, ejecutándose además de forma descoordinada.

3.1. DISTRIBUCIÓN DE HIDRÓGENO EN LAS REFINERÍAS

Una de las industrias que presentan este tipo de redes y problemáticas asociadas son las refinerías. En los últimos años, las refinerías modernas han transformado sustancialmente su estructura para adecuarla a la necesidad de convertir productos pesados en ligeros para hacerlos más rentables, como a la legislación medioambiental que impone fuertes reducciones de las concentraciones de contaminantes en los productos finales, en particular de azufre.

Para ello se han construido nuevas plantas, la mayoría de las cuales utilizan hidrógeno como elemento principal en el tratamiento de los hidrocarburos, convirtiéndose éste en un producto esencial, cuya gestión debe ser eficaz.

El tipo más habitual de plantas consumidoras de H₂ son las denominadas hidrodesulfuradoras y se basan en la mezcla de hidrógeno de alta pureza con el hidrocarburo para que reaccione con el azufre en una serie de reactores en presencia de un catalizador, formándose entre otros compuestos, sulfuro de hidrógeno que a la salida del reactor es separado del hidrocarburo en columnas de separación obteniéndose como producto final un hidrocarburo desulfurado.

La producción de hidrógeno en grandes cantidades para ser utilizado en las hidrodesulfuradoras es factible hoy en día a partir de hidrocarburos en plantas cuyo elemento principal son los hornos de reformado. Desde ellas se distribuye a las diferentes plantas consumidoras a través de un sistema complejo de tuberías (colectores) de varios kilómetros de longitud, diferentes diámetros y trabajando a distintas presiones y purezas.

La siguiente figura muestra la complejidad de un esquema típico de una red, con sus diversos colectores (de alta pureza C-H₄, de media pureza C-H₃ y de baja pureza C-BP) y con las principales plantas productoras de H₂, algunas dedicadas en exclusiva a producir H₂ (H₃ y H₄ remarcadas en azul) y otras como P1 y P2 (remarcadas en verde) cuya principal función es aumentar el octanaje de las gasolinas. Las plantas consumidoras están representadas en bloques remarcados en rojo.

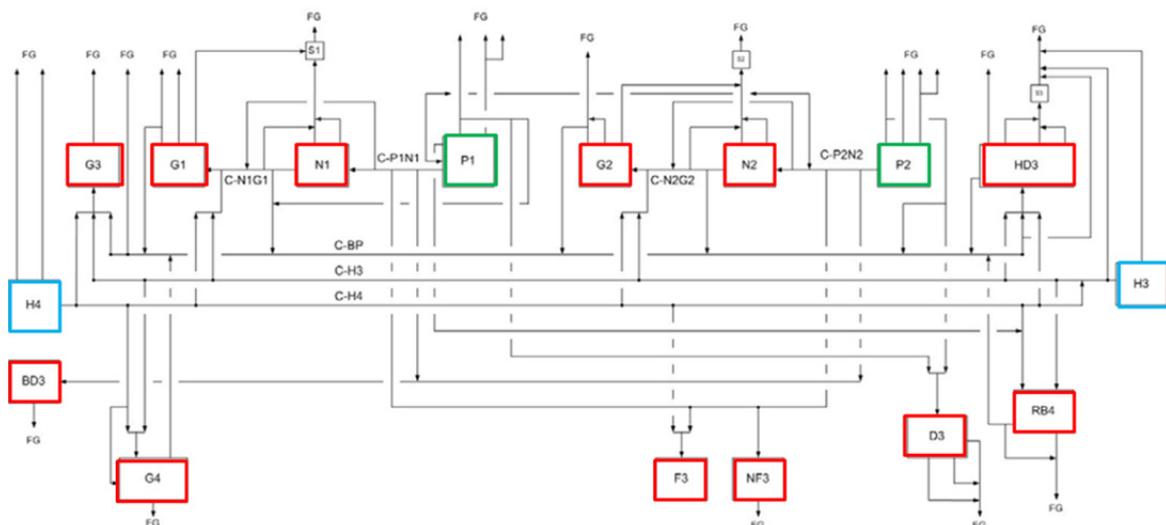


Figura 3.1. Esquema típico de una red de H₂ de una refinería [4]

Aunque las redes tienen varios kilómetros a lo largo de una refinería, su capacidad de almacenamiento es muy limitada, por lo que para garantizar el cumplimiento de las demandas cambiantes de las plantas consumidoras se mantiene un exceso de hidrógeno en la red de distribución. Sin embargo, al ser el hidrógeno un gas caro de producir, no es deseable una sobreproducción aunque ésta asegure la alimentación de las plantas. Por otro lado un defecto de hidrógeno puede limitar la producción o la vida de los catalizadores de los reactores.

Desde los años setenta, el diseño de los procesos químicos ha evolucionado hacia la integración de las operaciones que incrementan la eficiencia de la planta. La mejora del diseño del proceso incluye, entre otras cosas, integración de masa y energía, y el uso de corrientes de recirculación. Como resultado, los procesos se hacen más complejos y las unidades se llegan a encontrar más interrelacionadas [5].

La cuestión de control se aborda con el estudio de las diferentes estrategias de control para sistemas de gran escala interconectados.

4. DESCRIPCIÓN DE LA RED DE DISTRIBUCIÓN DE OXÍGENO

La planta piloto está compuesta por los siguientes elementos:

- Dos generadores de oxígeno, cada uno de ellos con una capacidad de producción y una pureza diferentes.
 - El generador de oxígeno 1, de alta pureza, proporciona flujo con una pureza del 62% en volumen, a una presión de trabajo de 0.3 barg en el colector 1, con una capacidad de generación máxima de 250 l/h.
 - El generador de oxígeno 2, de media pureza, proporciona flujo con una pureza del 50% en volumen, a una presión de trabajo de 0.3 barg en el colector 2, con una capacidad de generación máxima de 250 l/h.
- Dos colectores encargados de la distribución del oxígeno a las unidades de consumo.
- Tres unidades consumidoras de oxígeno que simulan el comportamiento de unidades de hidrodesulfuración de una refinería.
- Sistema de control distribuido, compuesto por un autómata programable (PLC), un sistema de control y adquisición de datos (SCADA) e instrumentación industrial.

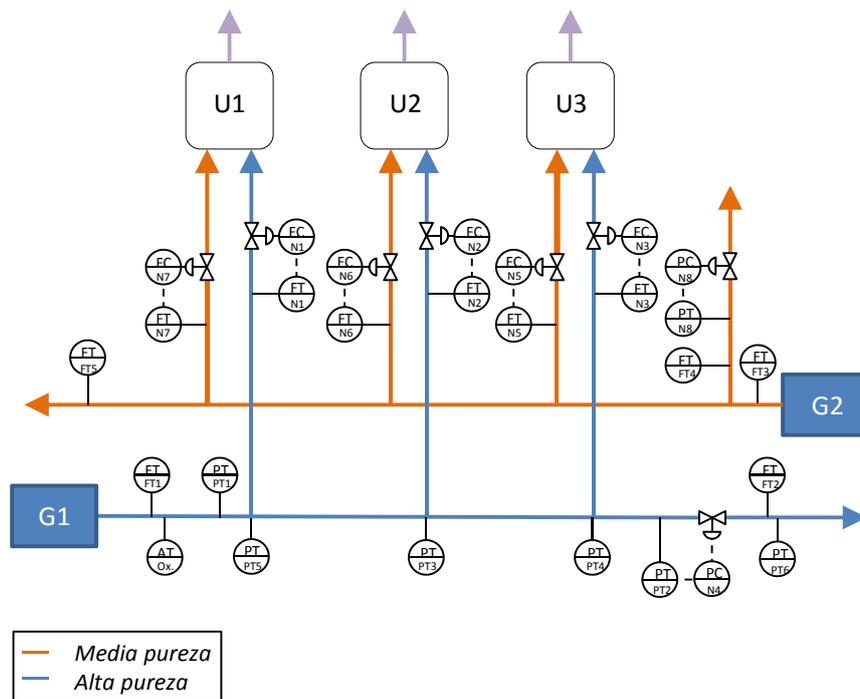


Figura 4.1. Diagrama de la planta de distribución de oxígeno

El control de presión de los colectores se realiza mediante purga a la atmósfera, controlada mediante un lazo de control de presión. En el primer colector la purga se sitúa en el extremo, mientras que en el segundo, se localiza a la salida del generador.

La arquitectura del sistema de control implementado en la planta piloto se muestra en la siguiente figura.

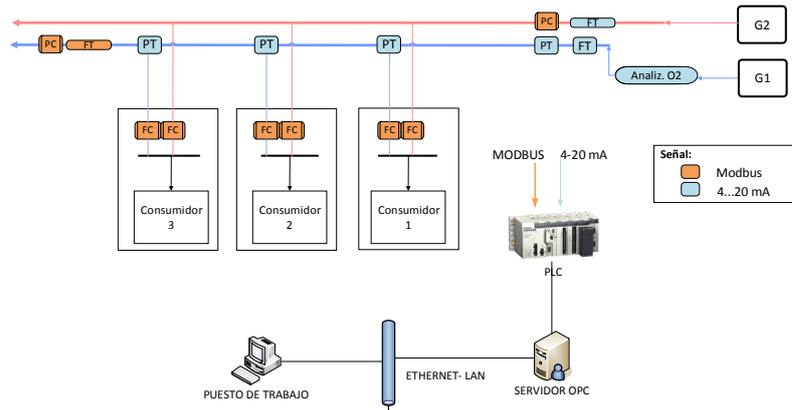


Figura 4.2. Arquitectura del sistema de control

Se dispone de la siguiente instrumentación:

- Un analizador de pureza de oxígeno, que se puede conectar tanto al primer colector como al segundo.
- Caudalímetros y sensores de presión a la salida de cada generador.
- Sensores de presión en cada una de las tomas de consumo.
- Caudalímetros en las purgas de ambos colectores.
- Seis lazos de control de caudal (sensor de caudal y controlador) para el control del flujo de oxígeno suministrado a cada una de las unidades consumidoras, desde ambos colectores.

Para la comunicación usuario/planta se emplea el sistema de adquisición y control (SCADA) EDUSCA (*Educational Scada*) [6].

En las siguientes figuras se presenta el interfaz gráfico del SCADA:

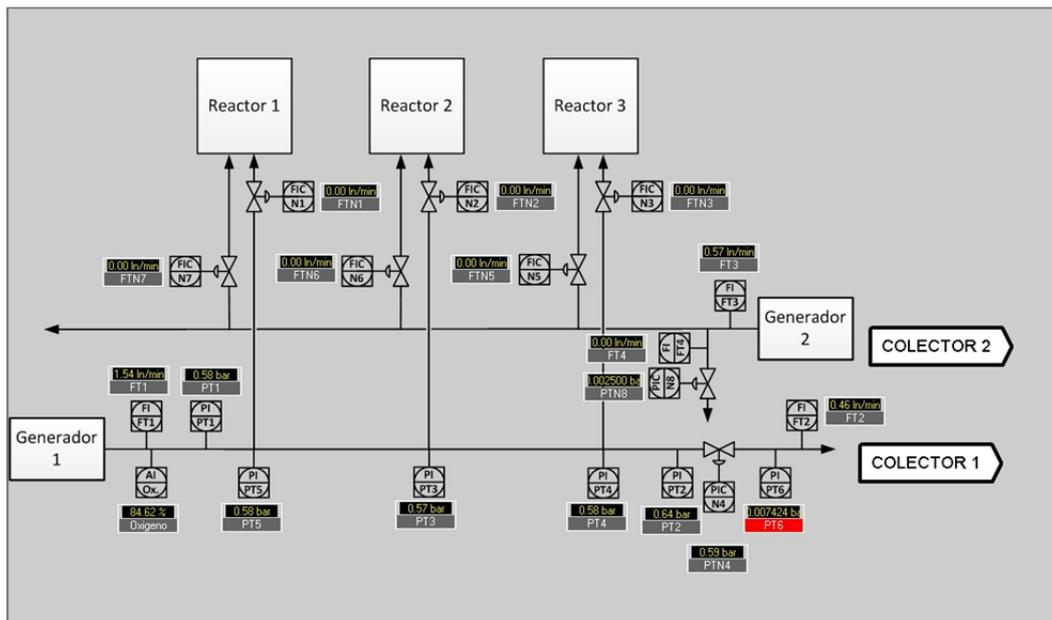


Figura 4.3. Sinóptico principal

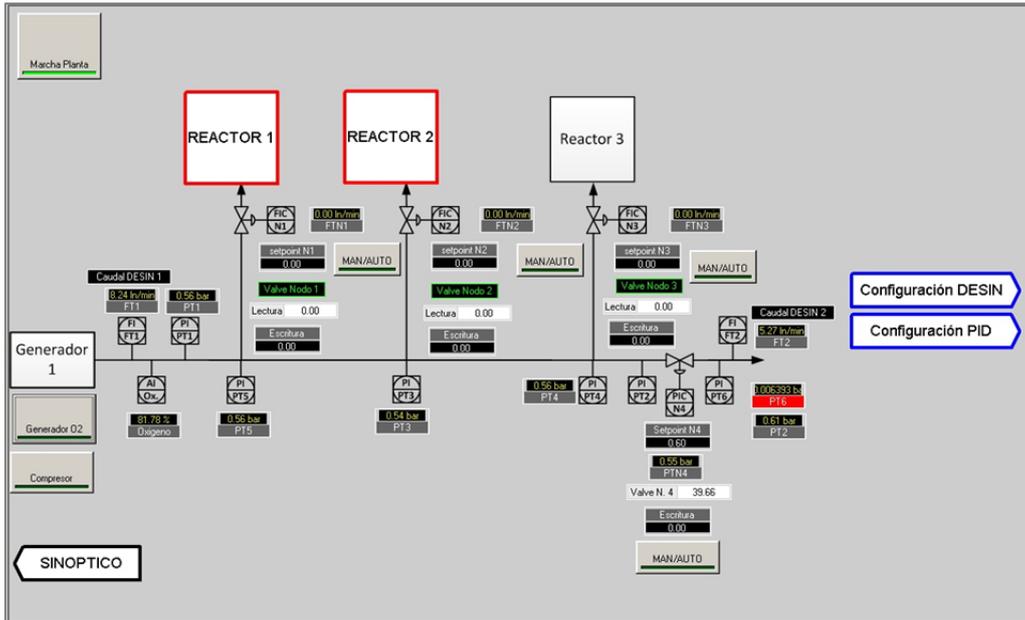


Figura 4.4. Colector 1

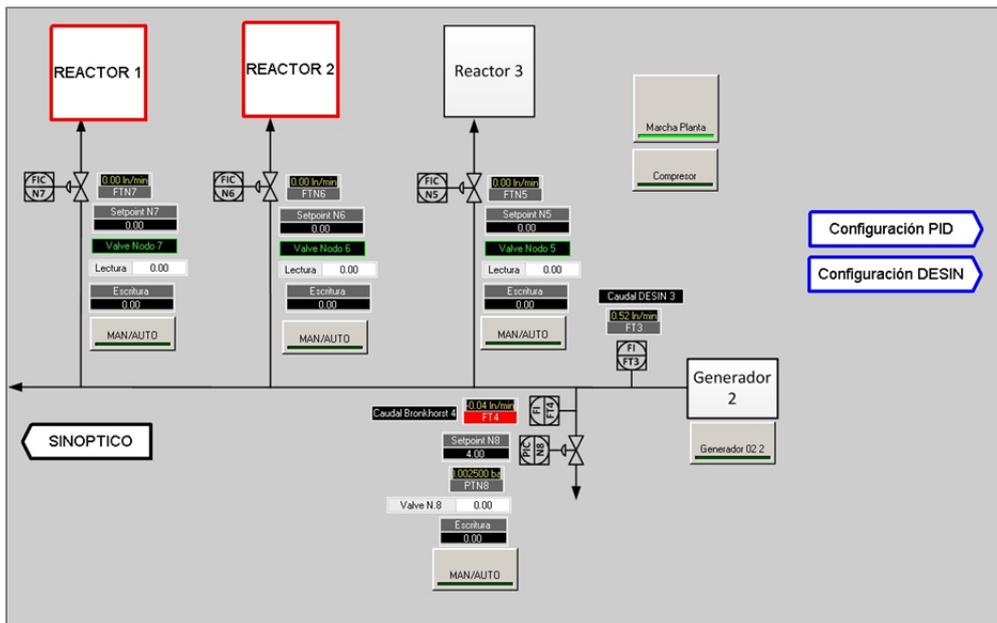


Figura 4.5. Colector 2

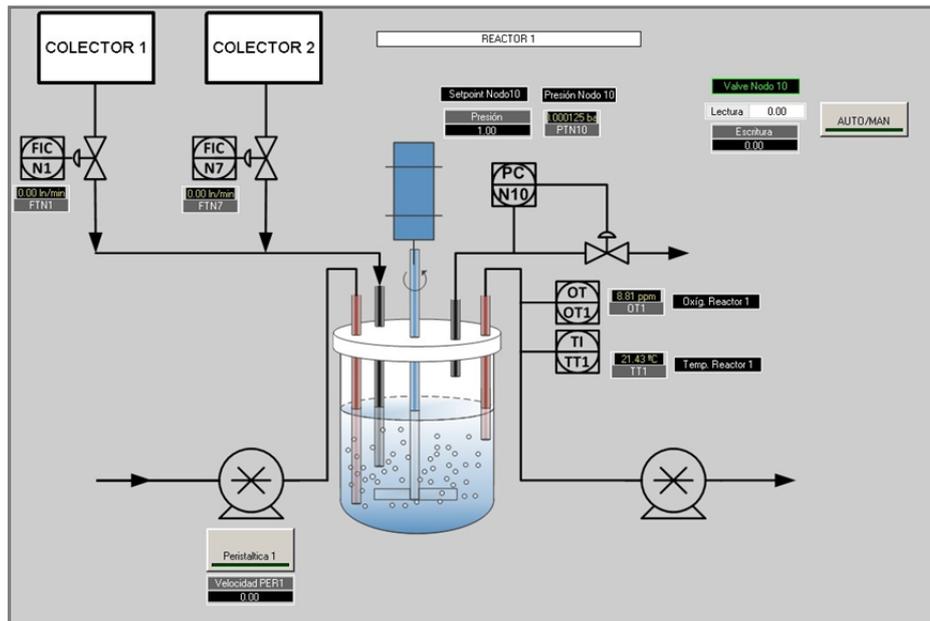


Figura 4.6. Reactor 1

Unidades consumidoras

Las unidades consumidoras simulan unidades de hidrodesulfuración de una refinería. El reactivo empleado en una hidrodesulfuradora es el hidrógeno, sin embargo en la planta piloto se trabaja con oxígeno.

Por otra parte, en lugar de tener la reacción correspondiente a la eliminación de azufre del hidrocarburo, se lleva a cabo el fenómeno físico de difusión de oxígeno en agua en las unidades consumidoras, también llamadas reactores, de modo que se simula el consumo de hidrógeno a través de la transferencia de oxígeno en un flujo continuo de agua.

La variable controlada, por tanto, es la concentración de oxígeno disuelto, la perturbación medible es la temperatura y las variables manipuladas son los flujos en ambos colectores.

La instrumentación de las unidades consumidoras consta de:

- Sonda de oxígeno disuelto y temperatura
- Bomba peristáltica para la impulsión de agua
- Controlador de presión

Se supone régimen de mezcla perfecta en el reactor, ya que se mantiene una agitación elevada para mantener uniforme la concentración de oxígeno en el reactor.



Imagen 4.1. Planta de oxígeno



Imagen 4.2. Primera unidad consumidora

4.1. CONDICIONES DE OPERACIÓN

Las condiciones de operación de la planta se pueden dividir en aquellas que determinan las condiciones de trabajo en las líneas de gas (colectores) y las correspondientes a las líneas de líquido en las unidades consumidoras.

En cuanto a las líneas de gas, es necesario determinar la presión de operación en ambos colectores, que debe ser la misma en ambos para evitar que en la unión de las ramas de ambos colectores antes de la entrada al reactor se produzca un flujo inverso de gas por gradiente de presión.

Además, han de tenerse en cuenta las siguientes limitaciones:

- La presión de los colectores debe ser suficiente para vencer las pérdidas de carga en las líneas y la altura de nivel de líquido en las unidades consumidoras.
- Ya que la presión determina el flujo total suministrado por los generadores y, por tanto, la pureza de oxígeno en ambos colectores, es necesario determinar la pureza con la que se quiere trabajar en las líneas de oxígeno. Ésta no debe exceder el valor máximo de 63%, ya que el rango de medida de las sondas de oxígeno va hasta 300% de la saturación en aire ($3 \cdot 21\% \text{ O}_2 = 63\% \text{ O}_2$).

Teniendo en cuenta las restricciones mencionadas, se decide trabajar a una presión relativa de **0.3 barg**, que se fija en la purga de ambos colectores, que proporciona una pureza del **62% en el primer colector y 50% en el segundo**.

Por otra parte, el tiempo de residencia del líquido (agua) en las unidades consumidoras se determina fijando el volumen y el caudal proporcionado por la bomba peristáltica.

Para un volumen dado, un flujo muy elevado conlleva un tiempo de residencia bajo, de modo que el rango de concentraciones de oxígeno disuelto alcanzable será muy bajo; en caso contrario, un flujo muy bajo de agua lleva consigo un tiempo de residencia elevado, siendo el rango de concentraciones más cercano a saturación y teniendo además la desventaja de un tiempo para alcanzar valores estacionarios más lento.

En el presente proyecto la identificación del sistema y la implantación del controlador predictivo se ha llevado a cabo en la primera unidad, fijando un volumen de **0.84 litros** con un flujo de agua de **0.55 l/min**, lo que proporciona un **tiempo de residencia de 1.5 min**.

Por último, la presión en la unidad consumidora de oxígeno se mantiene a presión atmosférica.

5. CONCEPTOS BÁSICOS SOBRE TRANSFERENCIA DE MATERIA G-L

Teoría de la película:

Cuando un componente se mueve desde el seno de una fase a la interfase, el fenómeno es conocido como transferencia de materia. El flujo de materia N_i es proporcional al gradiente producido entre la concentración C y la concentración de la interfase C_i .

La constante de proporcionalidad (k) resultante de la relación entre el flujo de materia y el gradiente de concentración se denomina coeficiente de transferencia de materia [7]:

$$N_i = k \cdot (C_i - C) \quad (5-1)$$

Donde k tiene un índice L ó G dependiendo de si la concentración corresponde a la fase gas o la fase líquida.

De acuerdo con la teoría de la película, aunque el seno de ambos fluidos se encuentre perfectamente mezclado, la turbulencia en cada fluido disminuye cerca de la interfase formándose una película fina de fluido en régimen laminar a cada uno de los lados de la interfase, de un espesor δ . La transferencia de materia a través de esta interfase se produce de forma perpendicular a la película y únicamente por difusión molecular.

La concentración de oxígeno varía en zonas cercanas a la interfase. La mayor parte de la resistencia a la transferencia de materia tiene lugar en la película de líquido más que en el seno de los fluidos (al estar perfectamente mezclados, la concentración tiende a ser constante).

A efectos prácticos, generalmente se supone despreciable la resistencia al transporte en la interfase, lo que equivale a suponer que las fases se encuentran en equilibrio en el plano de contacto, como se indica en la siguiente figura.

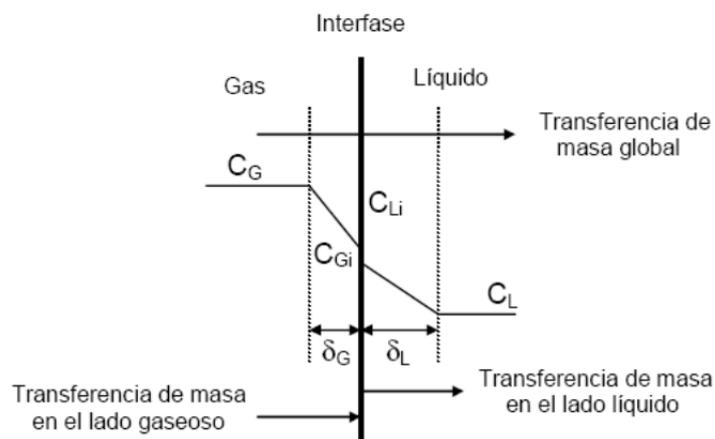


Figura 5.1. Perfil de concentraciones. Teoría de la película

Las dos concentraciones en equilibrio en la interfase C_{Gi} y C_{Li} de la película gaseosa y líquida respectivamente, pueden ser relacionadas linealmente a través de la ley de partición:

$$M \cdot C_{Li} = C_{Gi} \quad (5-2)$$

donde M es el coeficiente de distribución.

La velocidad de transferencia de oxígeno a la interfase gas-líquido debe ser igual a la velocidad de transferencia de oxígeno a través de la película de líquido. De esta forma pueden escribirse dos ecuaciones para la velocidad de transferencia:

$$\text{Flujo de oxígeno (lado gas)} = k_G \cdot (C_G - C_{Gi}) \quad (5-3)$$

$$\text{Flujo de oxígeno (lado líquido)} = k_L \cdot (C_{Li} - C_L) \quad (5-4)$$

Ya que no es posible obtener de forma experimental las concentraciones en la interfase, se expresa la transferencia de materia en términos de un coeficiente global de transferencia K_L y una fuerza impulsora global $(C_L^* - C_L)$, donde C_L^* es la concentración de la fase líquida cuando en equilibrio con la fase gas:

$$\text{Flujo} = K_L \cdot (C_L^* - C_L) \quad (5-5)$$

La relación entre los coeficientes globales e individuales de transferencia de materia viene dada por la expresión:

$$\frac{1}{K_L} = \frac{1}{k_L} + \frac{1}{M \cdot k_G} \quad (5-6)$$

Para especies poco solubles, como el oxígeno, M es considerablemente mayor que la unidad. Además, k_G es por lo general mucho mayor que k_L . Así, esencialmente toda la resistencia a la transferencia de materia se sitúa en el lado de la película líquida:

$$K_L \approx k_L \quad (5-7)$$

Correlaciones para el cálculo de $k_L a$:

En bibliografía se pueden encontrar varias correlaciones de $k_L a$ en función de propiedades físicas como la difusividad del gas en el líquido, la densidad y viscosidad del líquido, la temperatura a la que tiene lugar la transferencia; y condiciones de operación tales como la turbulencia, la fracción de burbujas de gas en el seno del líquido, el tamaño de las burbujas, etc.

Algunas de ellas son:

- **Modelo García-Ochoa, Gómez (2004) [8]**

$$k_L = \frac{2}{\sqrt{\pi}} \sqrt{D_L} \left(\frac{\varepsilon \cdot \rho_L}{\mu_L} \right)^{1/4} \quad (5-8)$$

Siendo D_L es la difusividad del gas en el líquido y ε la energía disipada por la turbulencia.

La difusividad se puede expresar mediante la fórmula de Wilke - Chang:

$$D_A = \frac{R \cdot T}{6 \cdot \pi \cdot \mu_B \cdot r_A} \quad (5-9)$$

donde R es la constante universal de los gases, r_A es el radio de la molécula de soluto A, la cual se asume de forma esférica y μ_B representa la viscosidad del solvente.

- **Correlación de Kawase (Dudley, 1995 [9]) [10]**

Se trata de una correlación deducida de forma teórica, recomendada para uso general:

$$k_L = 2 \cdot \sqrt{\frac{D_L}{\pi}} \sqrt{\frac{u_{sg} \cdot g}{\mu_L}} \quad (5-10)$$

donde u_{sg} es la velocidad superficial del gas.

- **Correlación de Higbie (Dudley, 1995 [9]) [10]**

$$k_L = 2 \cdot \sqrt{\frac{D_L}{\pi \cdot t}} \quad (5-11)$$

donde t se define según la ecuación:

$$t = d_b \cdot \varepsilon_g / u_{sg} \quad (5-12)$$

Siendo d_b es el diámetro de las burbujas de gas en el seno del líquido y ε_g la porosidad del difusor.

Solubilidad de gases:

La solubilidad de gases en medio líquido para un gas poco o moderadamente soluble es descrita por la ley de Henry [11]:

$$C_{O_2}^{sat} = \frac{P_{O_2}}{H} = \left(\frac{P}{H}\right) \cdot y_{O_2} \quad (5-13)$$

Donde:

- $C_{O_2}^{sat}$ es la concentración de saturación de oxígeno en la fase líquida.
- P es la presión total en la fase gas.
- P_{O_2} es la presión parcial de oxígeno en la fase gas.
- H es constante de Henry.
- y_{O_2} es la fracción molar de oxígeno en la fase gas.

La variación de la constante de Henry con la temperatura viene dada por la tabla:

TABLE 2-140 Oxygen (O₂)—Temperature

<i>t</i> , °C	0	5	10	15	20	25	30	35
$10^{-4} \times H$	2.55	2.91	3.27	3.64	4.01	4.38	4.75	5.07
<i>t</i> , °C	40	45	50	60	70	80	90	100
$10^{-4} \times H$	5.35	5.63	5.88	6.29	6.63	6.87	6.99	7.01

International Critical Tables, vol. 3, p. 257. Pray, Schweickert, and Minnich [*Ind. Eng. Chem.*, **44**, 1146 (1952)] give $H = 4.46 \times 10^{-4}$ at 25°C and other values up to 343°C.

Perry's Handbook of Chemical Engineering [11]

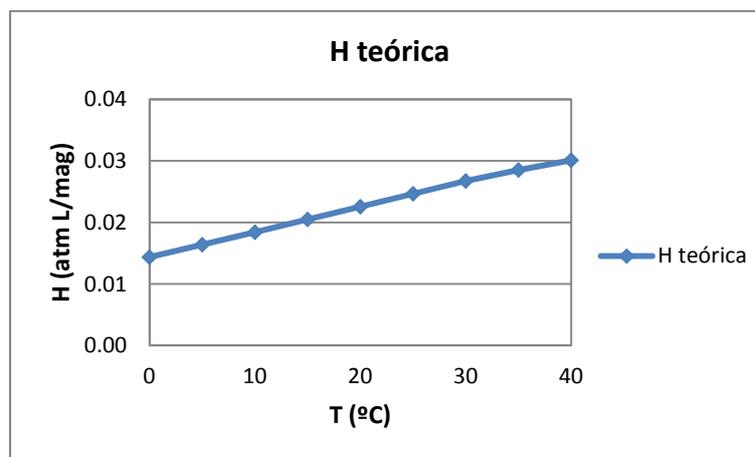


Figura 5.2. Variación de la constante de Henry con la temperatura

6. IDENTIFICACIÓN BASADA EN UN MODELO DE CONOCIMIENTO

El proceso que se identifica experimentalmente es la difusión de oxígeno en agua, en un reactor continuo de tanque agitado (modelo de mezcla perfecta) alimentado por una corriente de agua (F) y un flujo de oxígeno (F_{O_2}), que entra a través de un difusor situado en la parte inferior del reactor. El modelo de conocimiento viene dado por el balance de materia en fase líquida:

$$V_L \cdot \frac{dC_{O_2}}{dt} = N_a \cdot V_L + F \cdot (C_{O_2}^{in} - C_{O_2}) \quad (5-14)$$

donde:

- F es el flujo de agua.
- V_L es el volumen de agua en el reactor.
- $C_{O_2}^{in}$ es la concentración de oxígeno del agua a la entrada del reactor.
- C_{O_2} es la concentración de oxígeno a la salida del reactor.
- N_a es el flujo de oxígeno que se transfiere por difusión, que viene dado por la siguiente ecuación:

$$N_a = k_1 a \cdot (C_{O_2}^{sat} - C_{O_2}) \quad (5-15)$$

Siendo $C_{O_2}^{sat}$ la concentración de saturación y $k_1 a$ el coeficiente de transferencia de materia en la película líquida.

Sustituyendo la expresión 5-15 en el balance de materia, se obtiene:

$$V_L \cdot \frac{dC_{O_2}}{dt} = k_1 a \cdot V_L \cdot (C_{O_2}^{sat} - C_{O_2}) + F \cdot (C_{O_2}^{in} - C_{O_2}) \quad (5-16)$$

En estado estacionario $\frac{dC_{O_2}}{dt} = 0$, por tanto, despejando $k_1 a$:

$$k_1 a = -\frac{F \cdot (C_{O_2}^{in} - C_{O_2})}{V_L \cdot (C_{O_2}^{sat} - C_{O_2})} \quad (5-17)$$

Aplicando la ecuación 5-16, obtenida a partir del modelo de conocimiento, en régimen estacionario, se puede despejar el valor del coeficiente de transferencia de materia $k_1 a$ en distintos puntos de operación (expresión 5-17).

De esta forma se obtiene un modelo de $k_1 a$ en función del flujo de oxígeno.

Una vez se realizan distintos pre-tests para verificar la dinámica del sistema, se planifican distintos experimentos para identificar kla despejando de la ecuación 5-17. Es decir, un valor de kla para cada estado estacionario en distintas condiciones de operación.

La variable controlada es la concentración de oxígeno en el reactor, la variable manipulada es el flujo de oxígeno y como perturbación medible la temperatura, que como se ha revisado en los conceptos teóricos, influye en la constante de Henry, y por tanto en la concentración de oxígeno.

Es necesario, por tanto, conocer la variación experimental de la concentración de saturación con la temperatura. Para ello, se satura cierto volumen de líquido en el reactor a distintas temperaturas, obteniendo los siguientes resultados:

Saturación a presión atmosférica, pureza: 21% O₂ (aire):

T (°C)	C o ₂ sat (ppm)	H exp. (atm L/mg)
8.35	12.35	0.0170
13.63	10.70	0.0196
17.72	9.68	0.0217
19.98	9.40	0.0223
21.34	9.13	0.0230

Tabla 6.1. H vs T. Presión atmosférica, pureza de O₂ = 21% (aire)

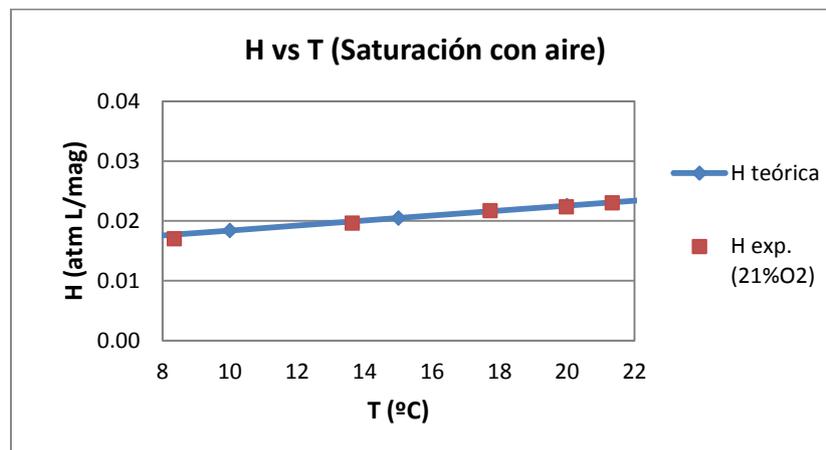
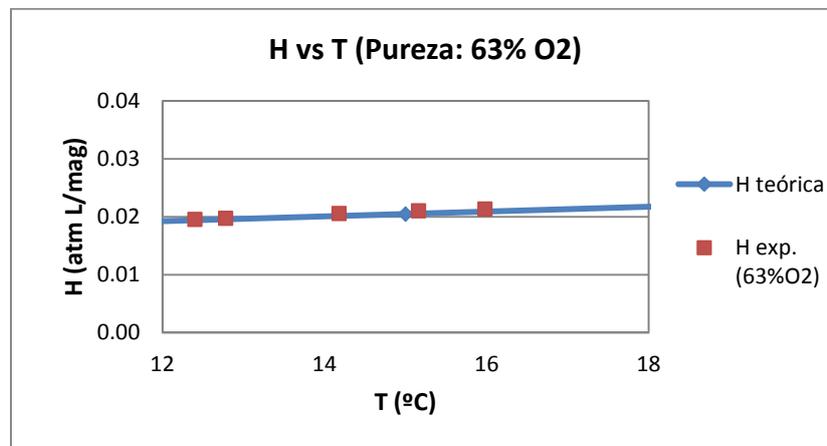


Figura 6.1. Identificación de H vs T. Presión atmosférica, pureza de O₂ = 21% (aire)

Saturación a presión atmosférica, pureza: 63% O₂:

T (°C)	C o ₂ sat (ppm)	H exp. (atm L/mg)
15.98	29.53	0.0213
15.16	29.95	0.0210
14.18	30.63	0.0206
12.78	31.91	0.0197
12.40	32.23	0.0195

Tabla 6.2. H vs T (Presión atmosférica, pureza de O₂ = 63%).**Figura 6.2.** Identificación de H vs T (Presión atmosférica, pureza de O₂ = 63%).Saturación a presión atmosférica, pureza: 50% O₂:

T (°C)	C o ₂ sat (ppm)	H exp. (atm L/mg)
16.57	23.11	0.0216
15.13	24.01	0.0208
14.13	24.48	0.0204
13.34	25.08	0.0199
12.25	25.76	0.0194

Tabla 6.3. H vs T (Presión atmosférica, pureza de O₂ = 50%)

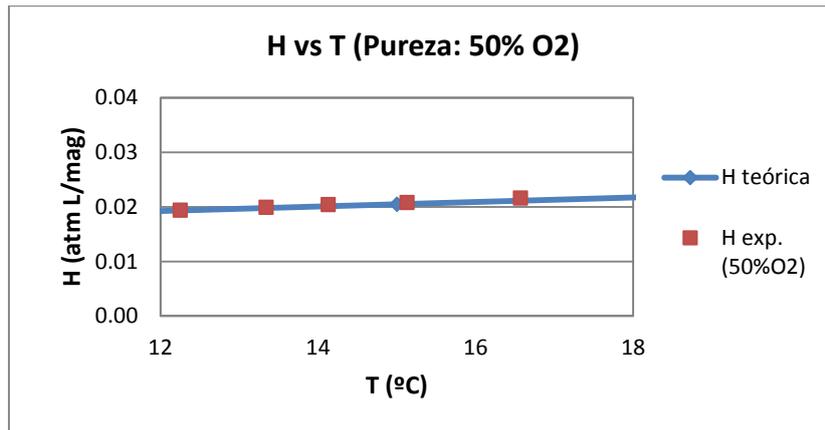


Figura 6.3. Identificación de H vs T (Presión atmosférica, pureza de O₂ = 50%)

Como puede apreciarse en las gráficas, la constante de Henry calculada a partir de los datos experimentales de concentración de oxígeno disuelto a distintas temperaturas, se ajustan muy bien a los valores de la constante de Henry teórica (*International Critical Tables, Perry's Handbook of Chemical Engineering [11]*).

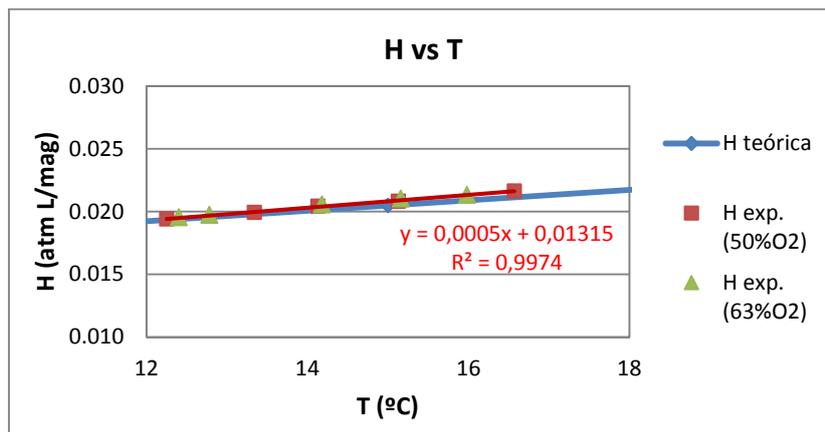


Figura 6.4. Identificación de H vs T (Presión atmosférica, pureza de O₂ = 50% y 63%). Ajuste lineal

Se obtiene la expresión:

$$H\left(\frac{\text{atm} \cdot \text{L}}{\text{mg}}\right) = 5 \cdot 10^{-4} \cdot T(^{\circ}\text{C}) + 1.315 \cdot 10^{-2} \quad (5-18)$$

De esta forma, pueden calcularse las concentraciones $C_{\text{O}_2}^{\text{in}}$ y $C_{\text{O}_2}^{\text{sat}}$ en función de la temperatura, a partir de la ecuación de Henry:

$$C_{\text{O}_2}^{\text{sat}} = \left(\frac{P}{H}\right) \cdot y_{\text{O}_2} = \left(\frac{P}{5 \cdot 10^{-4} \cdot T + 1.315 \cdot 10^{-2}}\right) \cdot y_{\text{O}_2} \quad (5-19)$$

6.1. IDENTIFICACIÓN DE $k_L a$

Se realizan dos experimentos a distintas temperaturas variando el flujo de O₂ (variable manipulada). A partir de los valores de concentración de oxígeno alcanzados en estado estacionario se calcula el coeficiente de transferencia de materia a partir de la ecuación 5-17.

Condiciones de operación	Exp 1	Exp 2
T media (°C)	11.32	17.23
P reactor (barg)	0	0
Pureza colector (%O ₂)	62	62
Presión colector (barg)	0.3	0.3
Nivel de agitación	400 rpm	400 rpm
Frecuencia de la bomba (%)	60	60
Flujo agua (L/min)	0.54	0.58
Volumen (L)	0.88	0.86
C o ₂ in media (ppm)	11.16	9.65
C o ₂ sat media (ppm)	32.96	28.49

Tabla 6.4. Condiciones de operación de los experimentos

Experimento 1:

F _{O₂} (NL/min)	T (°C)	C o ₂ out (ppm)	C o ₂ in (ppm)	C o ₂ sat (ppm)	k _L a (1/min)
4.00	11.31	29.74	11.17	32.97	3.53
2.00	11.31	28.66	11.17	32.97	2.49
3.20	11.28	29.40	11.18	33.00	3.11
2.40	11.24	29.02	11.19	33.03	2.73
3.60	11.34	29.57	11.16	32.94	3.35
2.80	11.58	28.98	11.09	32.73	2.92
4.40	11.35	29.80	11.16	32.93	3.65
0.80	11.19	26.17	11.20	33.08	1.33
1.60	11.15	28.20	11.21	33.11	2.12
0.40	11.48	22.94	11.12	32.82	0.73
1.20	11.33	27.41	11.16	32.95	1.80

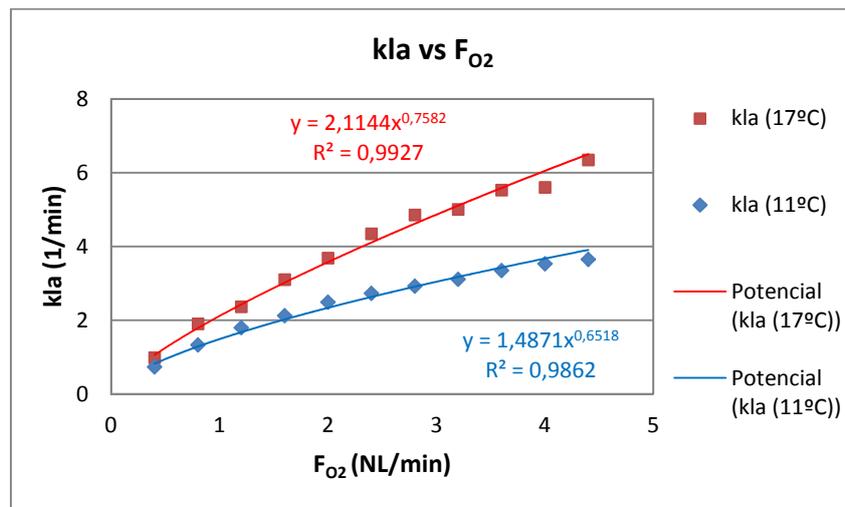
Tabla 6.5. Experimento 1. T media 11.3°C

Experimento 2:

F _{O₂} (NL/min)	T (°C)	C o ₂ out (ppm)	C o ₂ in (ppm)	C o ₂ sat (ppm)	k _{la} (1/min)
4.00	16.57	26.87	9.80	28.92	5.60
2.00	16.81	25.82	9.74	28.76	3.68
3.20	16.85	26.48	9.73	28.74	5.00
2.40	16.97	26.11	9.71	28.66	4.34
3.60	17.08	26.53	9.68	28.58	5.53
2.80	17.21	26.20	9.65	28.50	4.85
4.40	17.34	26.61	9.62	28.41	6.35
0.80	17.70	23.30	9.55	28.18	1.90
1.60	17.74	24.83	9.54	28.16	3.10
0.40	17.62	20.65	9.56	28.23	0.99
1.20	17.62	24.09	9.56	28.23	2.36

Tabla 6.6. Experimento 2. T media 17.2°C

Representando los datos experimentales de *k_{la}* en función del flujo de oxígeno:

Figura 6.5. Identificación de *k_{la}* vs Flujo de O₂ a dos temperaturas (Exp. 1: 11°C y Exp. 2: 17°C)

Se observa un incremento del coeficiente de transferencia de materia al aumentar el flujo de gas, obteniendo el mejor ajuste con una función potencial. En cuanto a la temperatura, el coeficiente de transferencia de materia aumenta al aumentar la temperatura, ya que *k_{la}* es directamente proporcional a la difusividad, siendo ésta proporcional a la raíz cuadrada de la temperatura según predicen las ecuaciones empíricas anteriormente mencionadas.

Mediante el comando *sftool* de Matlab [12], se obtiene un modelo de kla :

$$k_l a = (T - n)^{0.5} \cdot m \cdot F_{O_2}^a \quad (5-20)$$

Modelo	
$kla = ((T-n)^{0.5}) * m * Fo2^a$	
a	0.6822
m	0.7432
n	7.6670

Tabla 6.7. Parámetros del modelo

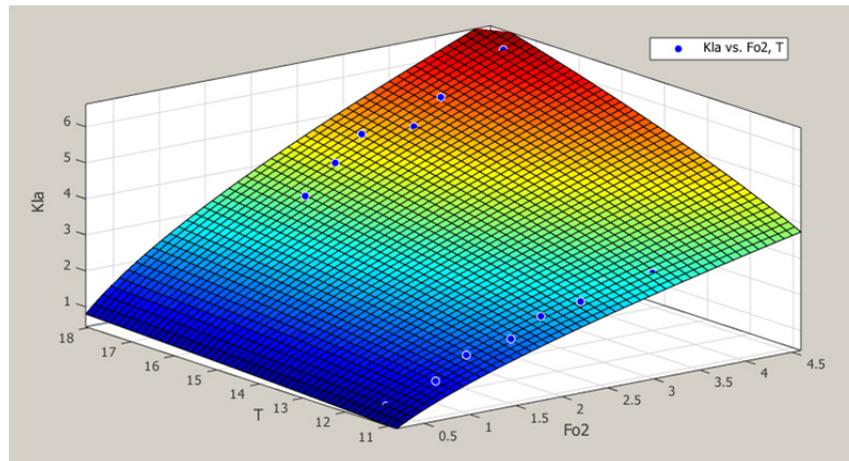


Figura 6.6. Ajuste de kla en función del flujo de gas y la temperatura

Bondad del ajuste:

- SSE (*Sum Squared Error*): 0.5821
- RMSE (*Root Mean Square Error*): 0.175
- R-square: 0.9883
- Adjusted R-square: 0.9871

El modelo, por tanto, viene dado por las expresiones:

- Balance de materia al oxígeno en fase líquida:

$$V_L \cdot \frac{dC_{O_2}}{dt} = \left[(T - n)^{0.5} \cdot m \cdot F_{O_2}^a \right] \cdot V_L \cdot (C_{O_2}^{sat} - C_{O_2}) + F \cdot (C_{O_2}^{in} - C_{O_2}) \quad (5-21)$$

- Concentraciones de saturación y a la entrada:

$$C_{O_2}^{sat} = \left(\frac{P}{H} \right) \cdot y_{O_2} \quad (5-22); \quad C_{O_2}^{in} = \left(\frac{P_{atm}}{H} \right) \cdot 0.21 \quad (5-23)$$

- Expresión de la constante de Henry en función de la temperatura:

$$H = 5 \cdot 10^{-4} \cdot T + 1.315 \cdot 10^{-2} \quad (5-24)$$

- Balance de materia global en fase gas

$$F_{O_2} = F_{O_2C1} + F_{O_2C2} \quad (5-25)$$

- Balance individual al oxígeno:

$$F_{O_2} \cdot y_{O_2} = F_{O_2C1} \cdot y_{O_2C1} + F_{O_2C2} \cdot y_{O_2C2} \quad (5-26)$$

Siendo el rango de validez:

- Flujo de gas: $0.4 \leq F_{O_2} \text{ (L/min)} \leq 4.4$
- Temperatura: $11 \leq T \text{ (°C)} \leq 17$

Calculando la concentración de oxígeno a partir de los valores de kla del modelo:

F_{O_2} (NL/min)	kla 11°C (1/min)	kla 17°C (1/min)	C o2 (11°C) (ppm)	C o2 (17°C) (ppm)
4.00	3.65	5.71	29.83	26.90
2.00	2.28	3.61	28.34	25.77
3.20	3.12	4.98	29.41	26.47
2.40	2.55	4.12	28.80	25.99
3.60	3.41	5.46	29.62	26.51
2.80	2.97	4.63	29.03	26.11
4.40	3.92	6.35	29.99	26.61
0.80	1.20	2.02	25.67	23.52
1.60	1.91	3.25	27.79	24.96
0.40	0.78	1.25	23.24	21.71
1.20	1.61	2.66	26.94	24.45

Tabla 6.8. Valores de concentración de oxígeno predichos por el modelo

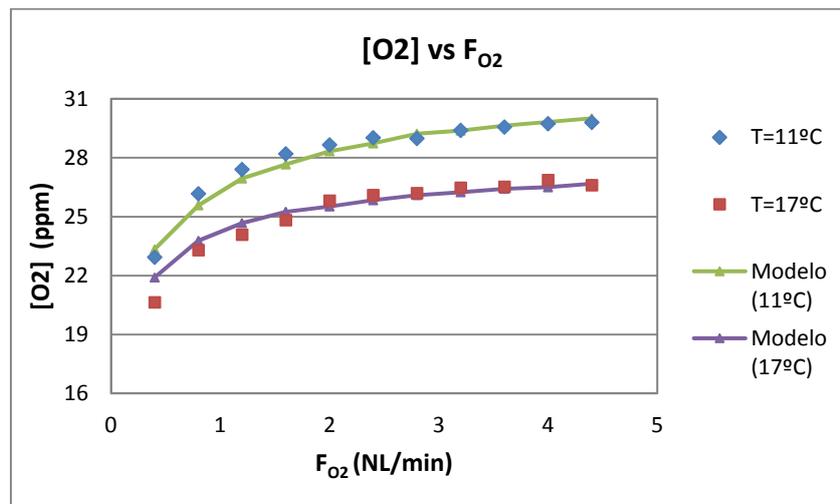


Figura 6.7. Ajuste entre los datos experimentales de C_{O_2} y la predicción del modelo a 11°C y a 17°C

6.2. VALIDACIÓN DEL MODELO

En la validación, llevada a cabo a una temperatura intermedia ($T = 15^{\circ}\text{C}$), se dan varios saltos positivos y negativos al flujo de oxígeno, tomando los valores de concentración en estacionario y comparando posteriormente el valor de kla así obtenido con el valor que predice el modelo.

Condiciones de operación	Validación
T media ($^{\circ}\text{C}$)	15.06
P reactor (barg)	0
Pureza colector (% O_2)	62
Presión colector (barg)	0.3
Nivel de agitación	400 rpm
Frecuencia de la bomba (%)	60
Flujo agua (L/min)	0.55
Volumen (L)	0.84
C o2 in media (ppm)	10.15
C o2 sat media (ppm)	29.98

Tabla 6.9. Condiciones de operación para el experimento de validación

Comparación entre kla experimental y kla del modelo:

F_{O_2} (NL/min)	T ($^{\circ}\text{C}$)	C o2 out (ppm)	kla exp. (1/min)	kla modelo (1/min)	Error kla (%)
0.60	15.57	23.15	1.33	1.47	10.99
1.00	15.53	24.61	1.90	2.08	9.97
0.80	15.52	24.11	1.66	1.79	7.60
1.20	15.04	25.04	1.97	2.29	16.23
0.40	15.39	21.38	0.89	1.11	24.89
2.00	14.99	26.44	2.97	3.23	8.81
4.00	14.86	27.56	4.43	5.13	15.90
1.60	14.89	26.22	2.70	2.75	1.91
2.50	14.86	26.98	3.49	3.72	6.68
4.40	14.89	27.71	4.79	5.49	14.60
3.00	14.73	27.24	3.73	4.18	11.93
3.50	14.50	27.52	3.93	4.57	16.28

Tabla 6.10. Validación. Coeficiente de transferencia de materia

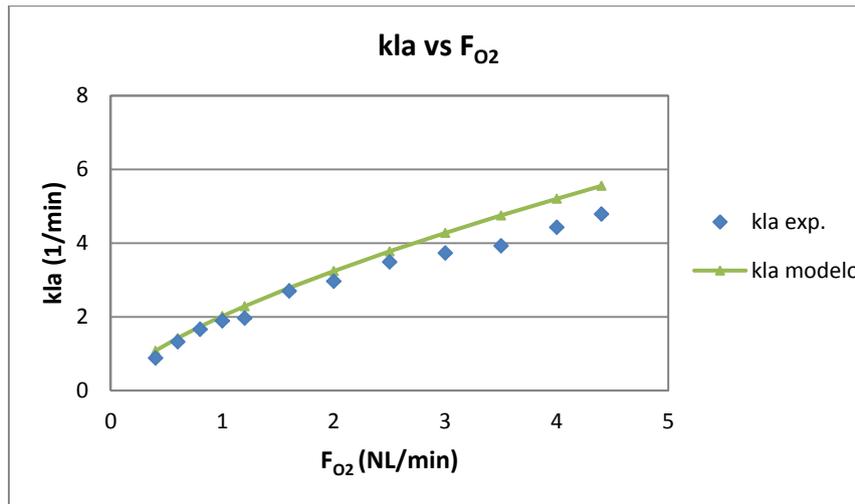


Figura 6.8. Experimento de validación. Ajuste de kla

La media de los errores entre los valores del coeficiente de transferencia predichos por el modelo y los valores calculados a partir de la concentración de oxígeno medida es 12%.

Comparación entre la concentración de oxígeno experimental y la del modelo:

F_{O_2} (NL/min)	T (°C)	C o2 exp. (ppm)	C o2 modelo (ppm)	Error O2 (%)
0.60	15.57	23.15	23.59	1.91
1.00	15.53	24.61	24.96	1.41
0.80	15.52	24.11	24.40	1.19
1.20	15.04	25.04	25.58	2.15
0.40	15.39	21.38	22.43	4.90
2.00	14.99	26.44	26.68	0.91
4.00	14.86	27.56	27.87	1.13
1.60	14.89	26.22	26.28	0.22
2.50	14.86	26.98	27.15	0.62
4.40	14.89	27.71	27.98	0.98
3.00	14.73	27.24	27.51	1.01
3.50	14.50	27.52	27.87	1.28

Tabla 6.11. Validación. Concentración de oxígeno

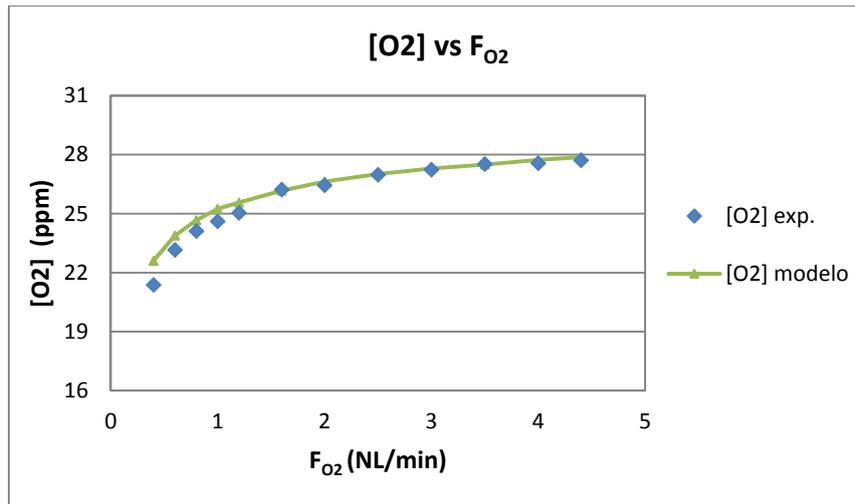


Figura 6.9. Experimento de validación. Ajuste de C_{O₂}

La media de los errores entre la concentración predicha por el modelo y la concentración medida es 1.5%.

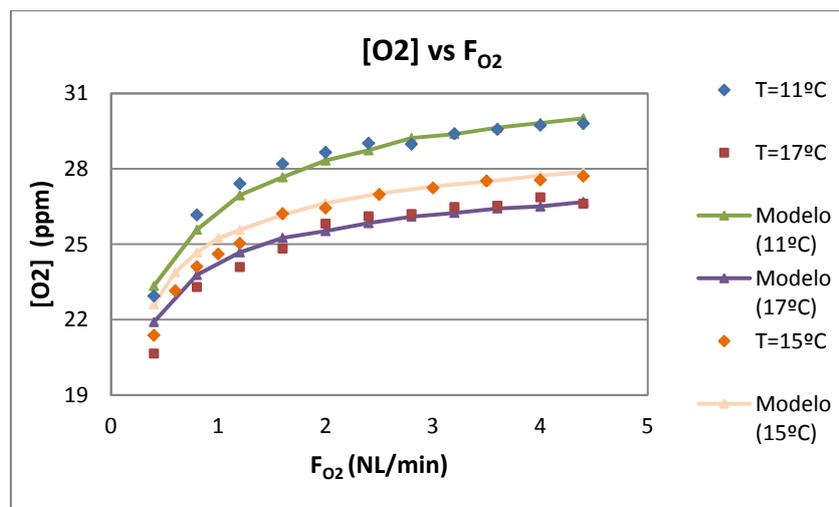


Figura 6.10. Ajuste de C_{O₂}. Experimentos de identificación (11°C y 17°C) y validación (15°C)

6.3. SIMULACIÓN DEL MODELO EN ECOSIM

```

COMPONENT Modelo

  DATA
    REAL V = 0.84          -- Volumen de líquido (L)
    REAL F = 0.55          -- Flujo de agua (L/min)
    REAL yo2c1 = 0.62      -- Fracción molar de O2 colector 1
    REAL yo2c2 = 0.50      -- Fracción molar de O2 colector 2
    REAL P_tot = 1         -- Presión total (atm)
    -- Parámetros del modelo (a, m, n) (kla = m*((T-n)^0.5)*Fo2^a)
    -- Rango de validez: T de 11 a 17°C y Fo2 de 0.4 a 4.4 L/min.
    REAL a = 0.6822
    REAL m = 0.7432
    REAL n = 7.667

  DECLS
    REAL Co2in  -- Concentración de O2 a la entrada (ppm)
    REAL Co2out -- Concentración de O2 a la salida (ppm)
    REAL Co2sat -- Concentración de O2 en saturación (ppm)
    REAL Fo2c1  -- Flujo de gas colector 1 (L/min)
    REAL Fo2c2  -- Flujo de gas colector 2 (L/min)
    REAL Fo2    -- Flujo de gas (L/min)
    REAL yo2    -- Pureza (tanto por 1)
    REAL T      -- Temperatura (°C)
    REAL H      -- Constante de Henry (atm L/mg)
    REAL kla    -- Coeficiente de transferencia de materia (1/min)

  CONTINUOUS
    -- Balance de materia global
    Fo2 = Fo2c1 + Fo2c2
    -- Balance de materia al O2
    Fo2*yo2 = Fo2c1*yo2c1+Fo2c2*yo2c2
    -- Constante de Henry experimental
    H = 0.0005*T + 0.01315
    -- Concentración de saturación a partir de la constante de Henry experimental
    Co2sat = P_tot*yo2/H
    -- Concentración de entrada en función de la constante de Henry experimental
    Co2in = P_tot*0.21/H
    -- kla modelo: kla = m*((T-n)^0.5)*Fo2^a
    kla = m*((T-n)**0.5)*Fo2**a
    -- Balance de materia en fase líquida
    V*Co2out' = kla*V*(Co2sat-Co2out) + F*(Co2in-Co2out)

END COMPONENT

```

Hasta este punto se ha realizado un análisis de los datos obtenidos en distintas condiciones de operación en régimen estacionario, sin embargo para llevar a cabo la identificación es necesario tener en cuenta la dinámica del sistema.

En las siguientes gráficas se muestra la evolución dinámica de los experimentos llevados a cabo y su comparación con el modelo.

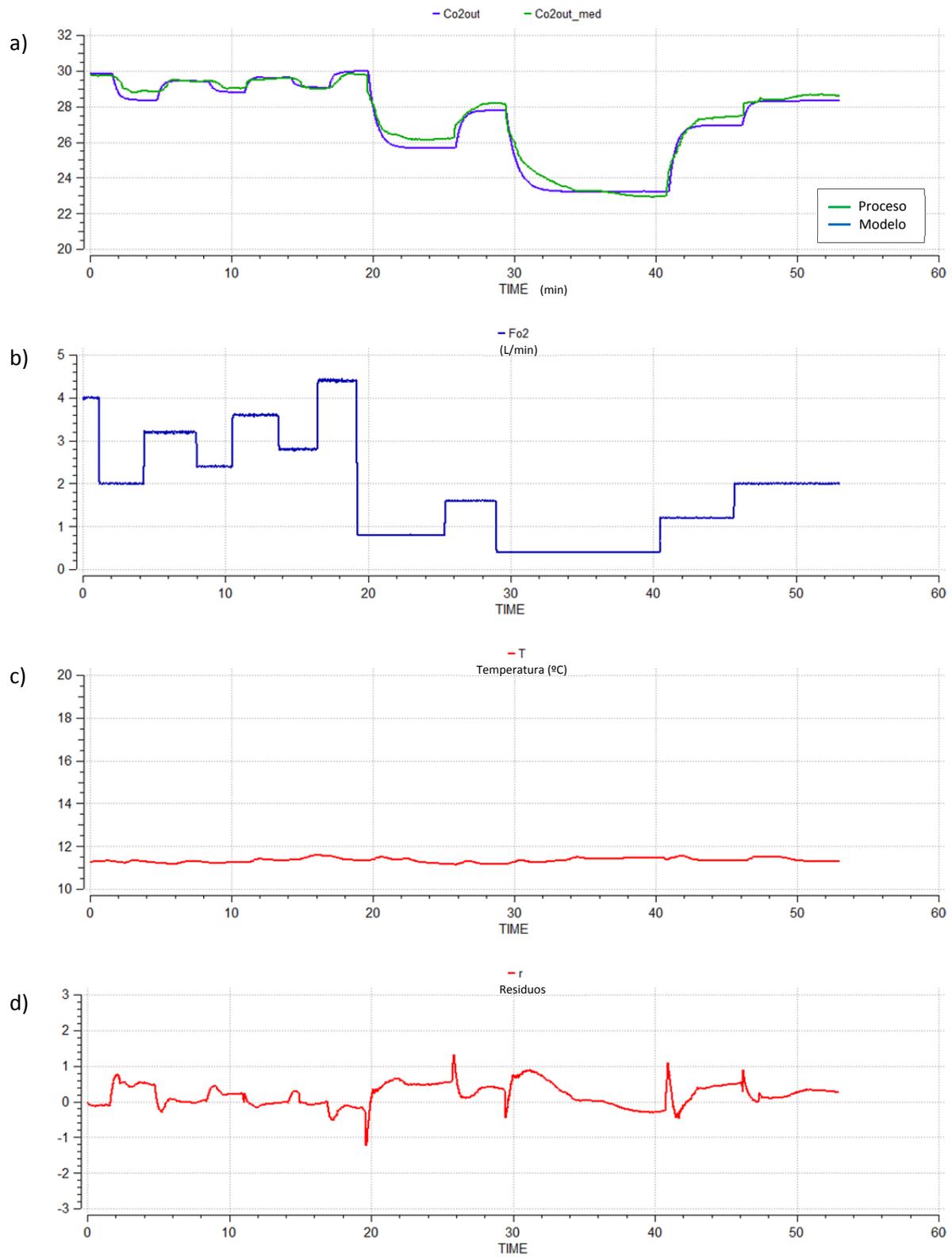
Experimento 1 (T = 11°C): Comparación dinámica del proceso y el modelo

Figura 6.11. Experimento 1. Comparación proceso – modelo. (a) Evolución de la concentración del proceso y del modelo en lazo abierto ante saltos en el flujo de gas (b). (c) Temperatura, (d) Residuos

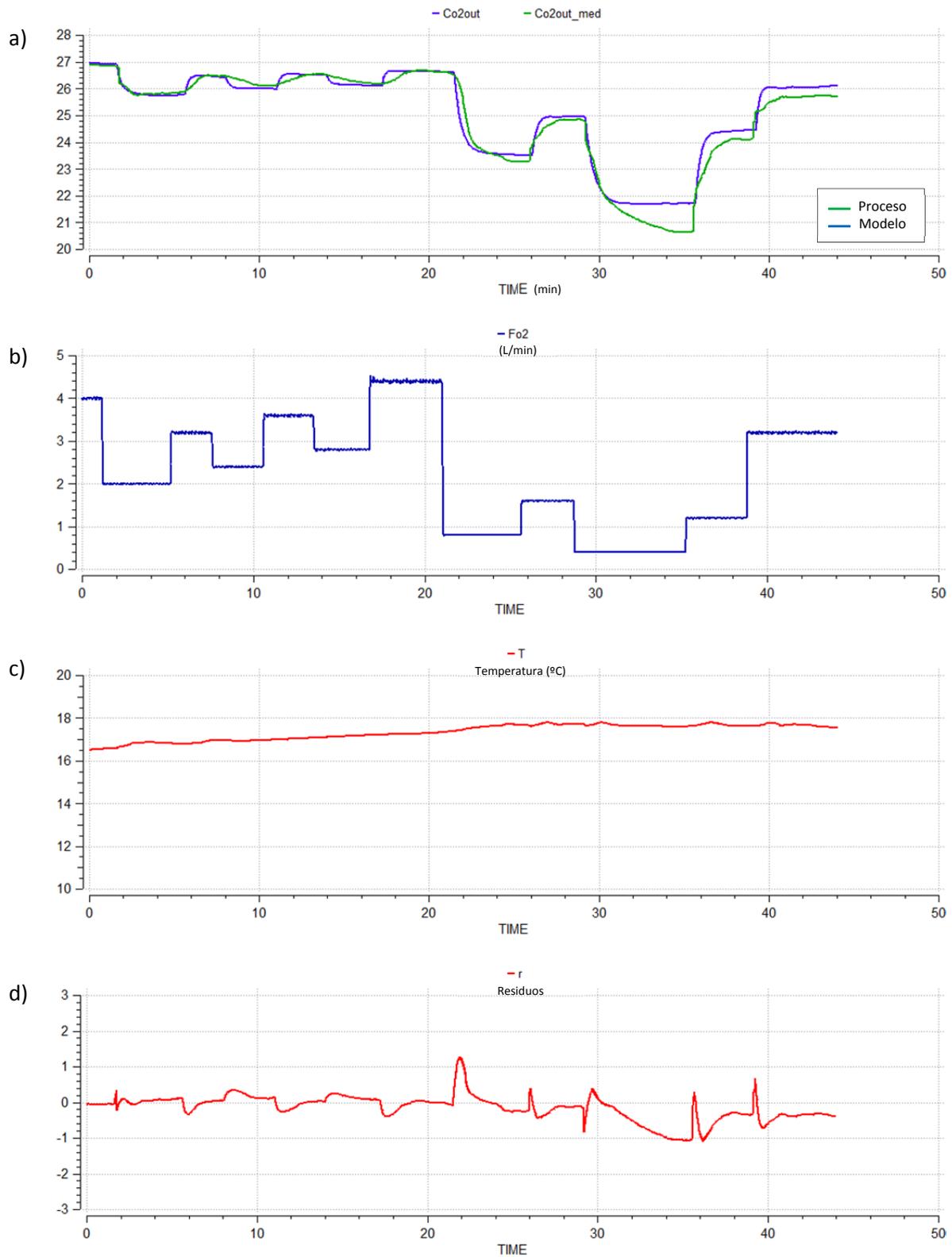
Experimento 2 (T = 17°C): Comparación dinámica del proceso y el modelo

Figura 6.12. Experimento 2. Comparación proceso – modelo. (a) Evolución de la concentración del proceso y del modelo en lazo abierto ante saltos en el flujo de gas (b). (c) Temperatura, (d) Residuos

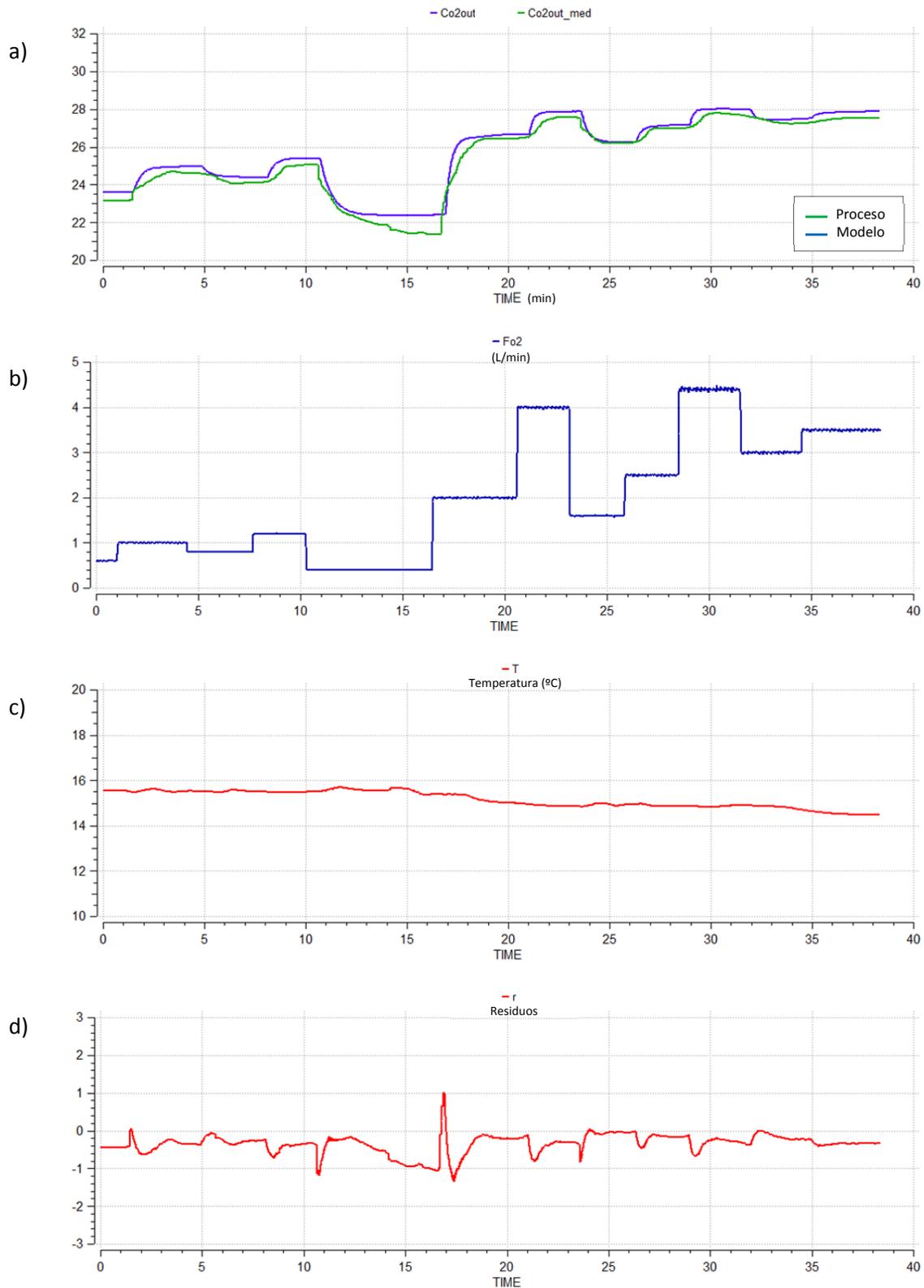
Validación (T = 15°C): Comparación dinámica del proceso y el modelo

Figura 6.13. Validación. Comparación proceso – modelo. (a) Evolución de la concentración del proceso y del modelo en lazo abierto ante saltos en el flujo de gas (b). (c) Temperatura, (d) Residuos

Error cuadrático medio:

$\sqrt{\frac{\sum_{t=1}^N (y_{real}(t) - y_{mod}(t))^2}{N}}$	Error cuadrático medio (RMSE)		
	Exp. 1 (T=11°C)	Exp. 2 (T=11°C)	Validación (T=15°C)
	0.37	0.39	0.43

Tabla 6.12. Errores cuadráticos medios obtenidos en los experimentos de identificación y validación**Índices asociados al modelo:**

Son índices que ponderan la suma de los cuadrados de los errores, obteniendo una medida de la bondad de ajuste del modelo, de modo que cuanto menor es el valor de tales índices, mejor es el modelo.

- Índice de Akaike (AIC):

$$\left[1 + \frac{2d}{N}\right] \cdot \sum_{t=1}^N (y_{real}(t) - y_{mod}(t))^2 \quad (6-1)$$

- Índice FPE (Final Prediction Error):

$$\frac{1}{N} \left[\frac{1+d/N}{1-d/N} \right] \cdot \sum_{t=1}^N (y_{real}(t) - y_{mod}(t))^2 \quad (6-2)$$

- Índice de Rissanen (Rissanen's Minimal Description Length):

Es el que más penaliza la complejidad del modelo

$$\left[1 + \frac{2d}{N} \log N\right] \cdot \sum_{t=1}^N (y_{real}(t) - y_{mod}(t))^2 \quad (6-3)$$

N es el número de datos tomados y d es el número de parámetros del modelo, en este caso d = 3.

Calculando dichos índices para los experimentos de identificación y el de validación del modelo, se obtiene:

Índices asociados al modelo	Experimento 1 T = 11°C	Experimento 2 T = 17°C	Validación T = 15°C
Akaike	433	396	432
FPE	0.14	0.15	0.19
Rissanen	435	399	435

Tabla 6.13. Índices asociados al modelo calculados en los experimentos de identificación y validación

7. CONTROLADOR PREDICTIVO

Actualmente la única técnica de control avanzado que ha tenido un impacto significativo en la ingeniería de control industrial ha sido el Control Predictivo basado en Modelos (*Model Predictive Control, MPC*) [13].

El concepto de control predictivo fue introducido simultáneamente por Richalet [14] y Cutler y Ramaker [15] en la década de los setenta y pertenece a la clase de controladores basados en modelos. Esto significa que utiliza de forma explícita un modelo del proceso que se desea controlar.

El fundamento del MPC es transformar el problema de control en uno de optimización [16], es decir, en cada periodo de muestreo se calcula la secuencia de acciones de control futuras como solución en un horizonte finito de un problema de control óptimo. A continuación, solo el primer elemento de la secuencia de acciones de control futuras calculadas se aplica al proceso, repitiendo todo el procedimiento en el siguiente periodo de muestreo de acuerdo con el principio del horizonte deslizante (*receding horizon*). La reformulación del problema de control en términos de uno de optimización permite un diseño con más grados de libertad en la selección de la función de coste que se desea minimizar y que puede reflejar diferentes objetivos de control.

Su utilización ha evolucionado principalmente en la industria de procesos, debido a sus características:

- El concepto de control predictivo no está restringido a procesos de una entrada y una salida (*Single Input - Single Output, SISO*), sino que permite su extensión directa a sistemas de múltiples entradas y salidas.
- Incorpora de forma natural la compensación *feedforward* de perturbaciones y puede utilizarse en sistemas con dinámicas complejas (retardos, respuesta inversa, etc.), así como para seguimiento de trayectorias de referencia.
- En contraste con otras técnicas de control basadas en modelos, el control predictivo puede usarse para el control de procesos no lineales. En este caso se usa explícitamente un modelo no lineal del proceso para el diseño del controlador.
- Es posible la introducción de restricciones en las variables de entrada y salida.
- El control predictivo es una metodología abierta, es decir, existen múltiples formas para su diseño. Como resultado, en la última década, se ha propuesto un elevado número de controladores predictivos, cada uno de ellos con propiedades diferentes, tales como el GPC (*Generalized Predictive Control*), DMC (*Dynamic Matrix Control*), EPSAC (*Extended Predictive Control*), DMC (*Dynamic Matrix Control*), PFC (*Predictive Functional Control*), etc.
- Finalmente, el control predictivo es conceptualmente simple de comprender y sintonizar.

La estructura básica para implementar el control predictivo viene representada por la siguiente figura [17]. Se usa un modelo para predecir la evolución de la salida o estado del proceso a partir de las señales de entrada y salidas conocidas. Las acciones de control futuras se calculan con el optimizador, que considera la función del coste y las posibles restricciones.

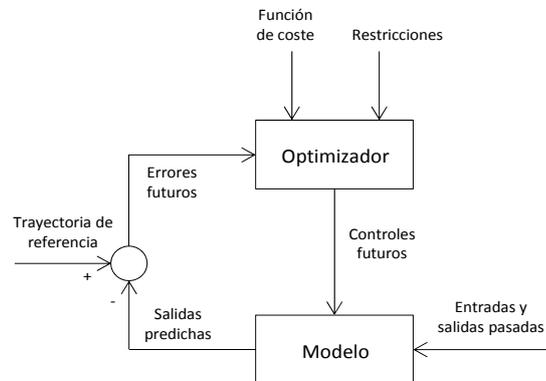


Figura 7.1. Estructura del MPC

La metodología de todos los controladores predictivos (MPC) se caracteriza por la siguiente estrategia, representada en la figura 7.1:

- Las salidas futuras para un horizonte determinado N_2 , llamado horizonte de predicción, se predicen cada instante de muestreo utilizando el modelo del proceso. Estas predicciones de la salida dependen de los valores conocidos hasta ese instante (entradas y salidas conocidas) y de las señales de control, que han de ser calculadas y enviadas al sistema.
- La secuencia de señales de control futuras se calcula minimizando un criterio para mantener al proceso lo más cerca posible de la trayectoria de referencia. Este criterio toma normalmente la forma de una función cuadrática del error entre la salida predicha y la trayectoria de referencia futura. Además, se suele incluir también el esfuerzo de control dentro de la función objetivo.
- La señal de control envía al proceso mientras que el resto de las señales calculadas no se consideran, ya que en el siguiente instante de muestreo es ya conocida y los pasos anteriores se repiten con este nuevo valor.

En la figura 7.2 se muestra un esquema gráfico de este procedimiento, con una variable controlada y una variable manipulada (sistema SISO). La variable controlada (y) se desea mantener en un valor de referencia; para ello, en cada instante de muestreo se plantea un problema de minimización del cuadrado del error entre el valor actual $y(t)$ del modelo y la referencia, a lo largo de un horizonte móvil dado por un número determinado de periodos de muestreo (N_2).

Es importante diferenciar el horizonte de predicción y el horizonte de control, N_u , siendo este último el número de periodos de muestreo que el problema de optimización calcula una nueva señal de control, permaneciendo constante hasta el horizonte de predicción N_2 . No obstante, se aplica solamente la primera acción de control calculada, ya que el próximo periodo de muestreo vuelve a ejecutarse el problema de optimización.

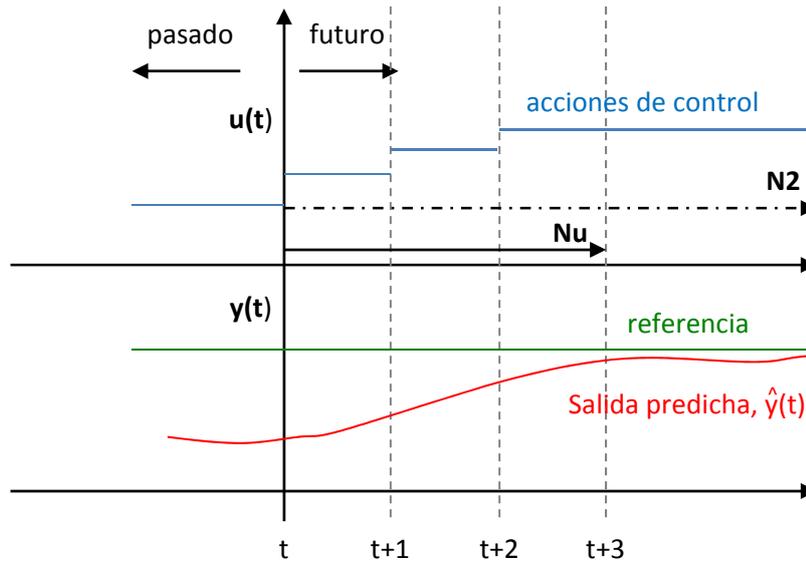


Figura 7.2. Esquema gráfico de la estrategia del horizonte móvil

La aplicación del controlador predictivo a la planta se muestra en la siguiente figura, donde las referencias de los controladores locales de flujo son fijadas por el MPC, que tiene a éstas como variables manipuladas. Se dispone de una única variable controlada, que es además la variable de estado del modelo (concentración de oxígeno disuelto en la unidad consumidora), siendo la perturbación medida la temperatura. En este caso, por tanto, se dispone de un sistema MISO (*Multiple Input, Single Output*).

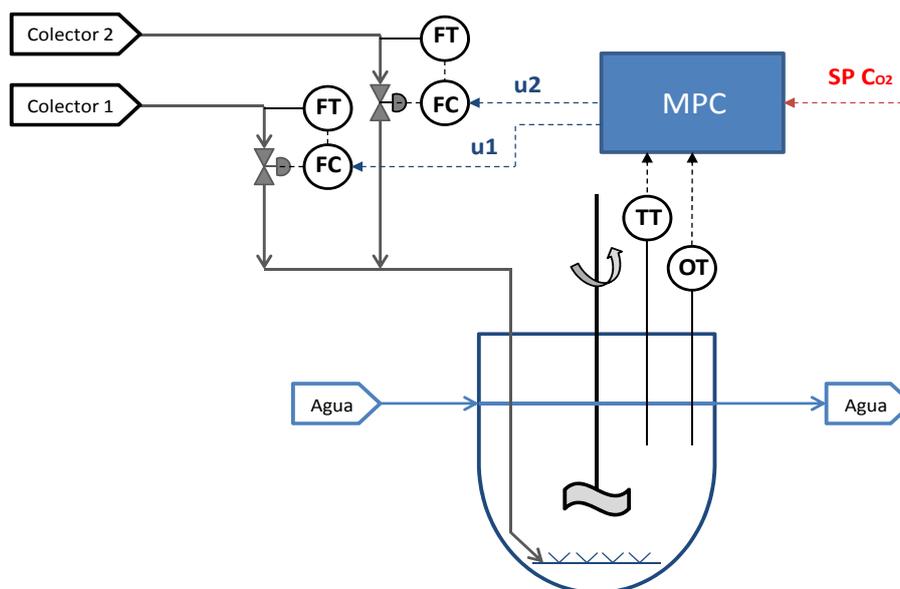


Figura 7.3. Esquema de funcionamiento del controlador predictivo aplicado a la planta

7.1. PROGRAMACIÓN DEL CONTROLADOR PREDICTIVO

Para llevar a cabo la programación, en primer lugar, se simula el control predictivo en Ecosim, creando una librería con los archivos:

- *coste.el*
- *control.el*

El archivo *coste.el* recibe los valores de las variables manipuladas (flujos en ambos colectores), la perturbación medible (temperatura) y la variable controlada (concentración de oxígeno) de *control.el*, calculando la función de coste (J). El archivo *control.el* recoge el valor de la función objetivo y lleva a cabo la optimización, proponiendo las nuevas acciones de control a aplicar (u).

Para simplificar la notación, las variables manipuladas se designan con la letra u , es decir, $u_1 = F_{O_2C1}$ y $u_2 = F_{O_2C2}$, mientras que la variable controlada y es la concentración de oxígeno disuelto C_{O_2} .

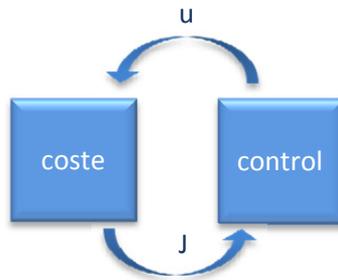


Figura 7.4. Esquema de funcionamiento de la simulación del control predictivo

En el fichero *coste* se define la función objetivo del problema de optimización del controlador predictivo, que se muestra a continuación. Por otro lado, en el archivo *control* se realiza la llamada a las funciones de optimización de la librería NAG [18], que ejecuta el problema de optimización mediante el método de programación cuadrática secuencial (SQP), pasando las nuevas acciones de control y el estado del proceso al fichero *coste*.

En resumen, se trata de llevar a cabo una simulación del controlador predictivo para comprobar la idoneidad de la respuesta del sistema, pudiendo modificar los parámetros de sintonía (horizonte de predicción, horizonte de control, tiempo de muestreo, etc.) de manera sencilla y observando su influencia en simulación, antes de aplicarlo a la planta real.

Igualmente pueden introducirse errores de modelado, modificando las ecuaciones que simulan el proceso en el código *control*, para implementar distintos métodos de corrección del error estacionario (corrección del error tras la integración de la función de coste, corrección en el punto inicial de integración, estrategia del estimador de horizonte móvil).

La formulación del problema de control es la siguiente:

$$\min_{F_{O_2C1}, F_{O_2C2}} J = \int_0^{t_f} \left((C_{O_2}(t) - w(t))^2 + \beta_1 \Delta F_{O_2C1}^2(t) + \beta_2 \Delta F_{O_2C2}^2(t) + \gamma \cdot \text{func}(t) \right) dt \quad (7-1)$$

$$\text{func}(t) = \max(F_{O_2C1}(t) + F_{O_2C2}(t) - F_{O_2}^{\max}, 0) \quad (7-2)$$

Sujeto a:

$$\frac{dw(t)}{dt} = \frac{C_{O_2}^{SP}(t) - w(t)}{\alpha} \quad (7-3)$$

Modelo:

$$V_L \cdot \frac{dC_{O_2}}{dt} = k_1 a \cdot V_L \cdot (C_{O_2}^{\text{sat}} - C_{O_2}) + F \cdot (C_{O_2}^{\text{in}} - C_{O_2}) \quad (7-4)$$

$$k_1 a = 0.7432 \cdot (T - 7.667)^{0.5} \cdot F_{O_2}^{0.6822} \quad (7-5)$$

$$C_{O_2}^{\text{sat}} = \left(\frac{P}{H} \right) \cdot y_{O_2} \quad (7-6)$$

$$C_{O_2}^{\text{in}} = \left(\frac{P_{\text{atm}}}{H} \right) \cdot 0.21 \quad (7-7)$$

$$H = 5 \cdot 10^{-4} \cdot T + 1.315 \cdot 10^{-2} \quad (7-8)$$

$$F_{O_2} = F_{O_2C1} + F_{O_2C2} \quad (7-9)$$

$$F_{O_2} \cdot y_{O_2} = F_{O_2C1} \cdot y_{O_2C1} + F_{O_2C2} \cdot y_{O_2C2} \quad (7-10)$$

Restricciones:

$$0.4 \leq F_{O_2C1}(t) \leq 4.4; \quad 0.4 \leq F_{O_2C2}(t) \leq 4.4 \quad (7-11)$$

$$-0.1 \leq \Delta F_{O_2C1}(t) \leq 0.1; \quad -0.1 \leq \Delta F_{O_2C2}(t) \leq 0.1 \quad (7-12)$$

Siendo:

$$\beta_1 = \beta_2 = 0.1$$

$$\gamma = 500$$

$$F_{O_2}^{\max} = 4.4 \text{ L/min}$$

$$\alpha = 0.3$$

Como puede observarse, la función de coste a minimizar cada periodo de muestreo incluye los siguientes términos:

- El cuadrado de la diferencia la predicción de la concentración del modelo y la trayectoria de referencia $w(t)$. Dicha trayectoria de referencia se modela para alcanzar el set point de concentración fijado por el operador con una velocidad que viene determinada por el parámetro α .
- La penalización de las acciones de control con un factor de peso igual a 0.1 para ambas, a fin de suavizar la respuesta del sistema, evitando respuestas bruscas en las acciones de control, lo que provocaría una respuesta subamortiguada en la evolución de la variable controlada.
- Penalización en caso de que se sobrepase el flujo máximo de identificación del modelo, correspondiente a 4.4 L/min. Esta es una forma de incluir una restricción en la propia función de coste:

En las restricciones, sí se especifican los límites de cada uno de los flujos (4.4 L/min como límite superior para cada uno de ellos), sin embargo eso no evita el hecho de que la suma de los dos sea superior al flujo de identificación, saliendo en ese caso del rango de validez del modelo.

Por ello, se define la expresión *func* que devuelve el máximo valor entre la suma de los flujos de los dos colectores menos el flujo máximo de identificación y cero. De esta forma, siempre que no se exceda el flujo máximo, *func* toma el valor cero y no influye en el problema de optimización. Sin embargo, cuando se excede, devuelve el valor correspondiente al exceso de flujo por encima del límite superior, que es multiplicado por un factor de peso elevado ($\gamma = 500$), de tal modo el algoritmo de optimización tratará de evitar en la medida de lo posible esa situación, ya que supone un coste muy elevado para la función objetivo.

El modelo de conocimiento viene dado por las expresiones ya comentadas en el apartado 6, es decir:

- Balance de materia en fase líquida al oxígeno (ecuación diferencial, en la que aparece la concentración de oxígeno como variable de estado).
- Expresión de ajuste del coeficiente de transferencia de materia, obtenida experimentalmente.
- Concentraciones de entrada y saturación según la ley de Henry.
- Expresión de la variación de la constante de Henry en función de la temperatura, obtenida experimentalmente.
- Balance de materia global al flujo de gas proveniente de cada colector y balance individual al oxígeno.

Finalmente, se definen restricciones sobre las variables manipuladas (flujos mínimos y máximos, correspondientes a los flujos de identificación) y sobre sus incrementos.

El flujo máximo de cada colector (4.4 L/min) viene dado por el hecho de que si se supera, se ha observado experimentalmente que la pureza del colector varía de forma significativa.

Los límites en los incrementos se fijan en ± 0.1 L/min a fin de lograr respuestas suaves, evitando así acciones de control muy forzadas y respuestas subamortiguadas en la variable controlada.

A continuación se explican los códigos programados en EcosimPro, que se adjuntan en el Anexo I. En el punto 7.1.3 se explica el fichero *controlador.el*, que, al igual que en el archivo *control.el*, realiza la llamada al optimizador de la librería NAG. Sin embargo, en lugar de incluir las ecuaciones del modelo en un nuevo componente, pasa a recibir los valores de la planta real vía OPC empleando el SCADA EDUSCA (*Educational Scada*) para la interacción del usuario con la planta.

En otras palabras, se trata de reemplazar el modelo por el proceso, llevando a cabo la sincronización en tiempo real para la lectura de las variables medidas en planta y la ejecución de las acciones de control propuestas por el optimizador aplicado a la función objetivo definida en el archivo *coste.el*. Dicha sincronización se lleva a cabo mediante la definición de un grafcet (**GRA**phe **F**unctionel de **C**ommande **E**tape **T**ransition) que se ejecuta cíclicamente con una etapa inicial de espera y otras dos etapas, de modo que en la primera se leen las variables del proceso en el instante actual y en la segunda se ejecutan las acciones de control, permitiendo así la comunicación en tiempo real con la planta. En el punto 7.1.3 se detalla la configuración del grafcet.

7.1.1. Archivo coste.el

En el bloque **DATA** se definen los parámetros del modelo, el flujo de agua, el volumen, las fracciones molares de oxígeno en ambos colectores y la presión de operación. Además, se especifica un estado inicial (subíndice i) mediante los valores en estacionario de los flujos de gas en ambos colectores, la temperatura y la concentración de oxígeno.

Igualmente, se indica el número de entradas al sistema (los dos flujos de gas), el número de salidas (la concentración de O_2) y las variables del controlador predictivo (horizonte de control, de predicción y el parámetro α) que tomarán los valores recibidos del código del controlador (*control.el*).

En el bloque **DECLS** se declaran las variables del proceso: temperatura, concentración de oxígeno a la entrada, a la salida y en saturación, flujo de gas en ambos colectores, flujo total, fracción molar de oxígeno, constante de Henry y coeficiente de transferencia de materia.

Además se declaran como variables la predicción de la variable controlada ($y_predic[nsal, maxn2]$), las variables manipuladas ($u[nent]$), el valor anterior de las variables manipuladas ($uant[nent]$), la variable controlada ($y[nsal]$), la referencia interna ($ref_int[nsal]$), la referencia futura ($ref_fut[nsal, maxn2]$), el índice que se minimiza (J), la consigna ($refy$), el error o BIAS ($Error$) y las variables iniciales de los bucles ($i, j, k, ind, Sample$) que se explicarán más adelante.

Aquellas variables que no son calculadas posteriormente en el bloque **CONTINUOUS** ni son tomadas como variables de contorno se definen como **DISCR REAL** ($T, y_predic[nsal, maxn2], u[nent], uant[nent], ref_fut[nsal, maxn2], J, Error$)

En el bloque **INIT** se asignan valores iniciales a la variable controlada, las variables manipuladas, la perturbación medible, la referencia interna y el índice a optimizar (valor inicial cero).

En el bloque **DISCRETE** se asignan desde el controlador los valores futuros a las variables manipuladas. Para ello se definen dos bucles FOR dentro de un bucle WHEN. Las variables que se ponen en juego en este bloque son:

- *Sample* (se activa cada periodo de muestreo)
- *ind* (inicialmente toma el valor de cero)
- *i* (va desde 1 hasta el número de variables manipuladas: 2)
- *j* (inicialmente vale 1 y va hasta el valor de Nu en el primer bucle FOR y hasta el número de salidas del sistema en el segundo)
- *k* (va desde 1 hasta Nu)

Cada vez que se llega a un nuevo periodo de muestreo se activa la variable *Sample* y la variable *ind* incrementa su valor en una unidad; por ejemplo, la primera vez que se entra al bucle $ind = 1$ y k toma el valor de ind ($k = 1$).

En el primer bucle FOR, que va desde $i = 1$ hasta $i = nent = 2$, se calcula j como el valor mínimo entre ind (1) y Nu (3), por tanto $j = 1$. Así se define el control futuro como $uf[i,j]$, es decir, $uf[1,1]$ y $uf[2,1]$.

En el segundo bucle FOR, que va desde $i = 1$ hasta el número de salidas al sistema (1), se define el valor predicho de la variable controlada $y_predic[i,k] = y[i]$, es decir, $y_predic[1,1] = y[1]$.

$$\begin{aligned} \text{Primera ejecución de los bucles FOR: } \quad & uf[1,j] = uf[1,1] \\ & uf[2,j] = uf[2,1] \\ & y_predic[i,k] = y_predic[1,1] \end{aligned}$$

$$\begin{aligned} \text{Segunda ejecución de los bucles FOR: } \quad & uf[1,j] = uf[1,2] \\ & uf[2,j] = uf[2,2] \\ & y_predic[i,k] = y_predic[1,2] \end{aligned}$$

$$\begin{aligned} \text{Tercera ejecución de los bucles FOR: } \quad & uf[i,j] = uf[1,3] \\ & uf[2,j] = uf[2,3] \\ & y_predic[i,k] = y_predic[1,3] \end{aligned}$$

De esta forma, en cada periodo de muestreo, se van construyendo las matrices:

$$\begin{pmatrix} u_{11} & u_{12} & u_{13} \\ u_{21} & u_{22} & u_{23} \end{pmatrix} = \begin{pmatrix} F_{O2C1(1)} & F_{O2C1(2)} & F_{O2C1(3)} \\ F_{O2C2(1)} & F_{O2C2(2)} & F_{O2C2(3)} \end{pmatrix}$$

$$\begin{pmatrix} y_{11} & y_{12} & y_{13} \end{pmatrix} = \begin{pmatrix} C_{o2(1)}^{out} & C_{o2(2)}^{out} & C_{o2(3)}^{out} \end{pmatrix}$$

En el bloque **CONTINUOUS** se escriben todas las ecuaciones que relacionan las variables y los parámetros definidos anteriormente:

- Variación de la constante de Henry con la temperatura.
- Balance global e individual (al O_2) aplicado a los flujos en ambos colectores.
- Concentraciones de entrada y de saturación en función la fracción molar de O_2 .
- Ecuación del modelo obtenido del coeficiente de transferencia de materia.
- Balance de materia al O_2 en fase líquida (ecuación diferencial).
- Asignación de $Fo2c1$ a la variable manipulada $u[1]$ y $Fo2c1$ a $u[2]$.
- Ecuación de la referencia interna.
- Ecuación de corrección del error de modelado (BIAS).
- El índice de coste a minimizar.
- La expresión *condicionFo2max*, que penaliza el hecho de que el flujo total supere el valor de 4.4 L/min, ya que en este caso se operaría fuera del rango de identificación del modelo.

Una vez compilado el código, se crea una partición seleccionando *refy* como variable de contorno y generando una clase asociada a la partición.

Los códigos de EcosimPro se encuentran en el Anexo I.

7.1.2. Archivo control.el

Inicialmente, se lleva a cabo la llamada a las funciones externas *optim_control* y *nagcd* [18]. Posteriormente, se define la clase *optimiz* asociada a la partición del archivo *coste.el* (bloque **CLASS**), por herencia de *coste_default*. En este bloque se declaran (**DECLS**) las variables:

- Horizonte de predicción ($H_p = 30$).
- Horizonte de control ($H_{con} = 3$).
- Período de muestreo ($per_muestreo = 0.2 \text{ min}$).
- Número de entradas ($nr_inputs = 2$).
- Límites en las variables manipuladas y en sus incrementos (para ambos colectores el flujo mínimo es 0.4 L/min y el máximo 4.4 L/min, mientras que los incrementos permitidos son de ± 0.1 L/min).
- La precisión en el cálculo de la función objetivo ($f_prec = 1. e-15$).
- El valor de la predicción ($y_prediccion$).

El siguiente bloque dentro de la clase es el conjunto de métodos (**METHODS**), que son las subrutinas asociadas a la definición de la clase. Dentro de este bloque se declaran los siguientes métodos:

- **METHOD NO_TYPE ini():**
Se inicia el control predictivo.
- **METHOD NO_TYPE objetivo():**
En el cuerpo del método, en primer lugar se resetean todas las variables a la situación inicial con la función `RESET_VARIABLES()`. Posteriormente se recupera el estado actual con la función `RESTORE_STATE()` y se ejecuta el bloque `INIT` del componente *coste* con la función `EXEC_INIT()`, donde se asignan los valores del estado actual a las variables manipuladas, la variable controlada y la perturbación medible, mientras que el índice a optimizar retoma el valor cero.
A continuación, se ejecuta el código del componente *coste* (optimización de la función de coste) desde el instante cero hasta el horizonte de predicción (N2), recogiendo los valores de:
 - El índice a optimizar (J)
 - Las funciones que definen las restricciones de camino
 - La predicción de la variable controlada ($y_prediccion$)
- **METHOD NO_TYPE control():**
Realiza la llamada al optimizador (`optim_control(objetivo, con)`), donde *con* es el vector de control. El algoritmo de optimización de la librería NAG es SQP.
- **METHOD REAL prediccion():**
Retorna el valor de la predicción de la variable controlada: $y_prediccion$.

Terminado el bloque **METHODS**, se finaliza la clase *optimiz* (**END CLASS**) y se inicia, a continuación, el componente *control* (**COMPONENT control**):

En el bloque **DATA** se definen los parámetros del modelo, el flujo de agua, el volumen, las fracciones molares de oxígeno en ambos colectores y la presión de operación. Además, se especifica el estado inicial (subíndice *inic*) mediante los valores en estacionario de los flujos de gas de ambos colectores, la temperatura y la concentración de oxígeno.

El período de muestreo (*tsamp*) se fija en 0.2 min (12 segundos).

En el bloque **DECLS** se declaran las siguientes variables:

- Variables del proceso: temperatura, concentración de oxígeno a la entrada, a la salida y en saturación, flujo de gas en ambos colectores, flujo total, fracción molar de oxígeno, constante de Henry y coeficiente de transferencia de materia.
- Concentraciones máxima y mínima, correspondientes a la saturación y la concentración cuando ambos colectores operan con el mínimo flujo (0.4 L/min), respectivamente.
- Los valores iniciales en estado estacionario definidos en *coste*.
- La consigna de concentración (*refy*).
- El error (*e*) que se define posteriormente como la diferencia entre la referencia y el valor de la concentración en cada instante.
- Las variables que devuelve el controlador (*u_new* para los flujos de los dos colectores).
- La llamada al controlador cada periodo de muestreo (**BOOLEAN** *sample* = **TRUE**).
- El valor predicho (*Valor_Predicho*) por el controlador.
- El error o BIAS (*Error*) definido posteriormente como diferencia entre el valor de la concentración y el valor predicho en cada instante.

Aquellas variables que no son calculadas posteriormente en el bloque **CONTINUOUS** ni son tomadas como variables de contorno se definen como **DISCR REAL** (*Valor_Predicho*, *Error*, *u_new*).

En el bloque **OBJECTS** se crea el objeto *modelo*, que hereda las propiedades de la clase *optimiz*.

En el bloque **INIT** se asigna a las variables (flujos en los colectores, temperatura, concentración y el valor predicho por el controlador) los valores iniciales en estacionario indicados en el bloque **DATA** (subíndice *inic*) y se inicia el controlador: *modelo.ini* () asignando los valores $u_new[1] = Fo2c1inic$, $u_new[2] = Fo2c2inic$, $refy = Co2outinic$.

En el bloque **DISCRETE** se hace la llamada al controlador predictivo. Cuando la variable booleana *sample* es cierta, es decir, cada periodo de muestreo, se guarda el estado actual, la temperatura, el valor de las variables manipuladas, el error (BIAS) y la referencia actual.

Se recoge la nueva señal de control (*modelo.control* (*u_new*)) y se asigna a las variables: $Fo2c1 = u_new[1]$, $Fo2c2 = u_new[2]$.

La variable booleana *sample* pasa a falso tras cada periodo de muestreo.

En el bloque **CONTINUOUS** se escriben todas las ecuaciones que relacionan las variables y los parámetros definidos anteriormente:

- Variación de la constante de Henry con la temperatura.
- Balance global e individual (al O₂) aplicado a los flujos de ambos colectores.
- Concentraciones de entrada y de saturación en función la fracción molar de O₂.
- Ecuación del modelo obtenido del coeficiente de transferencia de materia.
- Balance de materia al O₂ en fase líquida (ecuación diferencial).
- La concentración máxima se iguala a la de saturación.
- La concentración mínima se calcula a partir del balance de materia en fase líquida indicando 0.8 L/min en el flujo total de gas (flujo mínimo de cada colector = 0.4 L/min).
- El error como diferencia entre la referencia y la concentración de O₂.
- Asignación de las variables actuales (flujos, temperatura y concentración) para pasar sus valores al fichero *coste.el* (subíndice *i*).

Una vez compilado el código, se crea una partición seleccionando *Fo2c1*, *Fo2c2*, *T* y *refy* como variables de contorno.

7.1.3. Archivo controlador.el

Inicialmente se lleva a cabo la llamada a las funciones externas *optim_control* y *nagcd*. La clase asociada a la partición del archivo *coste.el* (bloque **CLASS**) contiene el mismo código que el definido en el archivo *contro.el*, ya explicado.

En **COMPONENT controlador**, se define el código para que el controlador lea las variables del estado actual y ejecute las nuevas acciones de control, a partir de variables de intercambio (*OPC_variable*).

En el bloque **DATA** se definen unos valores iniciales (subíndice *inic*) de flujo en ambos colectores, temperatura en el reactor y concentración de oxígeno correspondientes a un estado estacionario.

En el bloque **DECLS** se declaran las siguientes variables:

- La referencia de concentración (*set_point*).
- Las variables manipuladas por el controlador predictivo (los flujos de gas en ambos colectores *Fo2c1* y *Fo2c2*).
- Las variables de intercambio, que se leen vía OPC (*OPC_Co2out*, *OPC_Fo2c1*, *OPC_Fo2c2*, *OPC_T*).
- Los parámetros del controlador predictivo: *OPC_N2*, *OPC_Nu*.
- Las variables correspondientes al estado actual, que deben enviarse al modelo: *Fo2c1i*, *Fo2c2i*, *Ti*, *Co2outi*, *refy*.
- El periodo de muestreo (*tsamp* = 12 segundos). En este caso se especifica el tiempo en segundos para llevar a cabo el control de la planta en tiempo real.
- La variable booleana *sample*, que activa/desactiva la llamada al controlador cada periodo de muestreo.
- La variable entera *activate*, cuya función se explica más adelante.

En el bloque **OBJECTS** se crea el objeto *modelo*, que hereda las propiedades de la clase *optimiz*.

En el bloque **INIT** se inicia el controlador: *modelo.ini* () y se asignan los valores:

- $u_new[1] = Fo2c1inic$ y $u_new[2] = Fo2c2inic$.
- $Fo2c1 = Fo2c1inic$ y $Fo2c2 = Fo2c2inic$.
- $set_point = Co2outinic$.
- $OPC_Co2out = Co2outinic$, $OPC_Fo2c1 = Fo2c1inic$, $OPC_Fo2c2 = Fo2c2inic$ y $OPC_T = Tinic$.
- $OPC_N2 = 30$, $OPC_Nu = 3$.

En el bloque **DISCRETE** se realiza la llamada al controlador predictivo. Mediante el uso de la condición **WHEN** se especifican las acciones a realizar en función del valor de la variable entera *activate*, definida anteriormente en el bloque **DECLS**.

- Cada vez que se realiza una llamada al controlador (**sample** = **TRUE**), la variable *activate* toma el valor 0. Posteriormente *sample* pasa a **FALSE** después de cada llamada al controlador.

- Cuando *activate* toma el valor 1, se pasa al controlador el estado actual de las variables del proceso, asignando los valores actuales a las variables de intercambio ($Co2out_i = OPC_Co2out$, $Fo2c1_i = OPC_Fo2c1$, $Fo2c2_i = OPC_Fo2c2$, $T_i = OPC_T$, $refy = set_point$, $Hp = OPC_N2$, $H_con_comun = OPC_Nu$), se guarda el estado actual y la variable *activate* pasa a 2.
- Cuando la variable *activate* toma el valor 3, el controlador calcula la nueva señal de control ($modelo.control(u_new)$), pasando los nuevos valores a las variables manipuladas: $Fo2c1 = u_new[1]$, $Fo2c2 = u_new[2]$. El valor de *activate* pasa a 4.

Posteriormente, en la configuración del SCADA a partir del servidor OPC que se creará del presente código de Ecosim, debe realizarse la sincronización de lectura y envío de datos entre controlador y planta, ya que en el código *controlador.el* no está programada la ejecución cíclica de la lectura de las variables del proceso por parte del controlador y el envío de las nuevas señales de control cada periodo de muestreo. Para ello, por tanto, se define el siguiente grafcet a fin de realizar la sincronización, definiendo las condiciones de transición entre etapas:

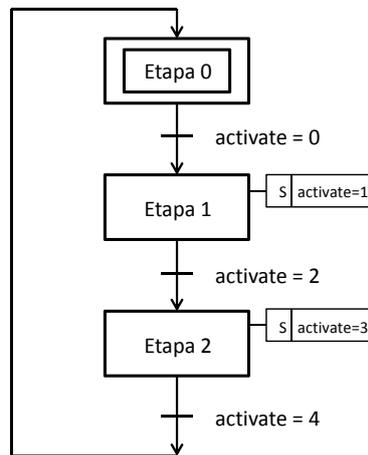


Figura 7.5. Grafcet de evolución de la variable *activate* para la sincronización de lectura/envío de datos

Inicialmente, se comienza con la etapa de espera, hasta que se cumple el periodo de muestreo, instante en que la variable *activate* toma el valor 1 (primera etapa del grafcet). En este momento, el controlador lee las variables actuales del proceso y las envía al código *coste*, donde está definida la función objetivo. Al final de esta etapa, *activate* pasa a valor 2, que es la condición de transición a la siguiente etapa del grafcet.

En la etapa 2, *activate* toma el valor 3, momento en que se lleva a cabo la optimización de la función objetivo, asignando las nuevas acciones de control, propuestas por el optimizador, a las variables manipuladas vía OPC. Posteriormente, *activate* pasa a 4 y se espera a que se cumpla un nuevo periodo de muestreo para volver al inicio del grafcet:

- Etapa 0: Espera.
- Etapa 1: El controlador lee las variables actuales vía OPC.
- Etapa 2: El controlador ejecuta las acciones de control.

7.2. SIMULACIÓN DEL CONTROLADOR EN ECOSIM

A continuación se muestran varias simulaciones del controlador predictivo aplicado a la simulación del proceso, suponiendo inicialmente que el modelo es idéntico al proceso y, posteriormente, introduciendo errores de modelado para comprobar la capacidad de corrección del error, en los siguientes casos:

- Sumando el error al valor de la variable de estado tras la integración de la función de coste.
- Sumando el error en el punto inicial de integración de la función de coste.

7.2.1. Modelo igual al proceso

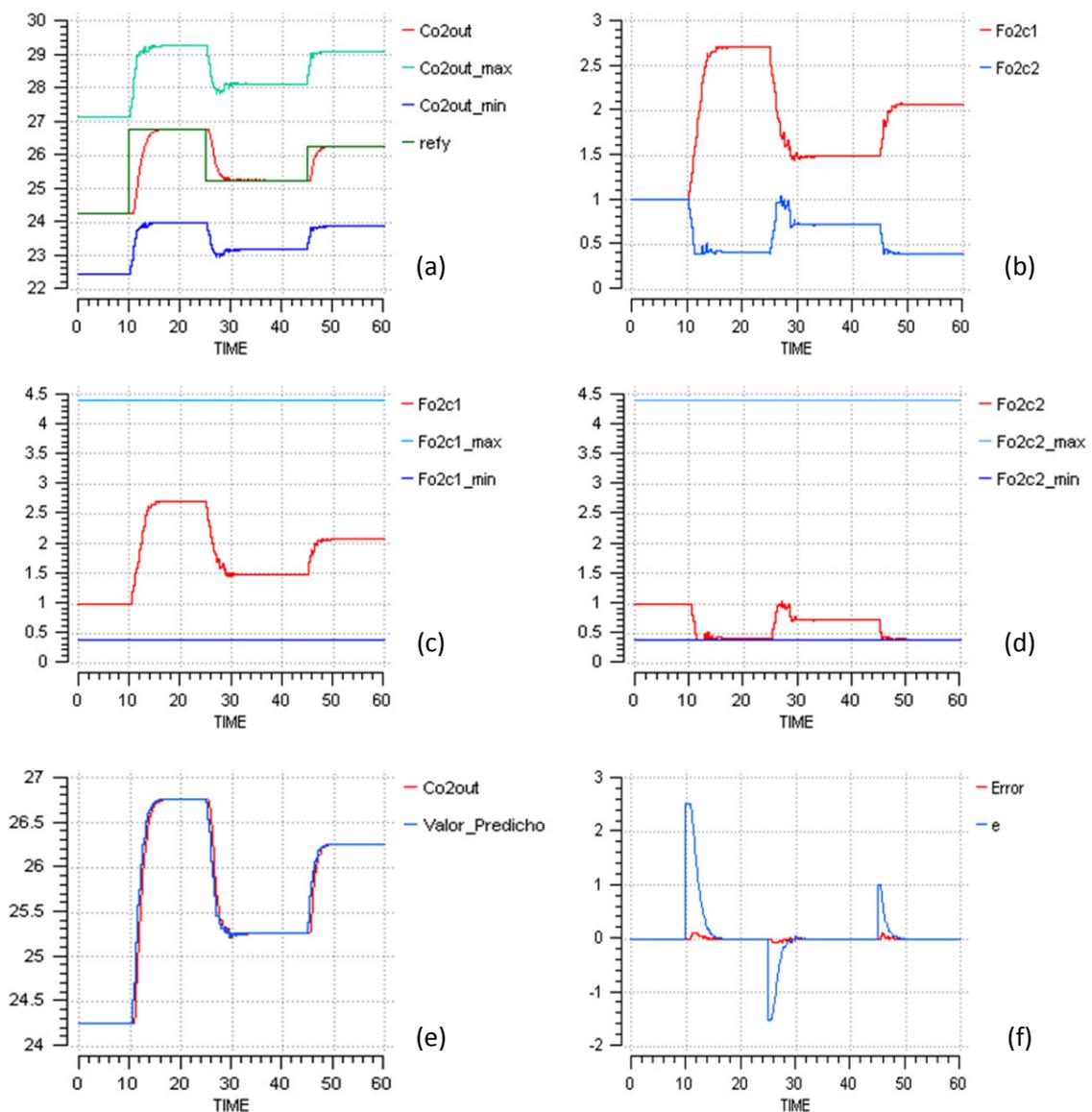


Figura 7.6. Simulación del controlador predictivo (modelo = proceso). (a) Evolución de la concentración de oxígeno ante saltos en la referencia, (b) Evolución de las variables manipuladas

La figura (a) muestra la evolución de la variable controlada respecto a saltos en la referencia, así como los límites: el superior marcado por la saturación (concentración máxima) y el inferior, que se da para el caso de flujo mínimo en ambos colectores. Dichos límites cambian al hacerlo el flujo de los colectores, ya que la fracción molar de oxígeno varía, y por tanto la saturación, según la ley de Henry.

La figura (b) muestra la evolución de las variables manipuladas, así como las figuras (c) y (d), en las que se incluyen los límites de operación. La figura (e) muestra la evolución del valor predicho por el modelo para la variable controlada y el valor real, cuya diferencia (Error) se muestra en la figura (f) en rojo, así como la diferencia entre el set point y la concentración del modelo, en azul.

Como se puede apreciar, cuando el modelo es idéntico al proceso, se alcanza siempre la referencia de manera exacta. No obstante, los modelos siempre incluyen errores, lo que conduce a que el proceso sea ligeramente distinto del modelo identificado. Por ello es necesario introducir un término de corrección del error entre el valor predicho por el modelo y el valor real medido. Dicho error puede sumarse tras la integración de la función de coste, o bien en el punto inicial de integración.

A continuación, se provocan distintos errores de modelado (cambiando los parámetros del modelo con cierto porcentaje de desviación respecto a los valores validados) y se comprueba la capacidad de corrección.

7.2.2. Modelo distinto al proceso

Cambiando el parámetro m: 1.2*m

- Sin corregir el error ($y[1] = \text{Co2out}$)

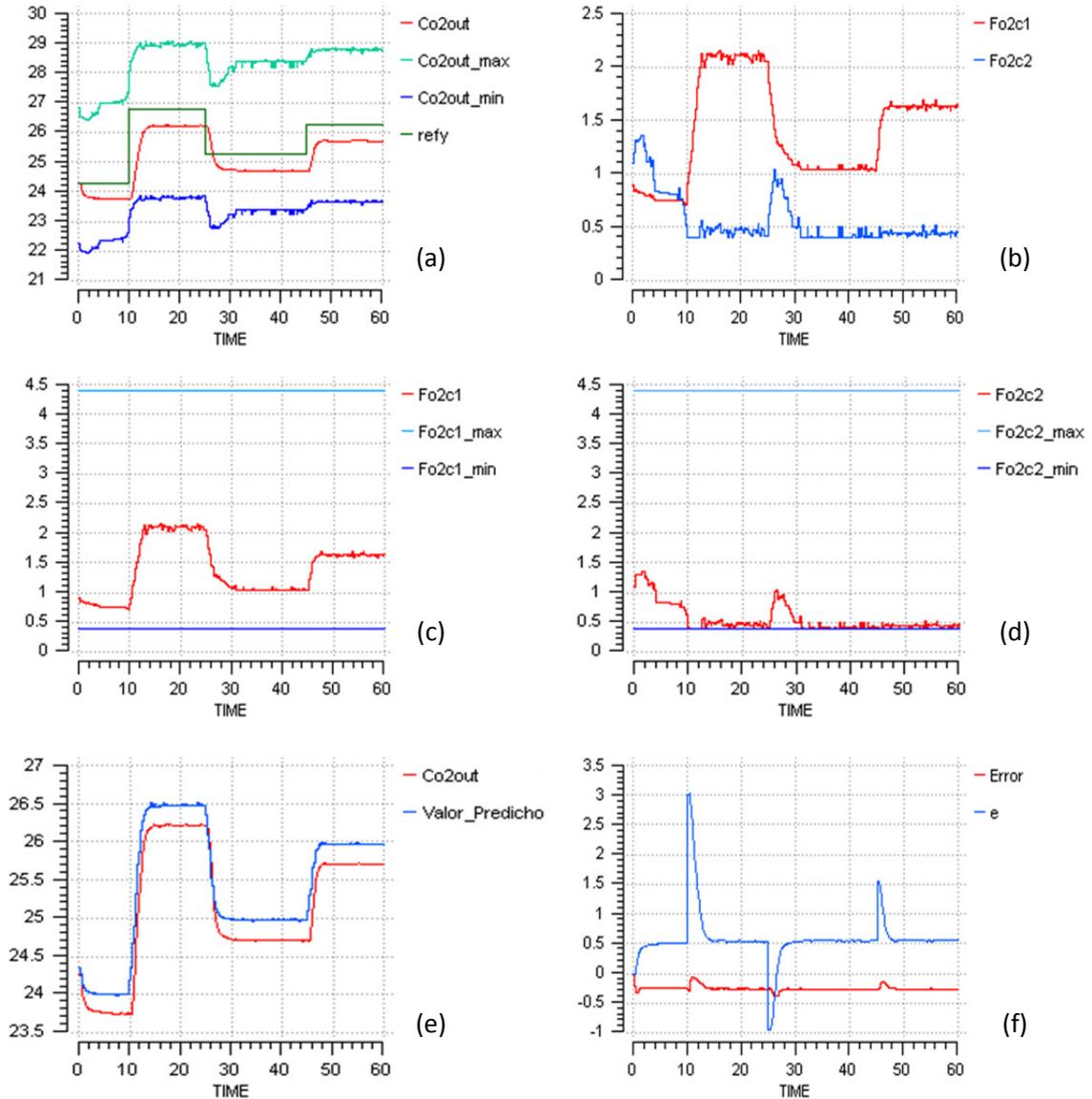


Figura 7.7. Simulación del controlador predictivo en EcosimPro. Error en el parámetro m (1.2*m), sin realizar corrección del error. (a) Evolución de la variable controlada, (b) Evolución de las variables manipuladas, (e) Diferencia entre valor predicho y valor real, (f) Error del modelo (línea roja)

▪ Corrigiendo el error ($y[1] = Co2out + Error$)

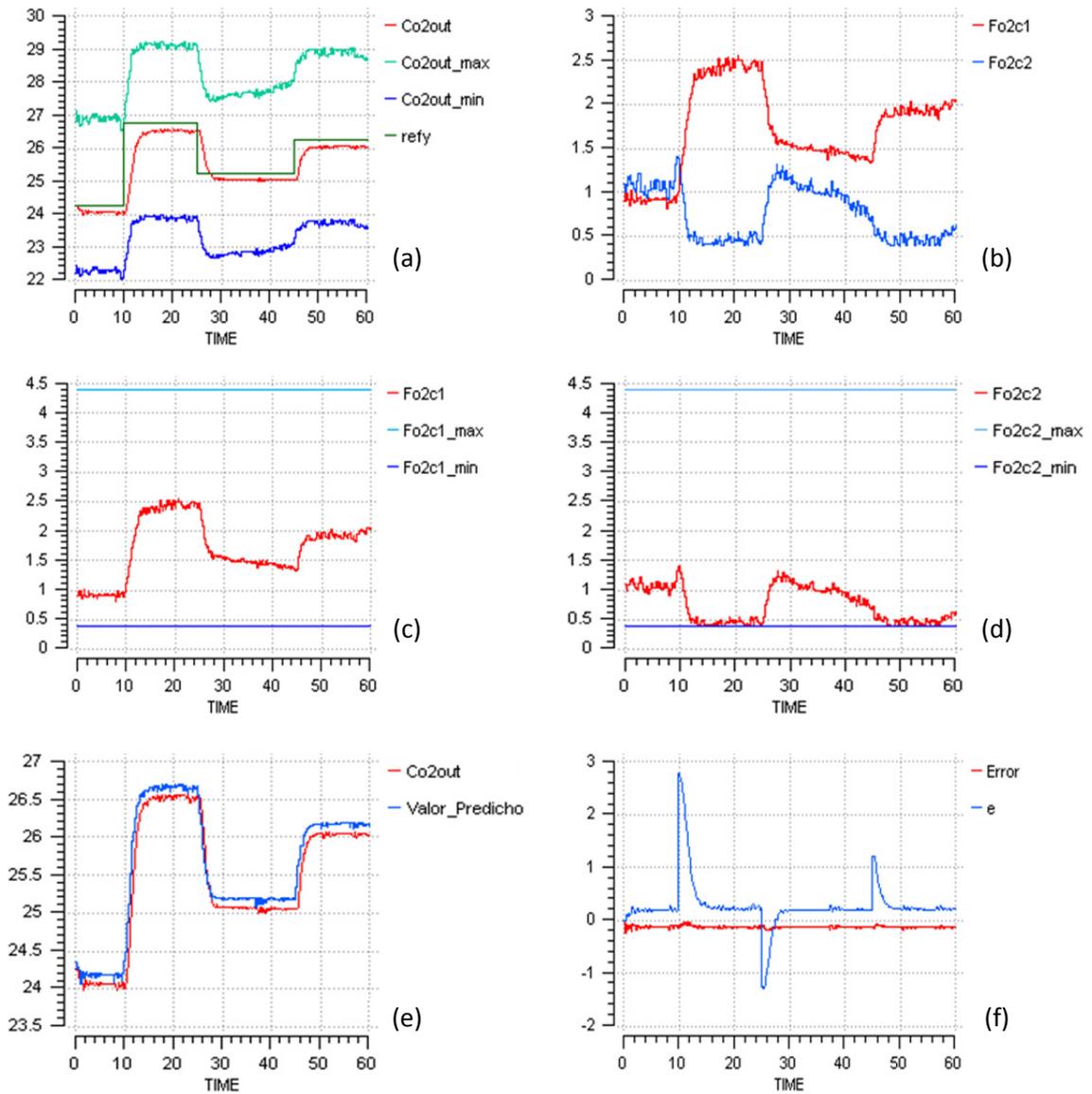


Figura 7.8. Simulación del controlador predictivo en EcosimPro. Error en el parámetro m (1.2*m), con corrección del error tras la integración de la función de coste. (a) Evolución de la variable controlada, (b) Evolución de las variables manipuladas, (e) Diferencia entre valor predicho y valor real, (f) Error del modelo (línea roja)

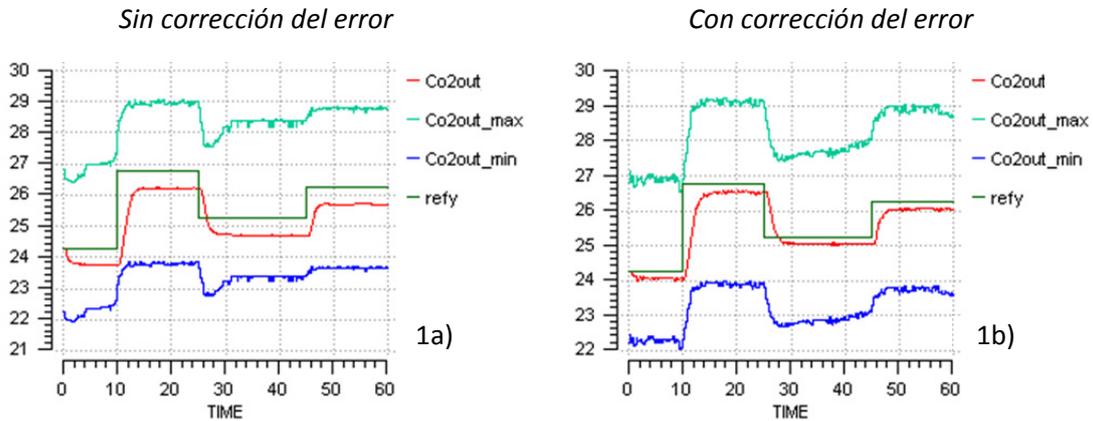
Se observa que disminuye significativamente el error estacionario (algo más de un 50%) al llevar a cabo la corrección del error tras la integración de la función de coste:

SP (ppm)	[O2] (ppm)		Error (%)	
	sin corrección	con corrección	sin corrección	con corrección
24.26	23.70	24.00	2.31	1.07
26.76	26.20	26.50	2.09	0.97
25.26	24.70	25.00	2.22	1.03
26.26	25.60	26.00	2.51	0.99

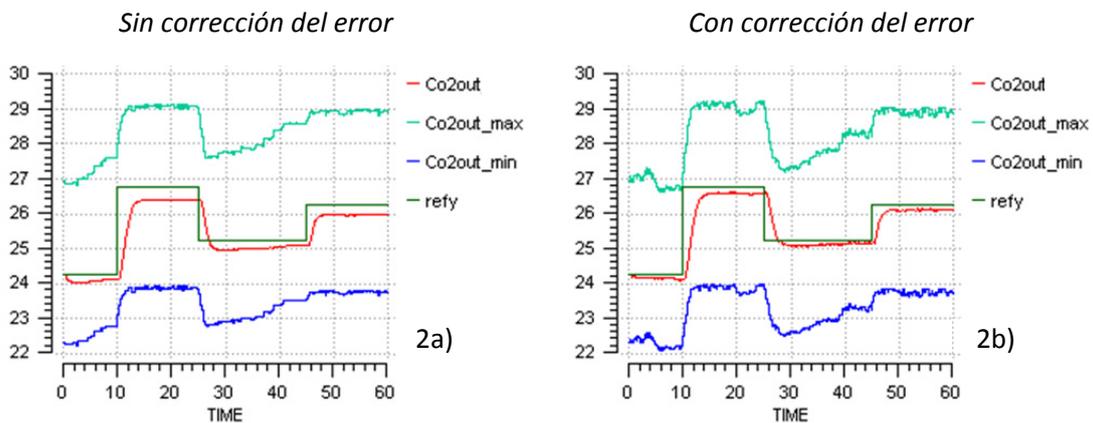
Tabla 7.1. Comparación del error estacionario con corrección y sin corrección del error

Igualmente, se simulan distintos errores del modelo cambiando otros parámetros:

Cambiando el parámetro m: $1.2 \cdot m$



Cambiando los parámetros: a: $1.2 \cdot a$; m: $0.9 \cdot m$; n: $0.8 \cdot n$



Cambiando k_{la}: $1.10 \cdot k_{la}$

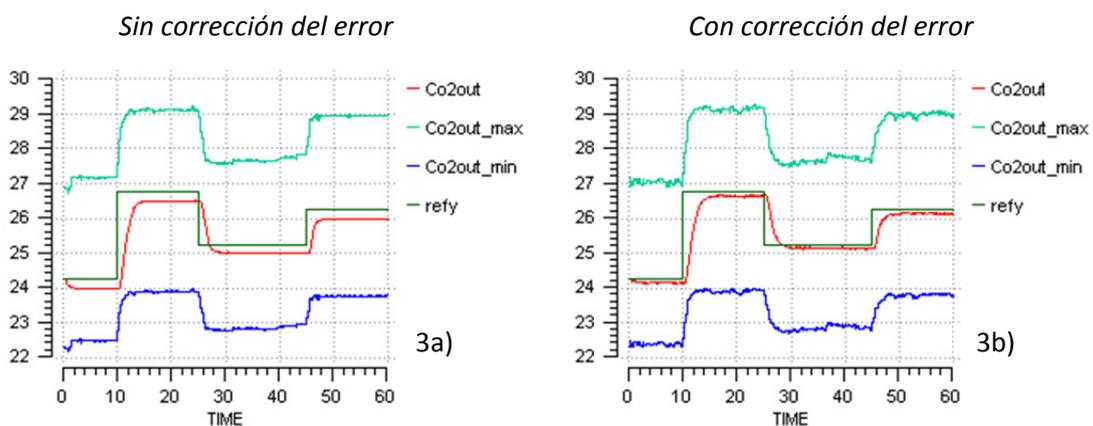


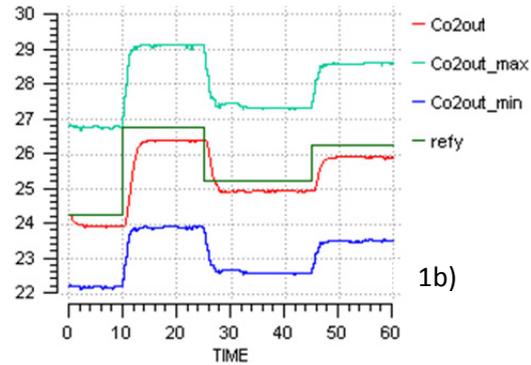
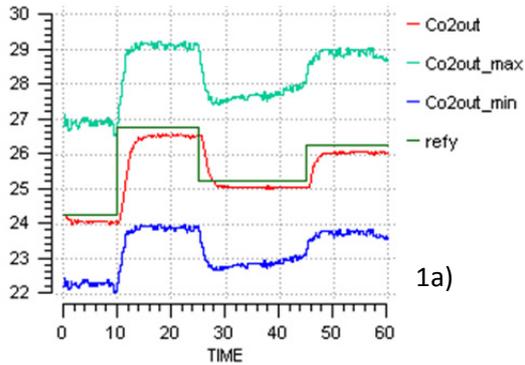
Figura 7.9. Simulación del controlador predictivo en EcosimPro. Comparación de la evolución de la variable controlada sin corrección del error (a) y con corrección del error tras la integración de la función de coste (b). 1) Error en el parámetro m ($1.2 \cdot m$). 2) Errores en los parámetros a ($1.2 \cdot a$), m ($0.9 \cdot m$) y n ($0.8 \cdot n$). 3) Error en la expresión del coeficiente de transferencia ($1.10 \cdot k_{la}$)

Otra opción es corregir el error en el punto inicial (en el bloque INIT del archivo coste, $\text{Co2out} = \text{Co2out}_i + \text{Error}$):

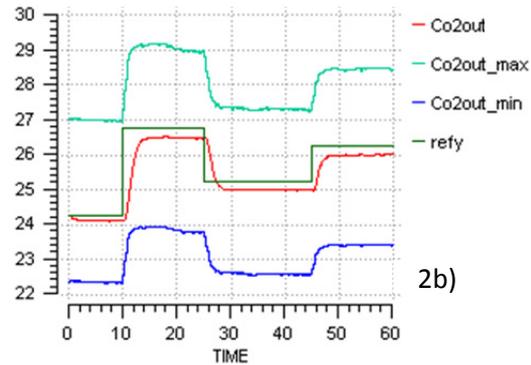
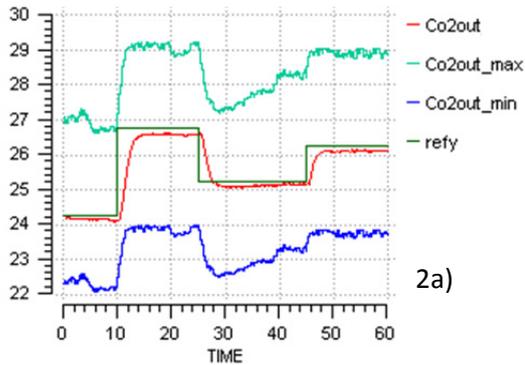
Corrección del error tras la integración

Corrección del error en INIT

Cambiando el parámetro m: $1.2 \cdot m$



Cambiando los parámetros: a: $1.2 \cdot a$; m: $0.9 \cdot m$; n: $0.8 \cdot n$



Cambiando k_{la}: $1.10 \cdot k_{la}$

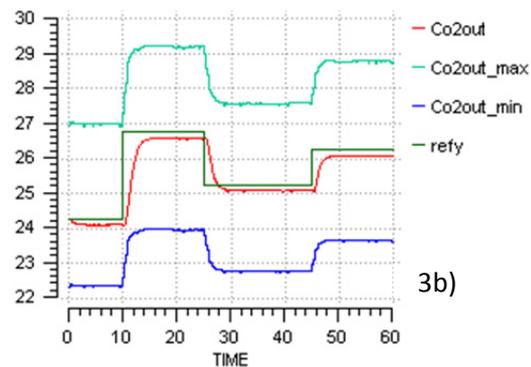
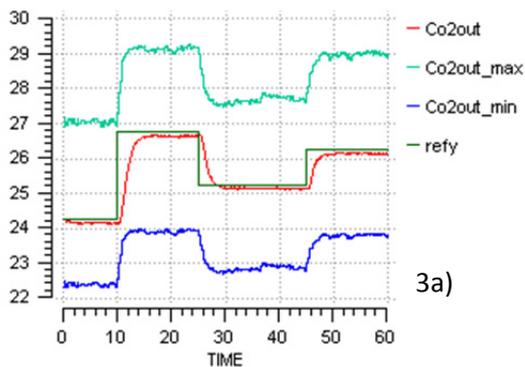


Figura 7.10. Simulación del controlador predictivo en EcosimPro. Comparación de la evolución de la variable controlada con corrección del error tras la integración (a) y en el punto inicial de la función de coste (b) para los errores de modelado: 1) Error en el parámetro m ($1.2 \cdot m$). 2) Errores en los parámetros a ($1.2 \cdot a$), m ($0.9 \cdot m$) y n ($0.8 \cdot n$). 3) Error en la expresión de k_{la} ($1.10 \cdot k_{la}$)

Se comprueba que el error estacionario es notablemente menor sumando el error tras la integración de la función de coste, en lugar de hacerlo en el punto inicial. Aun así no se alcanza la referencia de manera exacta.

7.3. MOVING HORIZON ESTIMATION

La estrategia para eliminar el error estacionario consiste en plantear un problema de optimización, paralelo al del control predictivo, que minimice el error entre la concentración predicha por el modelo ($C_{O_2}^{mod}$) y la concentración real medida del proceso ($C_{O_2}^{real}$) un número determinado de periodos de muestreo hasta llegar al actual [19], modificando un parámetro de perturbación del modelo como variable de decisión (d):

$$\min_d \sum_{k=1}^{N-1} (C_{O_2}^{mod}(t_k) - C_{O_2}^{real}(t_k))^2 \quad (7-13)$$

Siendo d una perturbación que se introduce al modelo multiplicando a la expresión ajustada del coeficiente de transferencia de materia $k_1 a$, es decir:

$$k_1 a = d \cdot ((T - n)^{0.5} \cdot m \cdot F_{O_2}^a) \quad (7-14)$$

De esta forma, se dispone de un grado de libertad más, permitiendo al modelo ajustarse al proceso real a través de este parámetro:

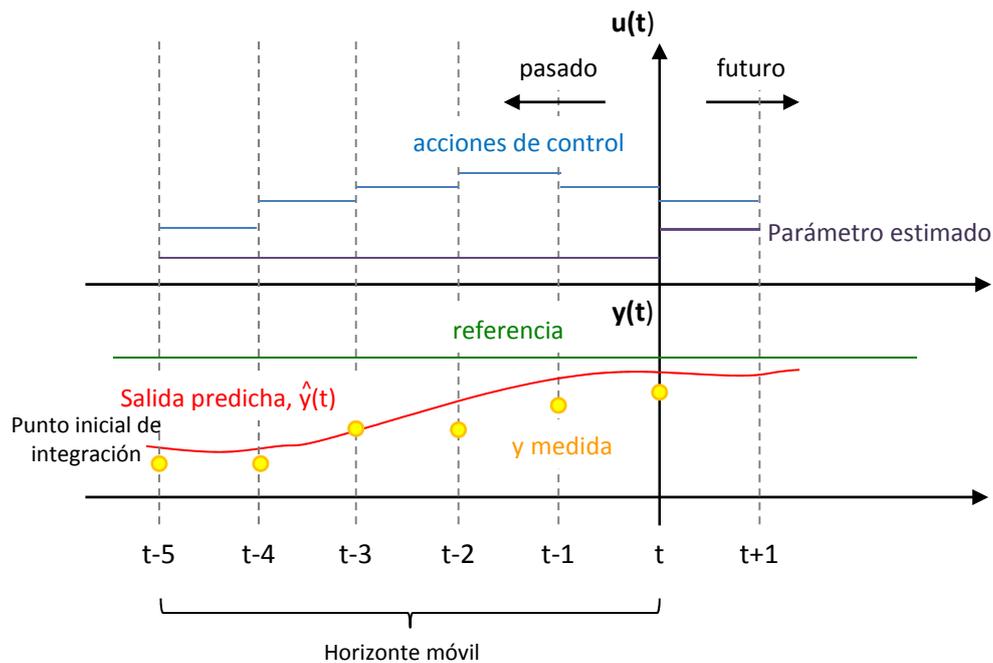


Figura 7.11. Esquema gráfico de la estrategia del horizonte móvil

Como se puede observar en la figura 7.11, debido a que el modelo no es perfecto, la medida de la variable controlada (en este caso la concentración) tiene cierta discrepancia respecto a la predicción del modelo. Esta estrategia busca detectar esa diferencia a lo largo de un número determinado de periodos de muestreo (horizonte móvil), que en este caso se ha fijado en 5. El objetivo es la minimización de la suma de los cuadrados de la diferencia entre el valor predicho por el modelo y el valor medido del proceso a lo largo de dicho horizonte, modificando como variable de decisión un parámetro de perturbación del modelo (d), que se aplica en el siguiente periodo de muestreo (horizonte de control = 1).

De esta forma lo que se está haciendo realmente es modificar el modelo a través del parámetro de perturbación no medible (d) a fin de que el modelo se ajuste lo más posible al proceso. Se ha decidido incluir el parámetro d multiplicando a la expresión ajustada del coeficiente de transferencia de materia k_1a , debido a que es conocido que dicha ecuación tiene implícitos errores de ajuste como puede comprobarse en el apartado 6 de identificación de la expresión de k_1a .

Como se ha esquematizado en la figura 7.11, se parte del punto inicial $t-5$ y se van guardando las variables del proceso (variable controlada, variables de control y perturbaciones medibles) en cada periodo de muestreo hasta llegar al instante actual t . En ese momento, se dispone de los datos suficientes para minimizar la función de coste, proponiendo un nuevo valor del parámetro de ajuste del modelo (d), que se aplica en $t+1$. El problema se repite a medida que el horizonte de estimación va avanzando cada periodo de muestreo.

La formulación del problema de control completo aplicando el MHE es:

$$\min_{F_{O_2C1}, F_{O_2C2}} J = \int_0^{t_f} \left((C_{O_2}(t) - w(t))^2 + \beta_1 \Delta F_{O_2C1}^2(t) + \beta_2 \Delta F_{O_2C2}^2(t) + \gamma \cdot \text{func}(t) \right) dt \quad (7-15)$$

$$\text{func}(t) = \max(F_{O_2C1}(t) + F_{O_2C2}(t) - F_{O_2}^{\max}, 0) \quad (7-16)$$

$$\min_d \sum_{k=1}^{N-1} (C_{O_2}(t_k) - C_{O_2}^{\text{real}}(t_k))^2 \quad (7-17)$$

$$\text{Sujeto a:} \quad \frac{dw(t)}{dt} = \frac{C_{O_2}^{\text{SP}}(t) - w(t)}{\alpha} \quad (7-18)$$

$$\text{Modelo:} \quad V_L \cdot \frac{dC_{O_2}}{dt} = k_1a \cdot V_L \cdot (C_{O_2}^{\text{sat}} - C_{O_2}) + F \cdot (C_{O_2}^{\text{in}} - C_{O_2}) \quad (7-19)$$

$$k_1a = d \cdot 0.7432 \cdot (T - 7.667)^{0.5} \cdot F_{O_2}^{0.6822} \quad (7-20)$$

$$C_{O_2}^{\text{sat}} = \left(\frac{P}{H} \right) \cdot y_{O_2} \quad (7-21)$$

$$C_{O_2}^{\text{in}} = \left(\frac{P_{\text{atm}}}{H} \right) \cdot 0.21 \quad (7-22)$$

$$H = 5 \cdot 10^{-4} \cdot T + 1.315 \cdot 10^{-2} \quad (7-23)$$

$$F_{O_2} = F_{O_2C1} + F_{O_2C2} \quad (7-24)$$

$$F_{O_2} \cdot y_{O_2} = F_{O_2C1} \cdot y_{O_2C1} + F_{O_2C2} \cdot y_{O_2C2} \quad (7-25)$$

$$\text{Restricciones:} \quad 0.4 \leq F_{O_2C1}(t) \leq 4.4; \quad 0.4 \leq F_{O_2C2}(t) \leq 4.4 \quad (7-26)$$

$$-0.1 \leq \Delta F_{O_2C1}(t) \leq 0.1; \quad -0.1 \leq \Delta F_{O_2C2}(t) \leq 0.1 \quad (7-27)$$

A continuación se explica la programación en Ecosim del problema de estimación.

7.4.1. Archivo coste MHE.el

Este código define la función objetivo para el cálculo de las estimaciones del parámetro de perturbación del modelo.

En el bloque **DATA** se definen los parámetros del modelo, el flujo de agua, el volumen, las fracciones molares de oxígeno en ambos colectores y la presión de operación. Además, se especifica el periodo de muestreo, los valores iniciales (subíndice i) de la perturbación no medible (d) y de la concentración de oxígeno.

Finalmente, se definen las variables medidas del proceso (flujos de ambos colectores, temperatura y concentración) a lo largo del número de periodos de muestreo correspondiente al horizonte para la estimación de la perturbación ($uf_Fo2c1[4]$, $uf_Fo2c2[4]$, $per_T[4]$, $per_CO2[4]$).

En el bloque **DECLS** se declaran las variables del proceso: temperatura, concentración de oxígeno a la entrada, a la salida y en saturación, flujo de gas en ambos colectores, flujo total, fracción molar de oxígeno, constante de Henry y coeficiente de transferencia de materia.

Además se declara como variable el parámetro de perturbación no medible (d), la función de coste a minimizar (J), la variable entera z para el bucle posterior, la variable booleana *Sample* y el valor medido de la concentración (*ValorMedido*).

Aquellas variables que no son calculadas posteriormente en el bloque **CONTINUOUS** ni son tomadas como variables de contorno se definen como **DISCR REAL** (T , $Fo2c1$, $Fo2c2$, d , J , *ValorMedido*)

En el bloque **INIT** se asignan valores iniciales a la variable controlada, el parámetro de perturbación no medible, la función de coste a minimizar ($J = 0$) y la variable entera para el bucle que se define en el bloque **DISCRETE** ($z = 0$).

En el bloque **DISCRETE** se define un bucle **WHEN** de tal forma que cuando se han cumplido los periodos de muestreo del *Moving Horizon Estimator* se calcula la función de coste J como sumatorio de los cuadrados de las diferencias entre el valor que predice el modelo y el valor medido del proceso, durante el horizonte de estimación.

Si aún no se han recogido los valores necesarios para iniciar la estimación (condición $z \leq 4$), se recogen los valores de las variables del proceso (flujos de ambos colectores, temperatura y concentración de oxígeno) hasta llenar la ventana para la estimación y poder comenzar a minimizar la función de coste.

De esta forma, se genera un vector de valores medidos cada periodo de muestreo:

$$\begin{pmatrix} F_{O_2C1}[z] \\ F_{O_2C2}[z] \\ T[z] \\ C_{O_2}[z] \end{pmatrix}, \text{ donde } z \text{ va desde } 1 \text{ hasta } 4.$$

En el bloque **CONTINUOUS** se definen las siguientes ecuaciones:

- Expresión de la constante de Henry en función de la temperatura.
- Balance global e individual (al O₂) aplicado a los flujos de ambos colectores.
- Concentraciones de entrada y de saturación en función la fracción molar de O₂.
- Expresión del coeficiente de transferencia de materia.
- Balance de materia al O₂ en fase líquida (ecuación diferencial).

Una vez compilado el código, al crear la partición se genera una clase asociada *estimacion* que hereda las propiedades del componente.

7.4.2. Archivo estimacion.el

En este código se define la clase asociada al componente *coste_MHE.el*, que hereda las propiedades de éste, cuyo objetivo es programar la minimización de la función objetivo del problema de estimación del parámetro de perturbación no medible.

En el bloque **DECLS** se define la matriz de valores (*matrizValores[4,5]*) formada por tantas columnas como variables de proceso medidas y tantas filas como se fije el tamaño de la ventana:

$$\begin{pmatrix} F_{O_2C1}[t_{k-4}] & F_{O_2C1}[t_{k-3}] & F_{O_2C1}[t_{k-2}] & F_{O_2C1}[t_{k-1}] & F_{O_2C1}[t_k] \\ F_{O_2C2}[t_{k-4}] & F_{O_2C2}[t_{k-3}] & F_{O_2C2}[t_{k-2}] & F_{O_2C2}[t_{k-1}] & F_{O_2C2}[t_k] \\ T[t_{k-4}] & T[t_{k-3}] & T[t_{k-2}] & T[t_{k-1}] & T[t_k] \\ C_{O_2}[t_{k-4}] & C_{O_2}[t_{k-3}] & C_{O_2}[t_{k-2}] & C_{O_2}[t_{k-1}] & C_{O_2}[t_k] \end{pmatrix}$$

Se definen también los vectores correspondientes a cada columna (*vec_Fo2c1[4]*, *vec_Fo2c2[4]*, *vec_T[4]* y *vec_CO2[4]*), el horizonte de control para la estimación ($H_{con} = 1$) y los límites en el parámetro de perturbación y en sus incrementos cada periodo de muestreo.

El siguiente bloque dentro de la clase es el conjunto de métodos (**METHODS**), que son las subrutinas asociadas a la definición de la clase. Dentro de este bloque se declaran los siguientes métodos:

- **METHOD NO_TYPE Actualizar():**
El objetivo de este método es la actualización de la matriz de valores con las variables del proceso a medida que la ventana va avanzando cada periodo de muestreo.
- **METHOD NO_TYPE DesplazarVentana():**
El objetivo de este método es la programación del desplazamiento de la ventana de modo que cada periodo de muestreo vaya tomando los valores de las variables del siguiente instante ($p=p+1$).
- **METHOD NO_TYPE ini():**
Se inicia el problema de estimación.

- **METHOD NO_TYPE objetivo():**
En el cuerpo del método, en primer lugar se resetean todas las variables a la situación inicial con la función `RESET_VARIABLES()`. Posteriormente se pasa el valor inicial de concentración al bloque `INIT` del código `coste_MHE` y se define el parámetro de perturbación como variable de decisión del problema de optimización. Se actualiza el vector de variables del proceso, se lleva a cabo la integración desde el estado inicial hasta cuatro veces el periodo de muestreo y se recoge el valor de la función de coste.
- **METHOD NO_TYPE EstimacionPerturbacion():**
Se realiza la llamada al optimizador para minimizar la función de coste. Dicha optimización se realiza llamando a la función `optim_control` de la librería NAG, que emplea un algoritmo SQP.

7.4.3. Archivo modelo control.el

Inicialmente, se lleva a cabo la llamada a las funciones externas `optim_control` y `nagcd` [18]. Posteriormente, se define la clase `optimiz` asociada a la partición del archivo `coste.el` (bloque **CLASS**), por herencia de `coste_default`. En este bloque se declaran (**DECLS**) las variables:

- Horizonte de predicción ($H_p = 30$).
- Horizonte de control ($H_{con} = 3$).
- Período de muestreo ($per_muestreo = 0.5 \text{ min}$).
- Número de entradas ($nr_inputs = 2$).
- Límites en las variables manipuladas y en sus incrementos (para ambos colectores el flujo mínimo es 0.4 L/min y el máximo 4.4 L/min, mientras que los incrementos permitidos son de $\pm 0.1 \text{ L/min}$).
- La precisión en el cálculo de la función objetivo ($f_prec = 1. e-15$).
- El estado actual para el cálculo de las acciones de control futuras (perturbación no medible, concentración, flujos de ambos colectores, temperatura y referencia), que posteriormente se pasa al código `coste.el`.

El siguiente bloque dentro de la clase es el conjunto de métodos (**METHODS**), que son las subrutinas asociadas a la definición de la clase. Dentro de este bloque se declaran los siguientes métodos:

- **METHOD NO_TYPE ini():**
Se inicia el control predictivo.
- **METHOD NO_TYPE objetivo():**
En el cuerpo del método, en primer lugar se resetean todas las variables a la situación inicial con la función `RESET_VARIABLES()`. Posteriormente se recupera el estado actual de las variables del proceso y se ejecuta el bloque `INIT` del

componente *coste* con la función EXEC_INIT(), donde se asignan los valores del estado actual a las variables del proceso, mientras que el índice a optimizar retoma el valor cero.

A continuación, se pasa el horizonte de control y de predicción al modelo y se realiza la integración hasta el horizonte de predicción, recogiendo el valor de la función objetivo.

➤ **METHOD NO_TYPE control():**

Realiza la llamada al optimizador (*optim_control(objetivo, con)*), donde *con* es el vector de control. El algoritmo de optimización de la librería NAG es SQP.

Terminado el bloque **METHODS**, se finaliza la clase *optimiz* (**END CLASS**) y se inicia, a continuación, el componente *modelo_control* (**COMPONENT modelo_control**):

En el bloque **DATA** se definen los parámetros del modelo, el flujo de agua, el volumen, las fracciones molares de oxígeno en ambos colectores y la presión de operación. Además, se especifica el estado inicial (subíndice *inic*) mediante los valores en estacionario de los flujos de gas de ambos colectores, la temperatura y la concentración de oxígeno. El periodo de muestreo (*tsamp*) se fija en 0.5 min (30 segundos) y se define el tamaño de la ventana igual a 5, para el problema de estimación.

En el bloque **DECLS** se declaran las siguientes variables:

- Variables del proceso: temperatura, concentración de oxígeno a la entrada, a la salida y en saturación, flujo de gas en ambos colectores, flujo total, fracción molar de oxígeno, constante de Henry y coeficiente de transferencia de materia.
- La perturbación no medible (*d*).
- Los valores iniciales en estado estacionario definidos en *coste*, incluyendo el valor inicial de la perturbación no medible (*d_i*).
- Concentraciones máxima y mínima, correspondientes a la saturación y la concentración cuando ambos colectores operan con el mínimo flujo (0.4 L/min), respectivamente.
- El error (*e*) que se define posteriormente como la diferencia entre la referencia y el valor de la concentración en cada instante.
- La consigna de concentración (*refy*).
- Las variables que devuelve el controlador (*u_new* para los flujos de los dos colectores) y la perturbación no medible calculada en el problema de estimación (*d_new*).
- La llamada al controlador cada periodo de muestreo (**BOOLEAN** *sample* = **TRUE**).
- La variable booleana *ActivacionEstimacion* y la variable entera *posicion* para el problema de estimación.

Aquellas variables que no son calculadas posteriormente en el bloque **CONTINUOUS** ni son tomadas como variables de contorno se definen como **DISCR REAL** (*d, u_new, d_new*).

En el bloque **OBJECTS** se crean los objetos *modelo*, que hereda las propiedades de la clase *optimiz*, y *Estimacion*, que hereda las propiedades de la clase *estimacion*.

En el bloque **INIT** se asigna a las variables del proceso los valores iniciales en estado estacionario indicados en el bloque **DATA** (subíndice *inic*) y se pasan las variables manipuladas del problema de control ($u_new[1] = Fo2c1inic$, $u_new[2] = Fo2c2inic$), la referencia ($refy = Co2outinic$) y la variable manipulada del problema de estimación ($d_new[1] = dinic$). Además, la variable entera *posicion* comienza en 1.

En el bloque **DISCRETE** se hace la llamada al controlador predictivo y al estimador del parámetro de perturbación no medible:

➤ Estimación de la perturbación no medible:

Cuando la posición es menor que el tamaño de la ventana, es decir, no se dispone de los valores suficientes para iniciar el problema de estimación, se lleva a cabo la actualización de la ventana, incrementando el valor de la posición una unidad cada periodo de muestreo.

Una vez que se ha completado la ventana, se activa la variable booleana *ActivacionEstimacion*, se desplaza la ventana y se ejecuta el problema de estimación, generando un nuevo valor de perturbación no medible (d_new) cada periodo de muestreo, a partir de la optimización de la función de coste definida en el archivo *coste_MHE.el*.

➤ Control predictivo:

Se realiza la llamada al controlador predictivo (*modelo.ini()*) recogiendo las nuevas señales de control (*modelo.control(u_new)*) y asignándolas a las variables manipuladas: $Fo2c1 = u_new[1]$, $Fo2c2 = u_new[2]$.

La variable booleana *sample* conmuta a falso tras cada periodo de muestreo.

En el bloque **CONTINUOUS** se escriben las ecuaciones que simulan el comportamiento del proceso:

- Variación de la constante de Henry con la temperatura.
- Concentraciones de entrada y de saturación en función la fracción molar de O₂.
- Ecuación del modelo obtenido del coeficiente de transferencia de materia.
- Balance global e individual (al O₂) aplicado a los flujos de ambos colectores.
- Balance de materia al O₂ en fase líquida (ecuación del modelo).
- La concentración máxima se iguala a la de saturación.
- La concentración mínima se calcula a partir del balance de materia en fase líquida indicando 0.8 L/min en el flujo total de gas (flujo mínimo de cada colector = 0.4 L/min).
- El error como diferencia entre la referencia y la concentración de O₂.

Finalmente se guarda el estado actual del proceso:

$$\begin{aligned} Fo2c1i &= Fo2c1; Fo2c2i = Fo2c2 \\ Ti &= T \\ Co2outi &= Co2out \\ d_i &= d \end{aligned}$$

7.4.4. Archivo controlador2.el

En el archivo *controlador2.el*, para tener en cuenta el problema de estimación de la perturbación no medible, se modifica el código *controlador.el*, visto anteriormente, como se especifica a continuación:

- En el bloque **DATA** se incluye la definición del tamaño de la ventana (*tamanoVentana = 5*).
- En el bloque **DECLS** se añade la variable *d_new*, que es la variable de decisión del problema de estimación para minimizar la diferencia entre el valor medido y el predicho por el modelo. Además se incluye la variable booleana *ActivacionEstimacion* y la variable entera *posicion* para coordinar el hecho de que cuando se haya llenado la ventana (*posicion > tamanoVentana*), la variable *ActivacionEstimacion* conmuta a TRUE.
- En el bloque **INIT** se comienza con un valor inicial del parámetro de perturbación *d_new = 1*. La variable booleana *ActivacionEstimacion* comienza en FALSE.
- En el bloque **DISCRETE**, cuando *activate* toma el valor 1 (lectura del estado actual del proceso), si la posición es menor que el tamaño de la ventana se ejecuta la actualización; mientras que si es mayor, *ActivacionEstimacion* conmuta a TRUE y se permite el desplazamiento de la ventana cada periodo de muestreo. Cuando *activate* toma el valor 3 (cálculo de las acciones de control), si se cumple la condición *ActivacionEstimacion = TRUE*, se ejecuta el problema de estimación y se recoge el nuevo valor de la perturbación no medible *d_new*.

7.4.5. Simulación del controlador predictivo con el problema de estimación

En la siguiente figura se muestra una simulación en la que inicialmente, al ser el modelo idéntico al proceso, la referencia se alcanza de manera exacta (figura 7.12 a). Posteriormente, en el minuto 5, se simula una perturbación cambiando el valor del parámetro m de la expresión del coeficiente de transferencia de materia k_1a (figura 7.12 b). Como consecuencia, al cambiar el modelo, la variable controlada se desvía de la referencia; sin embargo, al aplicar la técnica del estimador de horizonte móvil, el parámetro de perturbación del modelo, d , es modificado por el problema de optimización planteado, llegando a ajustar el modelo a la perturbación simulada y alcanzando de nuevo la referencia en la variable controlada.

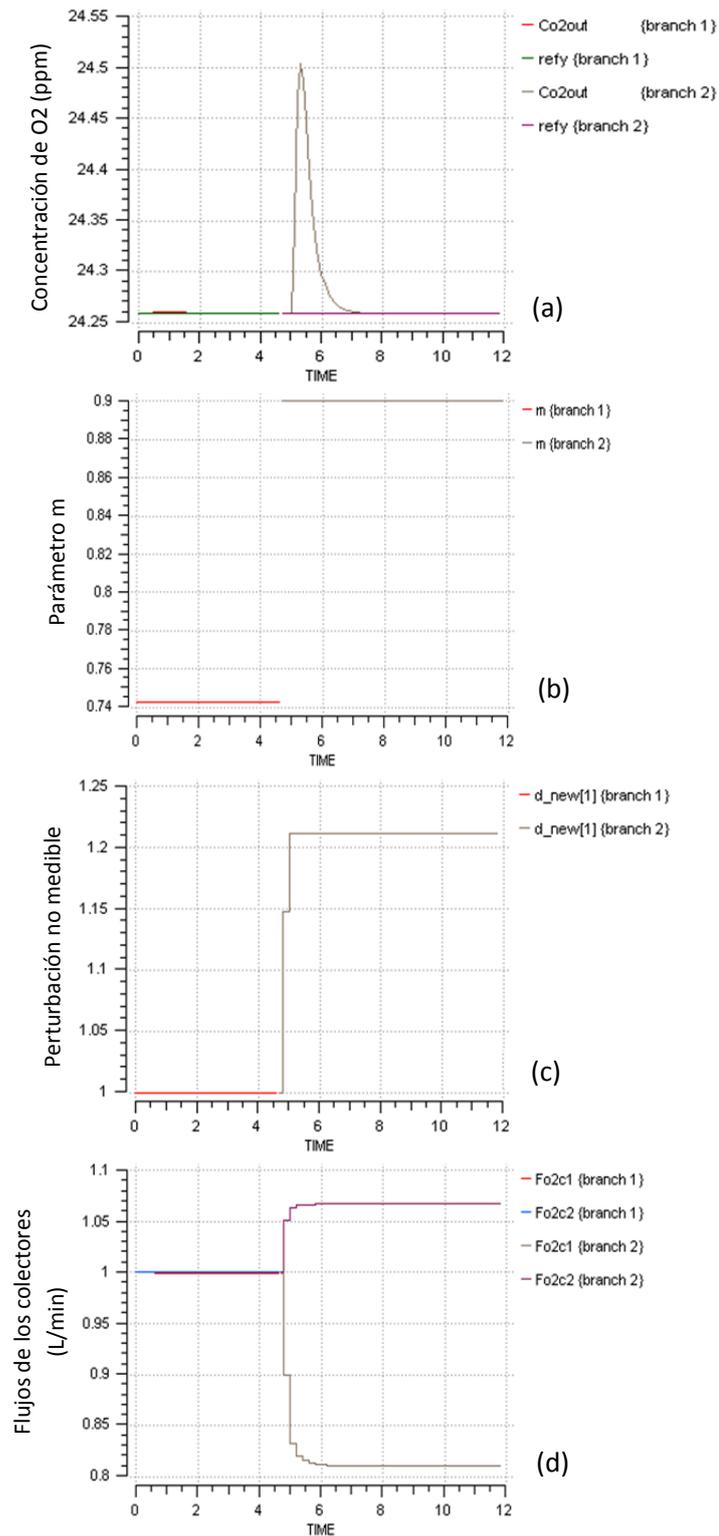


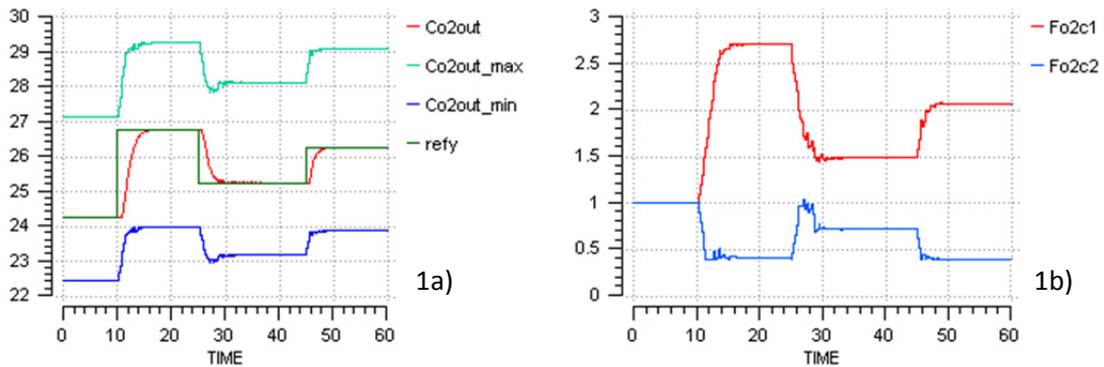
Figura 7.12. Simulación del controlador predictivo en EcosimPro, aplicando el método de estimación de horizonte móvil (Moving Horizon Estimation). (a) Evolución de la variable controlada, (b) Perturbación simulada, (c) Evolución del parámetro de perturbación no medible, (d) Evolución de los flujos de gas

7.4. SINTONÍA DEL CONTROLADOR

- $\alpha = 0.3$
- $N_u = 3$
- $N_2 = 30$

7.4.1. Alpha

Alpha = 0.3:



Alpha = 0.7:

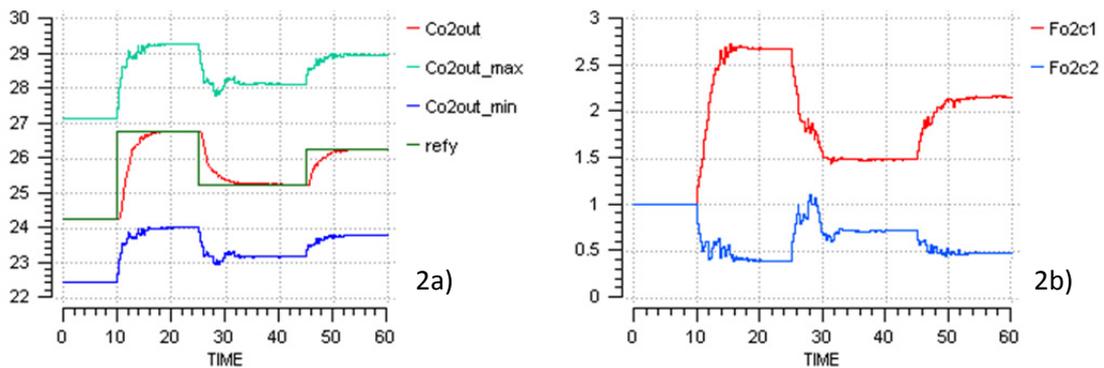


Figura 7.13. Simulación del controlador en EcosimPro. Sintonía del parámetro alpha. 1) $\alpha = 0.3$; 2) $\alpha = 0.7$. (a) Variable controlada, (b) Variables manipuladas

Como se puede observar, al aumentar el valor de alpha (factor de velocidad de la trayectoria de referencia) disminuye la velocidad de la respuesta. Por ejemplo, ante un salto positivo de 2.5 ppm en la concentración de referencia, en caso de $\alpha = 0.3$ la consigna se alcanza en 5 min, mientras que con $\alpha = 0.7$, se alcanza en 8 minutos.

Se selecciona un valor de $\alpha = 0.3$.

7.4.2. Horizonte de control

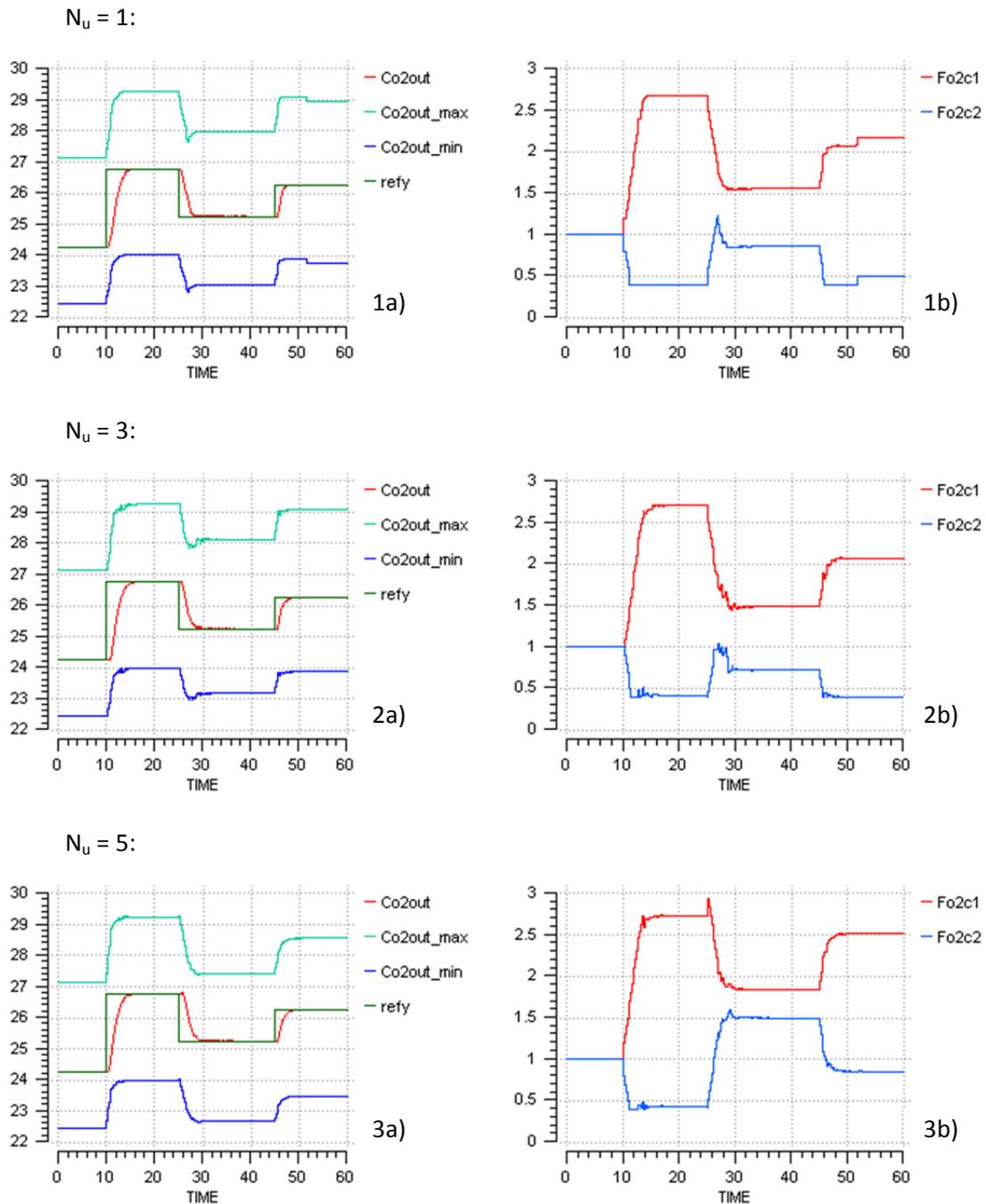


Figura 7.14. Simulación del controlador en EcosimPro. Sintonía del horizonte de control (N_u). 1) $N_u = 1$; 2) $N_u = 3$; 3) $N_u = 5$. (a) Variable controlada, (b) Variables manipuladas

La desventaja de aplicar un horizonte de control muy alto es que conlleva un mayor tiempo computacional. Por otra parte, no varía significativamente la respuesta del sistema.

Se selecciona un horizonte de control de 3.

7.4.3. Horizonte de predicción

En cuanto a la elección del horizonte de predicción (N_2) se toma como criterio el tiempo que tarda en alcanzar el estado estacionario, de modo que $N_2 * t_{\text{sample}}$ debe ser aproximado al tiempo de respuesta del sistema desde que se produce un salto hasta que la variable controlada toma un valor estacionario.

Con un tiempo de muestreo de 12 segundos, apto para captar adecuadamente la dinámica del sistema, teniendo en cuenta que ante cambios de flujo de magnitud entre 1 y 3 L/min se alcanza el nuevo estado estacionario en lazo abierto en 4 - 6 minutos, se selecciona $N_2 = 30$, de modo que $N_2 * t_{\text{sample}} = 6 \text{ min.}$

8. SIMULACIÓN EN EL SCADA EDUSCA

8.1. CREACIÓN DEL SERVIDOR OPC DE LA SIMULACIÓN DE LA PLANTA

CreaOPC [20] permite crear un servidor OPC a partir de un experimento realizado en Ecosim, añadiendo los ficheros .h, .stab y .spa del experimento. OPC (*OLE for Process Control*) es un estándar de comunicación en el campo del control y supervisión de procesos industriales, basado en la tecnología Microsoft OLE/COM/DCOM [21]. Ofrece una interfaz común para comunicación que permite que componentes software individuales interactúen y compartan datos a través de una arquitectura cliente-servidor.

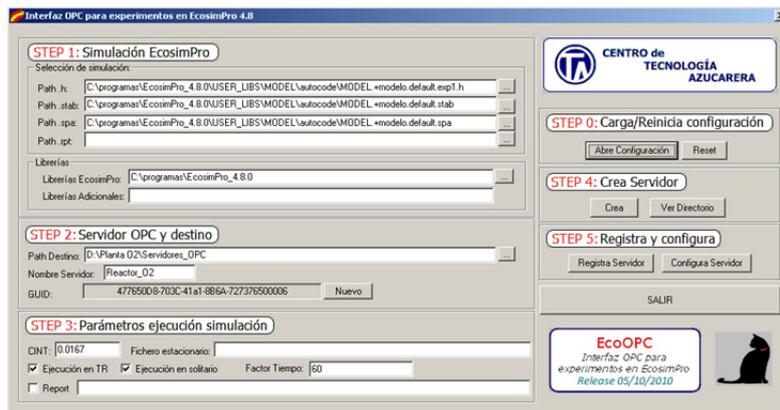


Figura 8.1. Creación del servidor OPC del modelo

Debido a que el la unidad de tiempo del modelo es el minuto, en los parámetros de ejecución de la simulación se indica un intervalo de comunicación de 0.0167, es decir 1/60 min/s y un factor de tiempo de 60, seleccionando la casilla de “Ejecución en Tiempo Real”, lo que permitirá simular el comportamiento del modelo del mismo modo que lo haría el proceso, según la identificación llevada a cabo en el apartado 6.

En la configuración se añade el grupo “Reactor”, en el que se añaden las variables asociadas al experimento. Posteriormente, se crea y se registra el servidor.

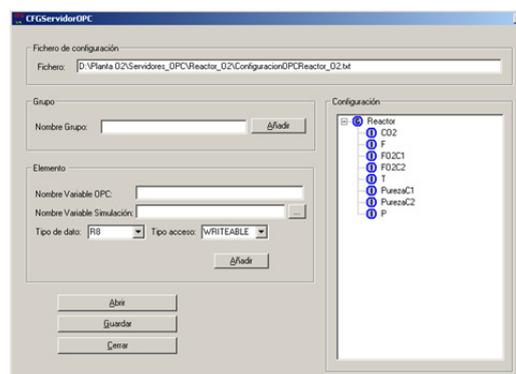


Figura 8.2. Configuración del servidor OPC del modelo

En Softing OPC se puede monitorizar la simulación del proceso:

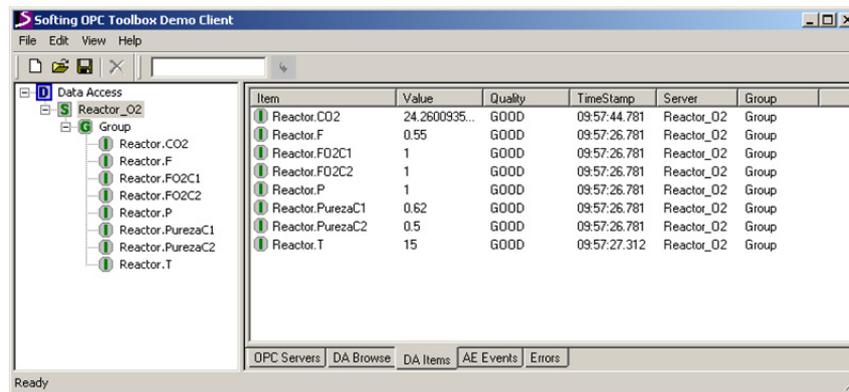


Figura 8.3. Simulación del proceso en Softing OPC

8.2. CONFIGURACIÓN DE LA SIMULACIÓN EN EL SACADA

Las variables manipuladas y las perturbaciones medibles se introducen como datos de lectura/escritura para poder manipular dichas variables en la simulación, mientras que la variable controlada y las variables que han sido fijadas como datos constantes (la presión y la pureza en los colectores) se introducen como variables de supervisión.



Figura 8.4. Configuración del SCADA para la simulación del proceso. Datos

Por ejemplo, la concentración de oxígeno se introduce como dato de supervisión, ya que es la variable que se desea controlar, especificando:

- El nombre de la variable (CO2)
- Las unidades (ppm)
- Los límites de representación (de 0 a 40 ppm)
- El tiempo de refresco (1 segundo)
- La llamada a la variable de OPC (\\localhost\Reactor_O2\Reactor.CO2)

The 'Tipo Supervisión' dialog box is used for configuring monitoring data. It includes the following fields and values:

- Datos Generales:**
 - Identificador: CO2
 - Descripción: Concentración de O2
 - Unidades: ppm
 - Conversion de: ppm
- Límites de representación:**
 - Inferior: 0
 - Superior: 40
- Límites de alarmas:**
 - Inferior: 0
 - Superior: 100
- Datos relativos a tiempos:**
 - Número de minutos a almacenar en el histórico: 600
 - Tiempo de refresco de dato (ms): 1000
- Datos OPC:**
 - Nombre variable: \\localhost\Reactor_02\Reactor.CO2

Figura 8.5. Dato de supervisión

Los flujos de gas en los colectores se introducen como datos de lectura/escritura, para poder modificar sus valores dando saltos en lazo abierto al igual que se haría en la planta real. Se introducen las unidades correspondientes (L/min), el tiempo de refresco (1 segundo) y la referencia a la variable de OPC.

The 'Dato Lectura' dialog box is used for configuring reading/writing data. It includes the following fields and values:

- Datos Generales:**
 - Identificador: F02C1
 - Descripción: Flujo del colector 1
 - Unidades: l/min
 - Conversion de: l/min
- Datos relativos a tiempos:**
 - Tiempo de refresco de dato (ms): 1000
- Datos OPC:**
 - Nombre variable: \\localhost\Reactor_02\Reactor.F02C1

Figura 8.6. Dato de lectura/escritura

En el apartado Sinópticos, se selecciona la imagen que mostrará el SCADA y se introducen los indicadores de las variables, situándolos en el esquema convenientemente:

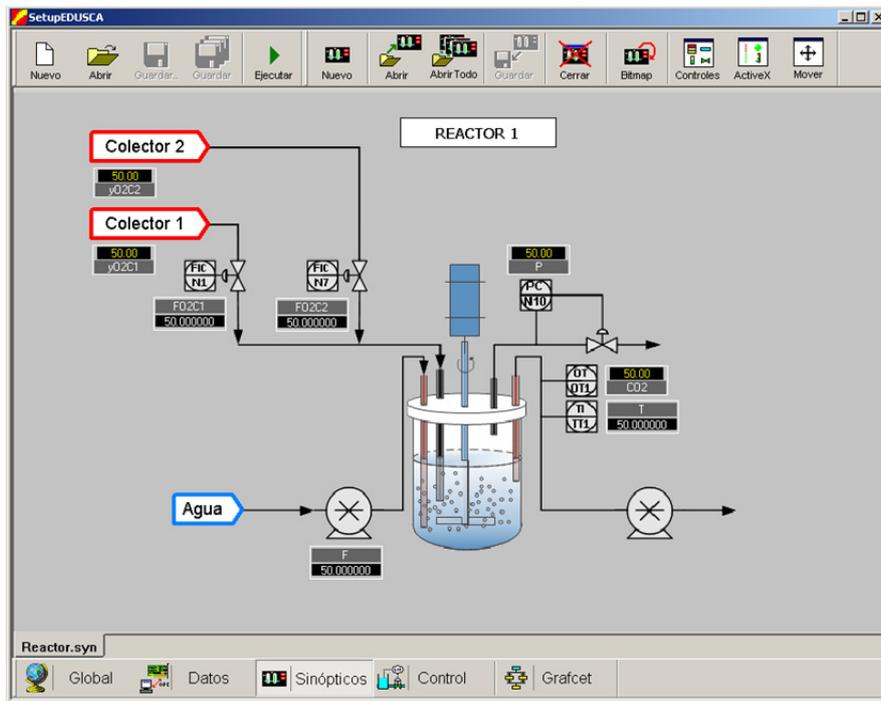


Figura 8.7. Configuración del sinótico de la simulación del proceso

Ejecutando la configuración guardada del SCADA, se pueden dar saltos a los flujos de ambos colectores y la temperatura, simulando el comportamiento real de la planta, al igual que en Ecosim:

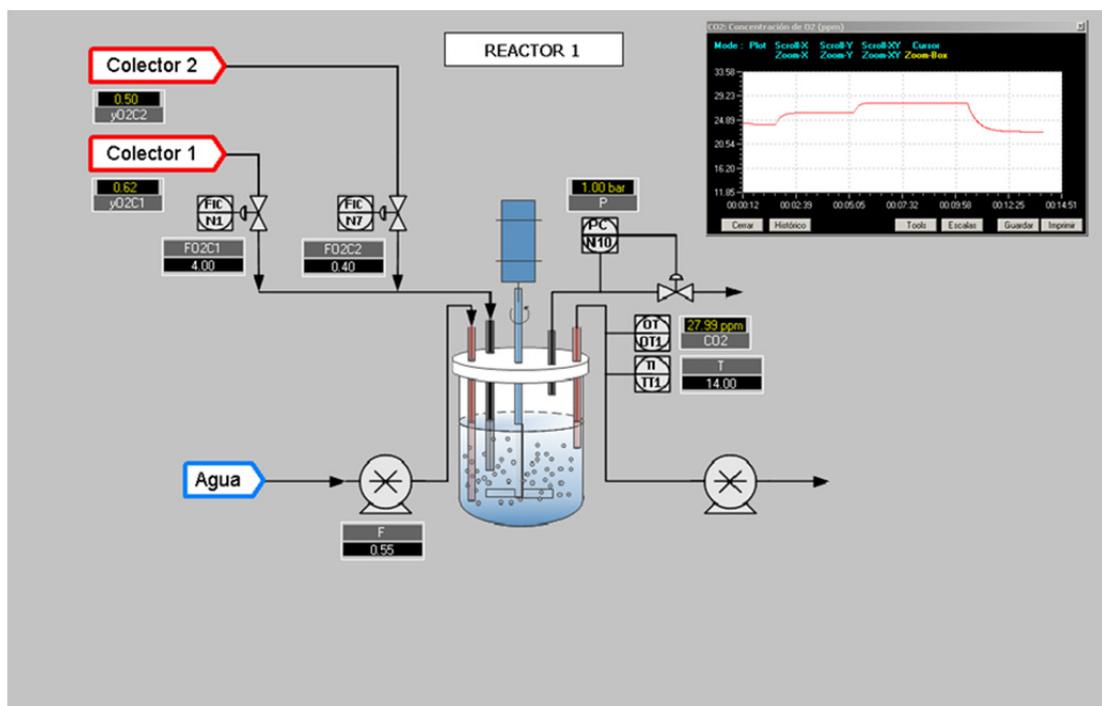


Figura 8.8. Simulación del proceso en lazo abierto en el SCADA

8.3. CREACIÓN DEL SERVIDOR OPC DEL CONTROL DE LA PLANTA

Con CreaOPC se genera un servidor OPC del controlador de la planta a partir del archivo *controlador.el* de Ecosim. En este caso es necesario añadir las librerías *optim_control* y *nagcd* como librerías adicionales, ya que son empleadas para la optimización de la función de coste. Además, en el directorio destino, se añade el fichero *.stab* del experimento del código *coste*, que contiene la definición de la función de coste.

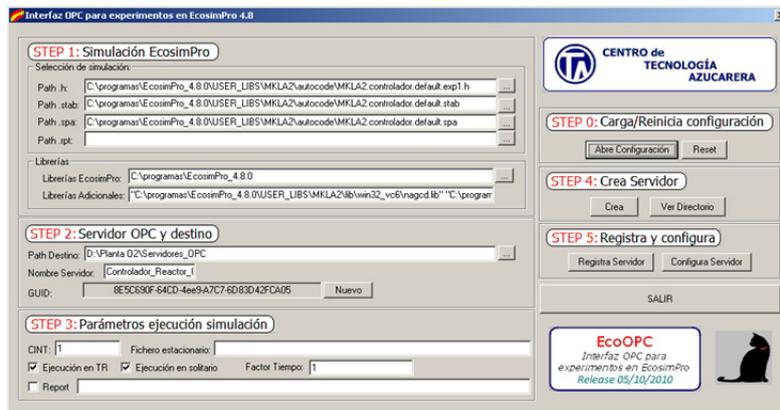


Figura 8.9. Creación del servidor OPC del controlador

En este caso, en los parámetros de ejecución de la simulación, debido a que se requiere llevar a cabo la simulación del controlador aplicado al modelo en tiempo real, se especifica un factor de tiempo y un intervalo de comunicación igual a 1.

En la configuración se añaden dos grupos:

- “Controlador”, donde se añaden las variables manipuladas y el set point.
- “Variables de proceso”, donde se añaden las variables de intercambio, los parámetros de sintonía y la variable *activate* para la sincronización de la lectura de las variables del proceso y la ejecución de las acciones de control calculadas por el optimizador.

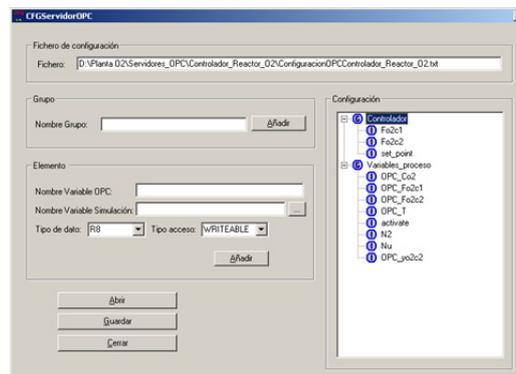
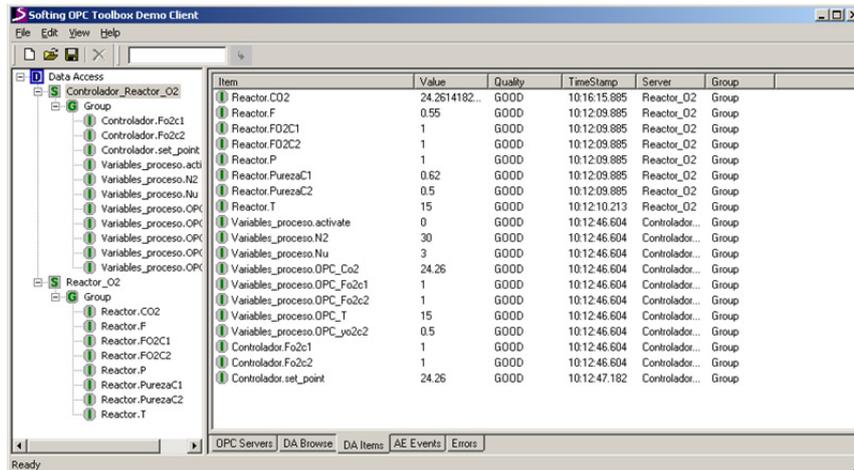


Figura 8.10. Configuración del servidor OPC del controlador

Posteriormente, se crea y se registra el servidor.

En Softing OPC se puede seguir la evolución de los valores de las variables. No obstante, no se puede ejecutar el problema de control desde esta pantalla, ya que es necesario que la variable *activate* pase por los valores 1 y 3 cada periodo de muestreo para que el controlador lea las variables del proceso (del modelo, en este caso) y ejecute las acciones de control, respectivamente, según se ha programado el código. En la configuración del SCADA, por ello, se genera un grafcet para la evolución cíclica de dicha variable, haciendo posible el problema de control en tiempo real.

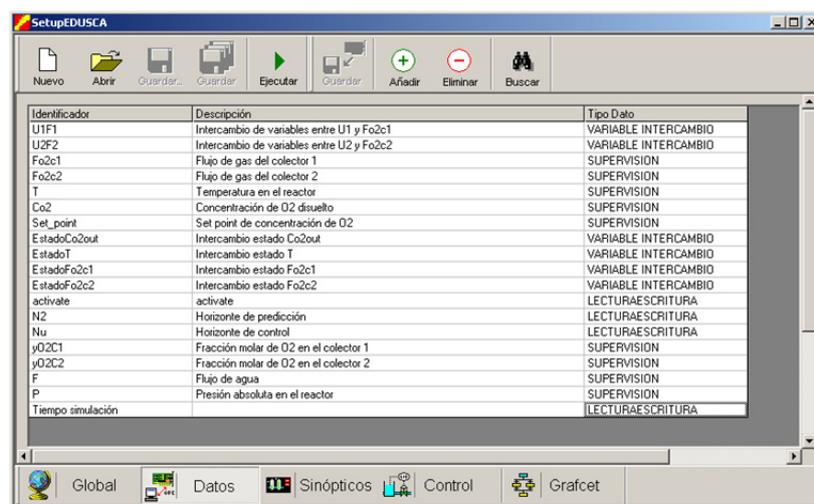


Item	Value	Quality	TimeStamp	Server	Group
Reactor.CO2	24.2614182...	6000	10:16:15.885	Reactor_02	Group
Reactor.F	0.55	6000	10:12:09.885	Reactor_02	Group
Reactor.FO2C1	1	6000	10:12:09.885	Reactor_02	Group
Reactor.FO2C2	1	6000	10:12:09.885	Reactor_02	Group
Reactor.P	1	6000	10:12:09.885	Reactor_02	Group
Reactor.PurezaC1	0.62	6000	10:12:09.885	Reactor_02	Group
Reactor.PurezaC2	0.5	6000	10:12:09.885	Reactor_02	Group
Reactor.T	15	6000	10:12:10.213	Reactor_02	Group
Variables_proceso.activate	0	6000	10:12:46.604	Controlador...	Group
Variables_proceso.N2	30	6000	10:12:46.604	Controlador...	Group
Variables_proceso.Nu	3	6000	10:12:46.604	Controlador...	Group
Variables_proceso.OPC_Co2	24.26	6000	10:12:46.604	Controlador...	Group
Variables_proceso.OPC_Fo2c1	1	6000	10:12:46.604	Controlador...	Group
Variables_proceso.OPC_Fo2c2	1	6000	10:12:46.604	Controlador...	Group
Variables_proceso.OPC_T	15	6000	10:12:46.604	Controlador...	Group
Variables_proceso.OPC_yo2c2	0.5	6000	10:12:46.604	Controlador...	Group
Controlador.Fo2c1	1	6000	10:12:46.604	Controlador...	Group
Controlador.Fo2c2	1	6000	10:12:46.604	Controlador...	Group
Controlador.set_point	24.26	6000	10:12:47.182	Controlador...	Group

Figura 8.11. Simulación del modelo y el controlador en Softing OPC

8.4. CONFIGURACIÓN DEL CONTROLADOR DE LA PLANTA EN EL SCADA

Se añaden las variables correspondientes a la simulación del proceso y al controlador, incluyendo el tiempo de simulación, aplicando la conversión de minutos a segundos para que tanto el modelo como el controlador se ejecuten en la misma unidad de tiempo.



Identificador	Descripción	Tipo Dato
U1F1	Intercambio de variables entre U1 y Fo2c1	VARIABLE INTERCAMBIO
U2F2	Intercambio de variables entre U2 y Fo2c2	VARIABLE INTERCAMBIO
Fo2c1	Flujo de gas del colector 1	SUPERVISION
Fo2c2	Flujo de gas del colector 2	SUPERVISION
T	Temperatura en el reactor	SUPERVISION
Co2	Concentración de O2 disuelto	SUPERVISION
Set_point	Set point de concentración de O2	SUPERVISION
EstadoCo2out	Intercambio estado Co2out	VARIABLE INTERCAMBIO
EstadoT	Intercambio estado T	VARIABLE INTERCAMBIO
EstadoFo2c1	Intercambio estado Fo2c1	VARIABLE INTERCAMBIO
EstadoFo2c2	Intercambio estado Fo2c2	VARIABLE INTERCAMBIO
activate	activate	LECTURA/ESCRITURA
N2	Horizonte de predicción	LECTURA/ESCRITURA
Nu	Horizonte de control	LECTURA/ESCRITURA
yO2C1	Fracción molar de O2 en el colector 1	SUPERVISION
yO2C2	Fracción molar de O2 en el colector 2	SUPERVISION
F	Flujo de agua	SUPERVISION
p	Presión absoluta en el reactor	SUPERVISION
Tiempo simulación		LECTURA/ESCRITURA

Figura 8.12. Configuración del controlador en el SCADA para su aplicación a la planta. Datos

Las variables de intercambio permiten la comunicación entre el controlador y el proceso (en este caso el modelo). En otras palabras, las variables de intercambio (de la concentración de oxígeno, de la temperatura y de los flujos de gas) leen las variables del modelo y envían sus valores al controlador. Por otro lado, en el caso de las variables manipuladas, además es necesario llevar a cabo la comunicación inversa, siendo el controlador el que envía al modelo los nuevos valores de los flujos de gas de los colectores, propuestos por el problema de control predictivo.

Así, por ejemplo, se define la variable U1F1, que recoge el valor que fija el controlador para la variable manipulada (flujo en el colector 1) y se lo pasa a la simulación del proceso (modelo):

Figura 8.13. Variable de intercambio U1F1

Igualmente, en el caso de la variable controlada (concentración de oxígeno), se lee el valor de la simulación del proceso (Variable Entrada) y se pasa al controlador (Variable Salida):

Figura 8.14. Variable de intercambio de la concentración de O2

Una vez introducidos los datos, se configura el sinóptico:

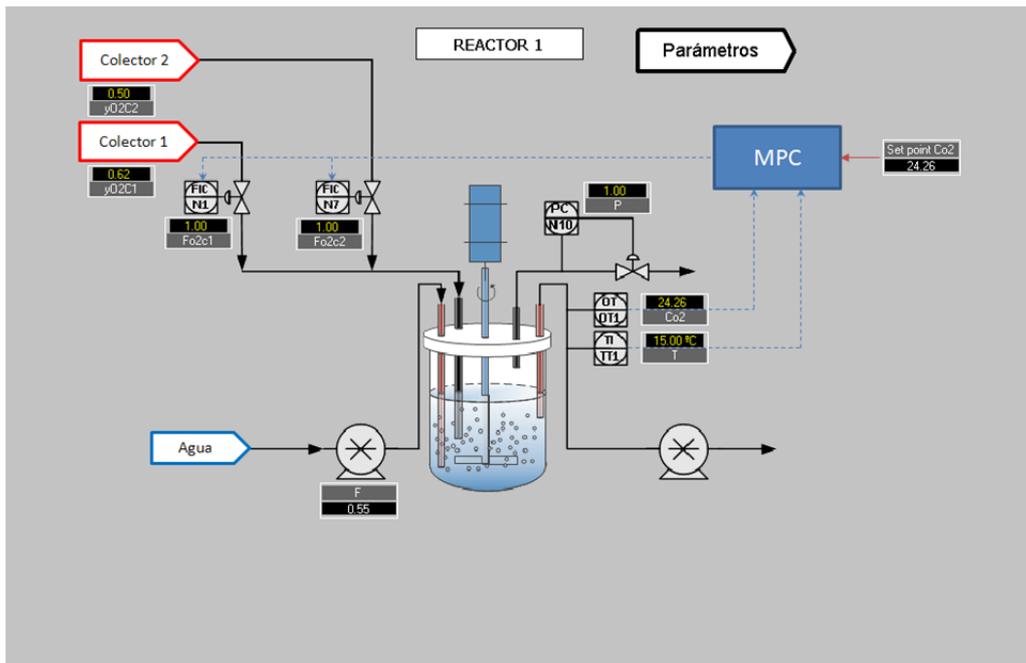


Figura 8.15. Sinóptico 1 de la simulación del controlador

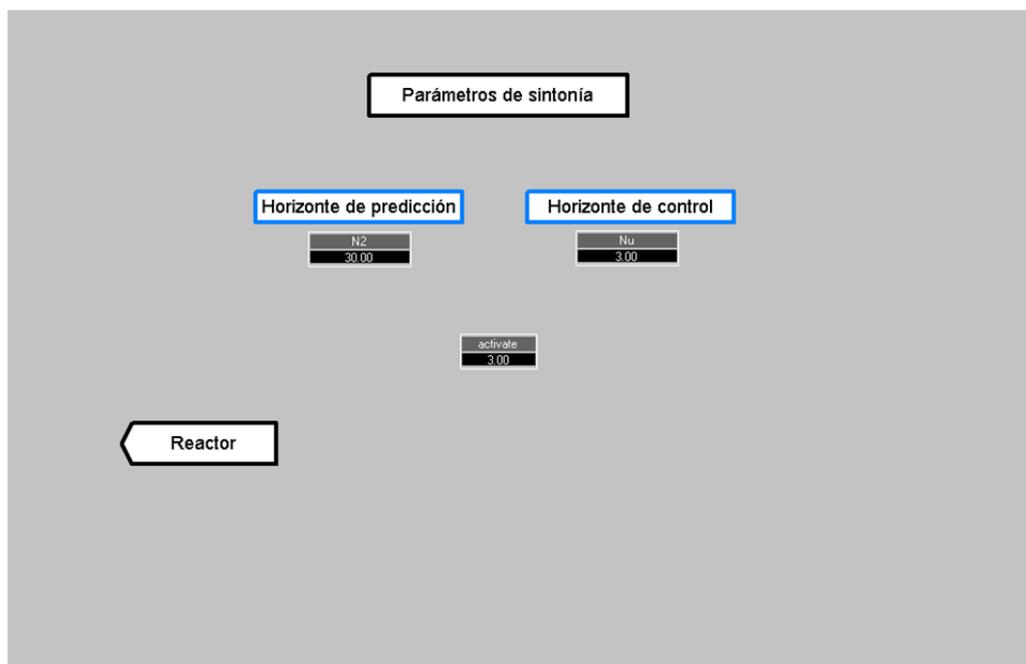


Figura 8.16. Sinóptico 2 de la simulación del controlador

Finalmente, en el bloque “Grafcet” se configura la transición entre etapas para llevar a cabo la sincronización de la lectura de las variables del modelo y el envío de las acciones de control, ya que en el código *controlador.el*, del que se ha creado el servidor OPC, cuyas variables se ponen en juego en el SCADA, no está programada la ejecución cíclica de la lectura de las variables del proceso por parte del controlador y el envío de las nuevas señales de control cada periodo de muestreo. Es necesario, por tanto, realizar la sincronización en el SCADA a través del siguiente grafcet:

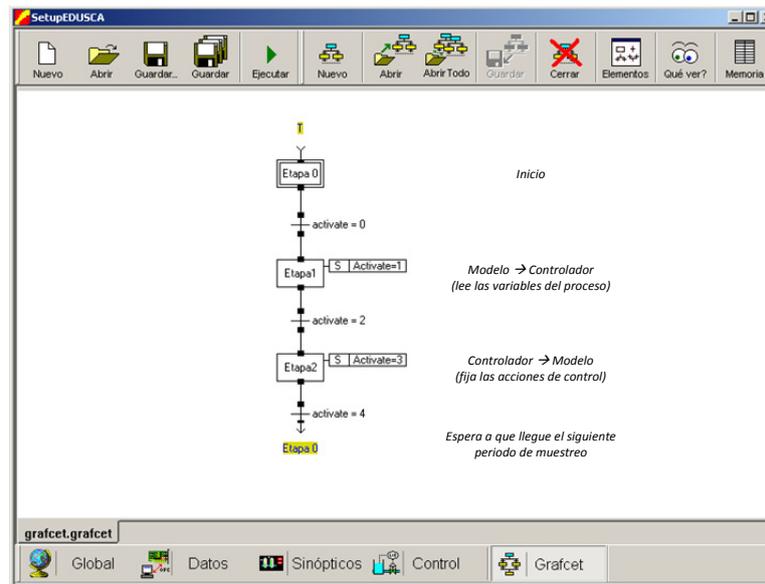


Figura 8.17. Esquema Grafcet para transición entre etapas

Inicialmente, se comienza con la etapa de espera, hasta que se cumple el periodo de muestreo, instante en que la variable *activate* toma el valor 1 (primera etapa del grafcet). En este momento, el controlador lee las variables actuales del modelo, a través de las variables de intercambio vía OPC, y las envía al código *coste*, donde está definida la función objetivo. Al final de esta etapa, *activate* pasa a valor 2, que es la condición de transición a la siguiente etapa del grafcet.

En la etapa 2, *activate* toma el valor 3, momento en que se lleva a cabo la optimización de la función objetivo, asignando las nuevas acciones de control, propuestas por el optimizador, a las variables manipuladas a través de las variables de intercambio correspondientes vía OPC. Posteriormente, *activate* pasa a 4 y se espera a que se cumpla un nuevo periodo de muestreo para volver al inicio del grafcet:

En resumen, las etapas son:

- Etapa 0: Espera.
- Etapa 1: El controlador lee las variables actuales vía OPC.
- Etapa 2: El controlador ejecuta las acciones de control.

De esta forma, la simulación del proceso en el SCADA EDUSCA debe reproducir exactamente la simulación en Ecosim. Para comprobarlo, se compara un experimento con los mismos saltos en la referencia en ambos soportes, obteniéndose:

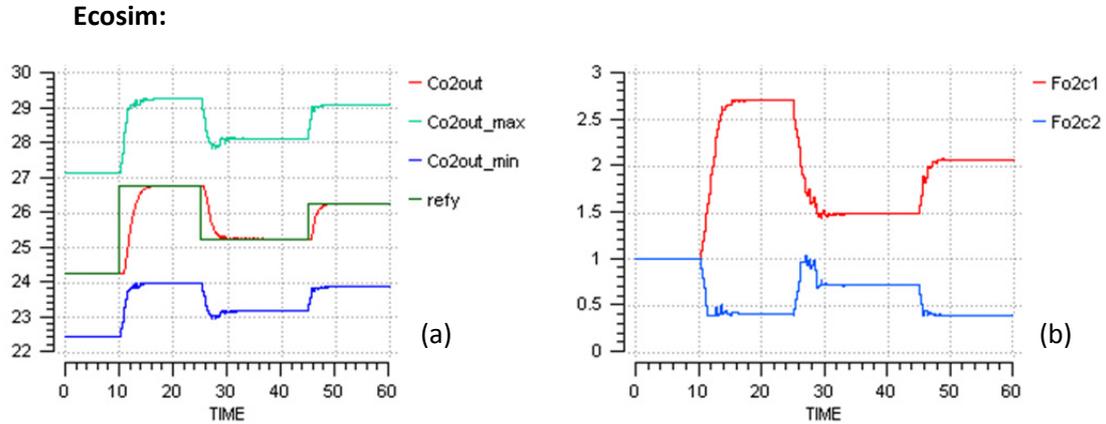


Figura 8.18. Simulación del controlador en Ecosim

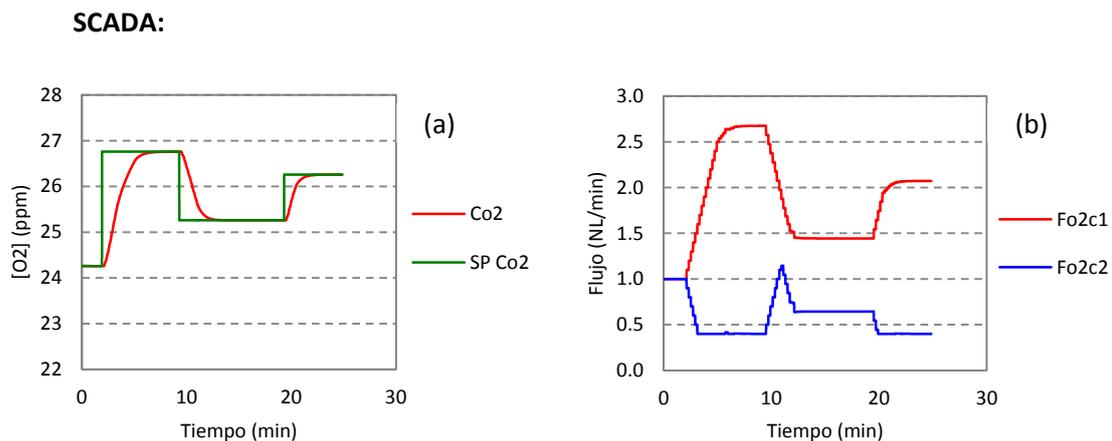


Figura 8.19. Simulación del controlador en el SCADA

Se observa el mismo tiempo de respuesta y la misma evolución en las variables manipuladas. De esta forma, se comprueba que la simulación del proceso en Ecosim se puede trasladar al SCADA gracias a la creación y configuración de los servidores OPC del modelo y del controlador y a la sincronización para el intercambio de datos en tiempo real entre modelo y controlador.

8.5. ESQUEMA DE COMUNICACIONES

Resumiendo, hasta este punto, se ha creado una configuración del SCADA [6] que permite simular el proceso del mismo modo que en Ecosim. Para ello se ha utilizado el programa CreaOPC 4.8, mediante el cual se han creado los servidores OPC de la simulación del proceso y del controlador y se ha configurado el SCADA EDUSCA para el intercambio de variables, tal como muestra el siguiente esquema:

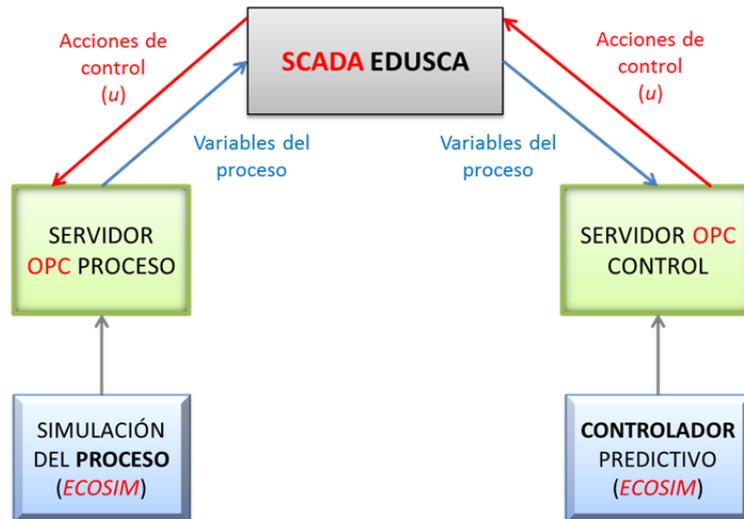


Figura 8.20. Esquema de comunicaciones

Como se observa en la figura se dispone de dos servidores OPC (Reactor_O2 y Controlador_Reactor_O2) en los que se han encapsulado los códigos programados en Ecosim del modelo, que simula el comportamiento del proceso, y del controlador, respectivamente.

Se ha configurado el SCADA EDUSCA para monitorizar la simulación del controlador aplicado al modelo, sincronizando la lectura de los valores de las variables del modelo por parte del controlador y el envío de las variables manipuladas del controlador al modelo a partir de las variables de intercambio, explicadas en el apartado 8.4, y la definición del grafset que permite coordinar cada periodo de muestreo el envío de señales del controlador al modelo y viceversa.

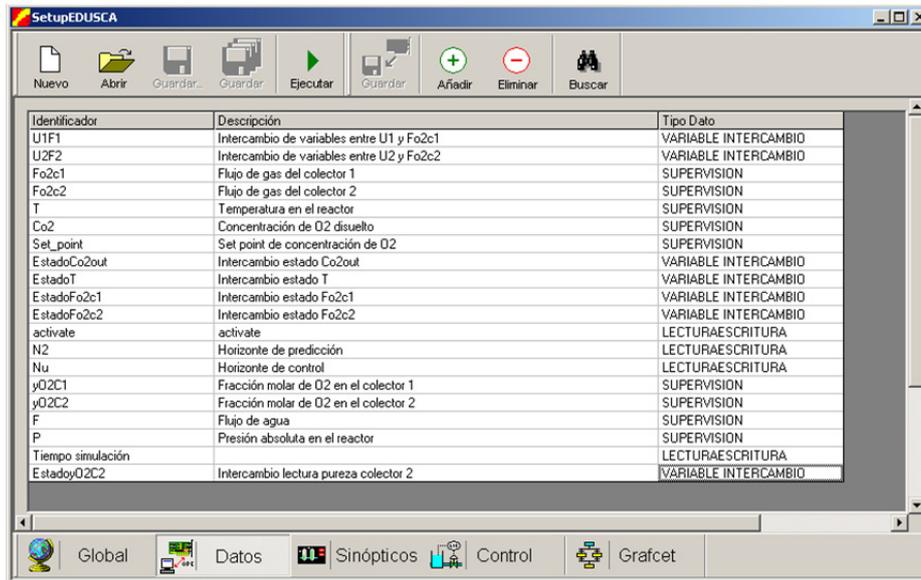
Por ello este esquema es clave para entender la comunicación controlador/modelo, ya que una vez creados y configurados los servidores, es necesario configurar el SCADA como soporte que coordina la interacción vía OPC.

El siguiente paso consiste en sustituir el servidor del modelo por el servidor de la planta, pasando así a controlar el proceso en tiempo real. En resumen, el proceso pasará a ser el servidor que se comunique con el controlador en lugar del modelo, comprobando así la idoneidad del controlador predictivo programado implementado en el proceso.

9. APLICACIÓN DEL CONTROLADOR PREDICTIVO A LA PLANTA

9.1. CONFIGURACIÓN DEL SCADA

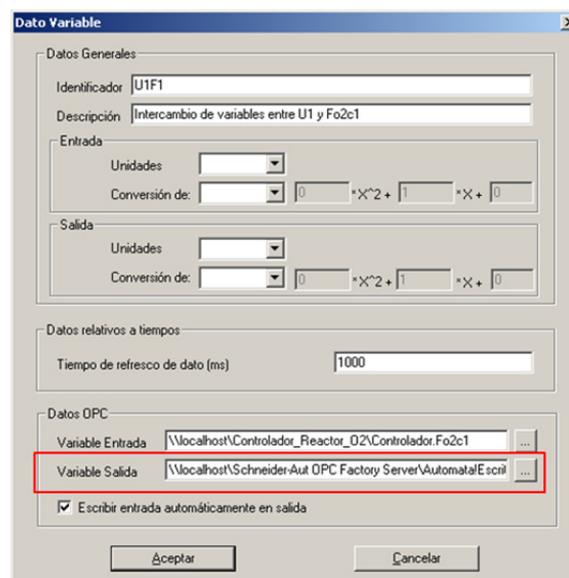
En este caso, se prescinde del servidor del modelo, sustituyéndolo por los valores recibidos de la planta.



Identificador	Descripción	Tipo Dato
U1F1	Intercambio de variables entre U1 y Fo2c1	VARIABLE INTERCAMBIO
U2F2	Intercambio de variables entre U2 y Fo2c2	VARIABLE INTERCAMBIO
Fo2c1	Flujo de gas del colector 1	SUPERVISION
Fo2c2	Flujo de gas del colector 2	SUPERVISION
T	Temperatura en el reactor	SUPERVISION
Co2	Concentración de O2 disuelto	SUPERVISION
Set_point	Set point de concentración de O2	SUPERVISION
EstadoCo2out	Intercambio estado Co2out	VARIABLE INTERCAMBIO
EstadoT	Intercambio estado T	VARIABLE INTERCAMBIO
EstadoFo2c1	Intercambio estado Fo2c1	VARIABLE INTERCAMBIO
EstadoFo2c2	Intercambio estado Fo2c2	VARIABLE INTERCAMBIO
activate	activate	LECTURAESCRITURA
N2	Horizonte de predicción	LECTURAESCRITURA
Nu	Horizonte de control	LECTURAESCRITURA
yO2C1	Fracción molar de O2 en el colector 1	SUPERVISION
yO2C2	Fracción molar de O2 en el colector 2	SUPERVISION
F	Flujo de agua	SUPERVISION
P	Presión absoluta en el reactor	SUPERVISION
Tiempo simulación		LECTURAESCRITURA
EstadoyO2C2	Intercambio lectura pureza colector 2	VARIABLE INTERCAMBIO

Figura 9.1. Configuración del controlador aplicado a la planta real en el SCADA. Datos

Ahora, en las variables de intercambio de los flujos de los colectores, en lugar de tener como variable de salida la correspondiente al modelo, se apunta al valor medido de la planta (Variable Salida):



Dato Variable

Datos Generales

Identificador: U1F1

Descripción: Intercambio de variables entre U1 y Fo2c1

Entrada

Unidades: []

Conversión de: [] 0 *X² + [] *X + [] 0

Salida

Unidades: []

Conversión de: [] 0 *X² + [] *X + [] 0

Datos relativos a tiempos

Tiempo de refresco de dato (ms): 1000

Datos OPC

Variable Entrada: \\localhost\Controlador_Reactor_O2\Controlador.Fo2c1

Variable Salida: \\localhost\Schneider-Aut OPC Factory Server\Automata\Escri

Escribir entrada automáticamente en salida

Aceptar Cancelar

Figura 9.2. Variable de intercambio U1F1

Del mismo modo, en el caso de la variable controlada, se toma la medida de la sonda de oxígeno de la planta real (Variable Entrada), no de la simulación del proceso:

Figura 9.3. Variable de intercambio para la concentración de O_2

9.2. ESQUEMA DE COMUNICACIONES

El esquema de comunicaciones cuando se aplica el control a la planta real en lugar de realizarlo sobre la simulación del proceso (modelo), es el siguiente:

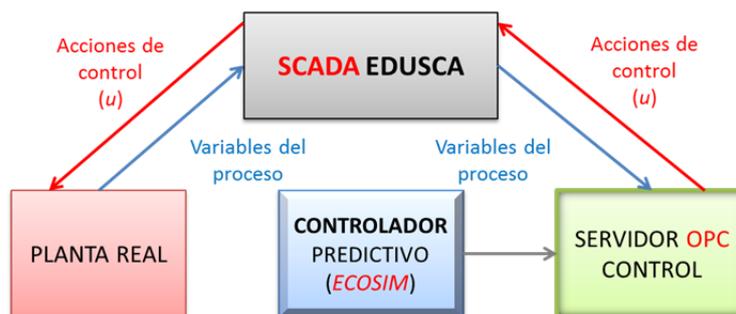


Figura 9.4. Esquema de comunicaciones del controlador aplicado a la planta real

Se comprueba que independientemente de que se lean los valores del servidor OPC creado a partir de la simulación del proceso en Ecosim, o bien del servidor de la planta real, la comunicación se realiza de forma correcta, comprobando que la sincronización del envío y lectura de datos del controlador al proceso y viceversa es idéntica.

9.3. EXPERIMENTOS DE CONTROL APLICADO A LA PLANTA

Se realizan varios experimentos de aplicación del controlador predictivo a la planta, obteniendo los resultados que se muestran a continuación. La variable controlada en todos ellos es la concentración de oxígeno disuelto en el reactor, manipulando los flujos en ambos colectores. La perturbación medible es la temperatura, cuya influencia es muy significativa ya que influye tanto en la solubilidad de oxígeno en agua (ley de Henry) como en el coeficiente de transferencia de materia.

Igualmente, se observó que la pureza de oxígeno en los colectores variaba con el flujo demandado, siendo dicha perturbación mucho más significativa en el segundo colector (entre 45 y 52%), mientras que en el primero fluctúa entre el 62 y el 63% de oxígeno. Dado que solo se dispone de un analizador de O_2 , se toma como perturbación medible la pureza del segundo colector (de media pureza).

En los siguientes experimentos de aplicación del controlador a la planta se comparan las distintas formas de corrección del error explicadas en el apartado 7, comprobando que la con la estrategia de estimación de horizonte móvil se consiguen eliminar los errores estacionarios.

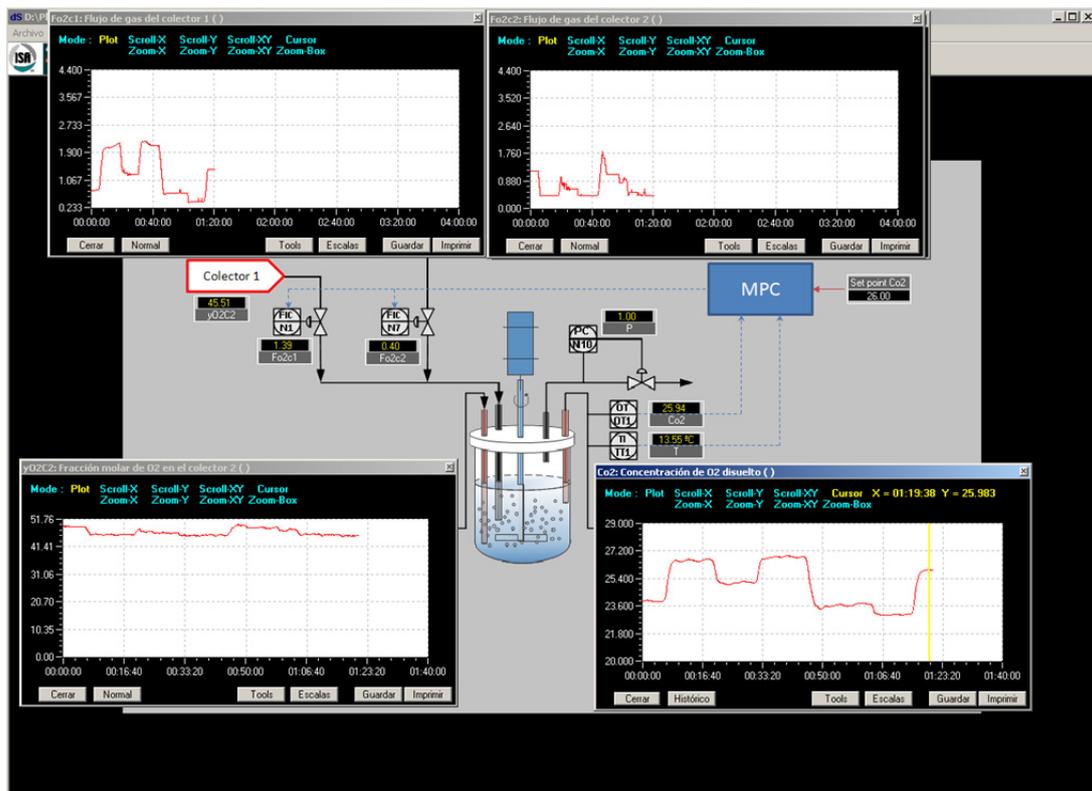


Figura 9.5. Experimento de aplicación del controlador predictivo a la primera unidad consumidora

9.3.1. Experimento 1: Corrección del error en el punto inicial de integración

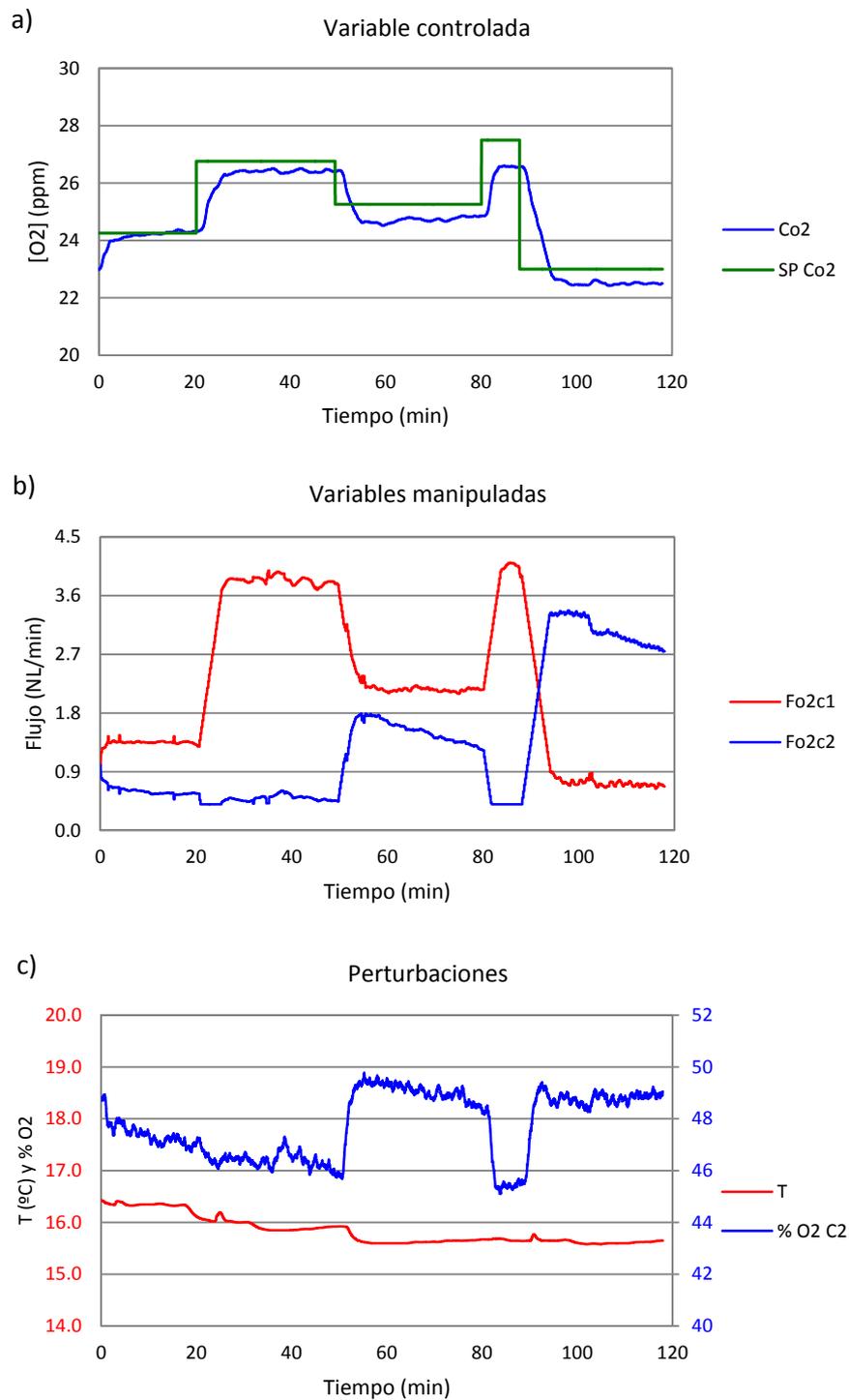


Figura 9.6. Experimento 1. Corrección del error en el punto inicial de integración. (a) Evolución de la variable controlada ante saltos en la referencia, (b) Evolución de las variables manipuladas, (c) Evolución de las perturbaciones medidas del proceso (temperatura y pureza del segundo colector)

Errores estacionarios:

SP (ppm)	[O ₂] (ppm)	Error (%)
24.26	24.31	0.21
26.76	26.41	1.31
25.26	24.85	1.62
27.50	26.55	3.45
23.00	22.47	2.30
		1.36

Tabla 9.1. Experimento 1. Errores estacionarios obtenidos llevando a cabo la corrección del error en el punto inicial de integración de la función de coste

Se obtiene un error estacionario medio aproximado de 0.30 ppm, correspondiente a un porcentaje de 1.36%. Cabe señalar que en este experimento se fijó una referencia de 27.50 ppm en el minuto 80, la cual no es alcanzable a una temperatura cercana a 15.5 °C. Por ello se observa un error tan amplio en esa referencia, que fue cambiada por una referencia distinta a los 8 minutos, a diferencia de los demás saltos que se mantuvieron alrededor de 30 minutos.

En cuanto a la dinámica del sistema, se observa una respuesta similar a la obtenida en simulación, con un tiempo comprendido entre 5 y 10 minutos hasta llegar a los estados estacionarios.

Se puede ver cierta oscilación en la variable controlada (± 0.2 ppm) en los estados estacionarios, lo cual se puede deber a perturbaciones no medidas como el volumen de líquido en el reactor por el efecto de las burbujas de gas.

9.3.2. Experimento 2: Corrección del error tras la integración de la función de coste

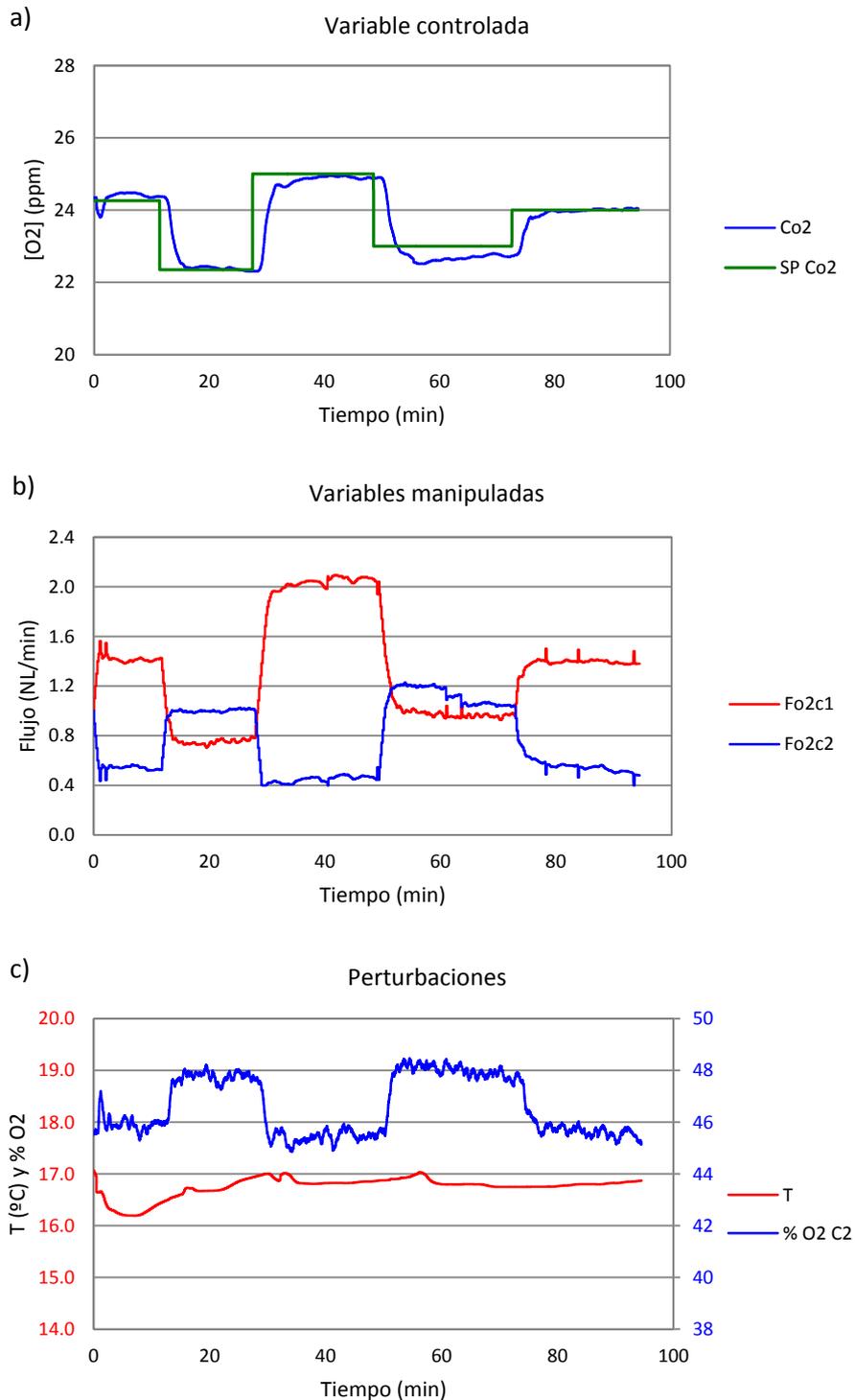


Figura 9.7. Experimento 2. Corrección del error tras la integración de la función de coste. (a) Evolución de la variable controlada ante saltos en la referencia, (b) Evolución de las variables manipuladas, (c) Evolución de las perturbaciones medidas del proceso (temperatura y pureza del segundo colector)

Errores estacionarios:

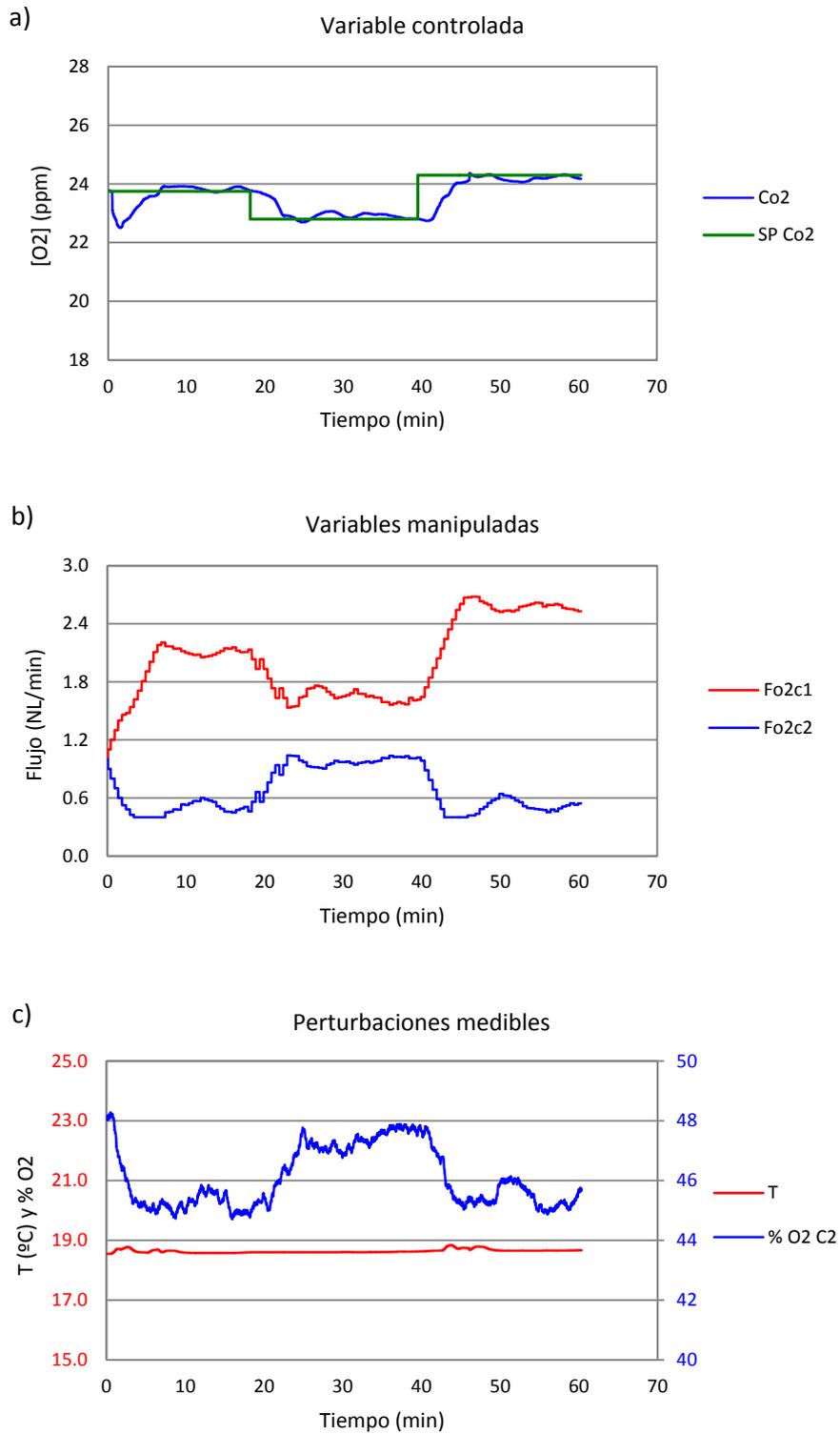
SP (ppm)	[O₂] (ppm)	Error (%)
24.26	24.38	0.49
22.35	22.36	0.04
25.00	24.91	0.36
23.00	22.73	1.17
24.00	24.02	0.08
		0.43

Tabla 9.2. Experimento 2. Errores estacionarios obtenidos llevando a cabo la corrección del error tras la integración de la función de coste

Se obtiene un error medio entre la referencia y el valor de concentración alcanzado en los estados estacionarios de 0.43%, significativamente menor comparado con el anterior experimento. Se observa, por tanto, una mejor corrección del error cuando se suma al valor de concentración obtenido tras la integración de la función de coste en cada periodo de muestreo, como se observó en simulación. De hecho, se reduce el error más de un 50% con respecto a la corrección en el punto inicial de integración.

Por otro lado, la dinámica del sistema es similar, con tiempos de asentamiento cercanos a los 10 minutos.

9.3.3. Experimento 3: Moving Horizon Estimation



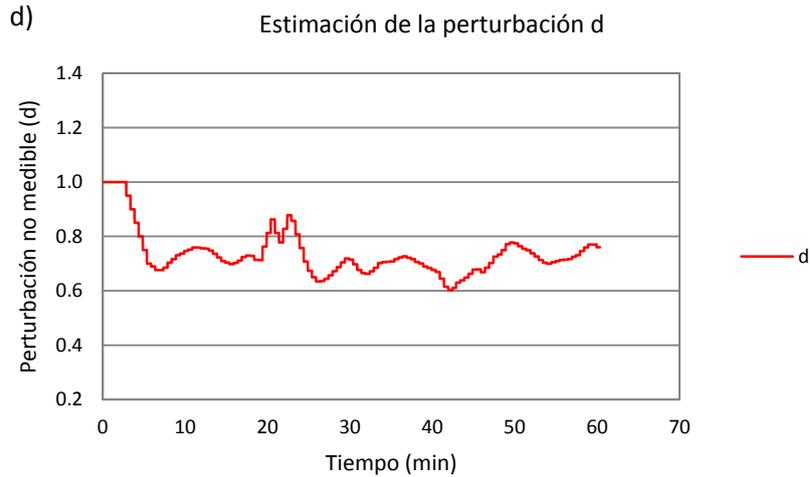


Figura 9.8. Experimento 3. Aplicación del Moving Horizon Estimator. (a) Evolución de la variable controlada ante saltos en la referencia, (b) Evolución de las variables manipuladas, (c) Evolución de las perturbaciones medidas del proceso (temperatura y pureza del segundo colector), (d) Estimación del valor de la perturbación no medible (d)

Al aplicar el problema de estimación del parámetro de perturbación no medible, d , que es multiplicado por la ecuación algebraica ajustada del coeficiente de transferencia de materia, se consigue llegar a la referencia, ya que el modelo es ajustado a las perturbaciones no medibles del proceso y se corrigen los errores de modelado.

Inicialmente, se observa que el parámetro d comienza con el valor 1, variando a medida que avanza el tiempo hasta que la variable controlada alcanza la referencia, momento en que la estimación del parámetro d varía en un intervalo más estrecho.

Como puede observarse, aunque la referencia es alcanzada, existe cierta oscilación de ± 0.25 ppm en torno al set point, que al realizar la estimación del parámetro d , se va atenuando.

10. CONCLUSIONES Y TRABAJO FUTURO

En el presente trabajo se ha aplicado la tecnología de control predictivo a una de las unidades consumidoras de la planta de distribución de oxígeno instalada en el Laboratorio del Departamento de Ingeniería de Sistemas y Automática (EII, Sede Dr. Mergelina, Universidad de Valladolid), mostrando los pasos a seguir para su implementación.

Para ello, inicialmente se ha identificado y validado un modelo de conocimiento, obteniendo una expresión ajustada del coeficiente de materia en función de las variables medidas en la planta.

Posteriormente, se ha empleado dicho modelo para la programación y simulación de un controlador predictivo no lineal a fin de controlar la concentración de oxígeno en la unidad consumidora, variando las referencias de los controladores PID de flujo de gas, instalados en las ramas de aporte de cada colector.

El mejor comportamiento del controlador se consigue aplicando la estrategia del estimador de horizonte móvil, consiguiendo eliminar los errores estacionarios causados por la diferencia entre los valores predichos por el modelo y los valores reales, ya que el modelo no reproduce de forma exacta el comportamiento del proceso, debido a los errores de modelado.

La aplicación del control predictivo programado en EcosimPro® a la planta real se lleva a cabo vía OPC mediante la creación del servidor OPC del controlador y configurando el SCADA para el intercambio de las variables del proceso con la planta, constituyendo así una herramienta fácil de manejar a la hora de implementar el problema de control al proceso en tiempo real.

En cuanto a las líneas de investigación futuras, cabe destacar el desarrollo de técnicas de control distribuido, aplicando el control a las tres unidades de manera simultánea. En este caso la función de coste de los controladores debe incluir los términos de coordinación (suma del producto del flujo demandado de cada colector por el precio asociado a ese recurso). De esta forma, el coordinador, en base a la demanda de flujo de cada colector, fija los precios asociados a ambos, incrementándose cuando la demanda excede el límite superior de aporte.

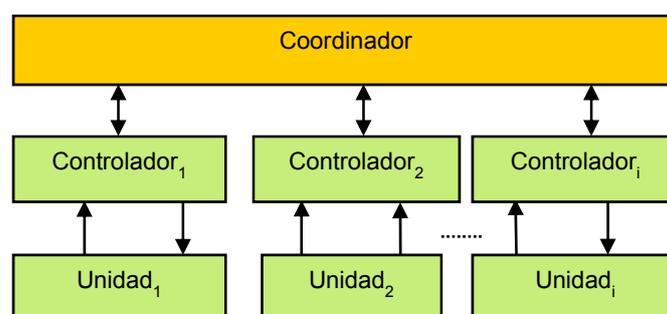


Figura 10.1. Esquema de control distribuido

11. ANEXO I - CÓDIGOS DE ECOSIMPRO

Código del archivo coste.el:

```

COMPONENT coste      (INTEGER nent = 2, INTEGER nsal = 1,
                    INTEGER maxnu = 3, INTEGER maxn2 = 30)

-- Variable medida y controlada (nsal = 1):
-- 1. Co2out      -- Concentración de O2

-- Variables manipuladas (nent = 2):
-- 1. Fo2c1      -- Flujo de gas en colector 1
-- 2. Fo2c2      -- Flujo de gas en colector 2

-- Perturbación medible
-- 1. T          -- Temperatura

DATA
REAL F = 0.55      -- Flujo de agua (L/min)
REAL V = 0.84      -- Volumen de líquido (L)
REAL yo2c1 = 0.62  -- Fracción molar de O2 colector 1
REAL yo2c2 = 0.50  -- Fracción molar de O2 colector 2
REAL P_tot = 1     -- Presión total (atm)
-- Parámetros del modelo (a, m, n) (kla = m*((T-n)^0.5)*Fo2^a)
-- Rango de validez: T de 11 a 17°C y Fo2 de 0.4 a 4.4 L/min.
REAL a = 0.6822, m = 0.7432, n = 7.667

-- Valores iniciales (estado estacionario)
REAL Fo2c1i = 1
REAL Fo2c2i = 1
REAL Co2outi = 24.26
REAL Ti = 15

-- Variables del controlador predictivo que tomarán los valores
-- recibidos desde el código del controlador
-- Periodo de muestreo
REAL tsamp = 0.2 "min"
-- Horizonte de predicción
INTEGER N2 = 10
-- Horizonte de control
INTEGER NU[2] = {3, 3} -- NU[nent]
-- Control futuro
REAL uf[2, 3]={{0.5,0.6,0.7}, {0.7,0.6,0.4}}
-- Límites en las variables controladas
REAL Lsup[1] = {40} -- Lsup[nsal]
REAL Linf[1] = {20} -- Linf[nsal]
-- Parámetro para la trayectoria de referencia
REAL alfa[1] = {0.3} -- alfa[nsal]
-- Valor del índice que activa las funciones de penalización en el coste
INTEGER ind_penal = 1 -- 1 activa; 0 desactiva

DECLS
DISCR REAL y_predic[nsal,maxn2]
REAL Co2in -- Concentración de O2 a la entrada (ppm)
REAL Co2sat -- Concentración de O2 en saturación (ppm)

```

```

REAL Co2out    -- Concentración de O2 a la salida (ppm)
REAL Fo2c1    -- Flujo de gas colector 1 (L/min)
REAL Fo2c2    -- Flujo de gas colector 2 (L/min)
REAL Fo2      -- Flujo total de gas (L/min)
REAL yo2      -- Pureza de O2 (tanto por 1)
DISCR REAL T  -- Temperatura (°C)
REAL H        -- Constante de Henry (atm L/mg)
REAL kla      -- Coeficiente de transferencia de materia (1/min)

```

```
-- Variables manipuladas
```

```
DISCR REAL u[nent]
```

```
-- Valor anterior de las variables manipuladas
```

```
DISCR REAL uant[nent]
```

```
-- Variables controladas
```

```
REAL y[nsal]
```

```
-- Trayectoria de referencia interna
```

```
REAL ref_int[nsal]
```

```
DISCR REAL ref_fut[nsal,maxn2]
```

```
-- El índice que se minimiza
```

```
DISCR REAL J
```

```
-- Pesos para las funciones de penalización
```

```
REAL R1inf, R1sup
```

```
-- Vector de las restricciones no lineales
```

```
REAL val_func[2]
```

```
-- Variable auxiliar para el cálculo de máximos y mínimos
```

```
REAL ymin[1], ymax[1]
```

```
-- Flujo max
```

```
REAL condicionFo2max
```

```
-- Consigna
```

```
REAL refy
```

```
-- Error (BIAS)
```

```
DISCR REAL Error
```

```
-- Variables bucles
```

```
INTEGER i = 1
```

```
INTEGER j = 1
```

```
INTEGER k = 1
```

```
INTEGER ind = 0
```

```
BOOLEAN Sample = TRUE
```

```
INIT
```

```
-- Variable controlada
```

```
Co2out = Co2outi
```

```
-- Referencia interna
```

```
ref_int[1] = Co2outi
```

```
-- Variables manipuladas
```

```
Fo2c1 = Fo2c1i
```

```
Fo2c2 = Fo2c2i
```

```
u[1] = Fo2c1
```

```
u[2] = Fo2c2
```

```
T = Ti
```

```
-- Índice de optimizar
```

```
J = 0.0
```

```
-- Concentración máxima y mínima
```

```
ymin[1] = Co2outi
```

```
ymax[1] = Co2outi
```

DISCRETE

```

WHEN (Sample) THEN
    ind = ind + 1
    k = ind
-- Asignar valores futuros a las manipuladas:
    FOR (i IN 1,nent)
        j = min (ind, NU[i])
        uant[i] = u[i]
        u[i] = uf[i,j]
    END FOR

    FOR (i IN 1,nsal)
        y_predic[i,k] = y[i]
    END FOR
Sample = FALSE
Sample = TRUE AFTER tsamp
END WHEN

```

CONTINUOUS

-- ECUACIONES ANALÍTICAS

```

-- Constante de Henry experimental
H = 0.0005*T + 0.01315
-- Balance global
Fo2 = Fo2c1 + Fo2c2
-- Balance de materia al O2
Fo2*yo2 = Fo2c1*yo2c1+Fo2c2*yo2c2
-- Concentración de saturación a partir de la constante de Henry experimental
Co2sat = P_tot*yo2/H
-- Concentración de entrada a partir de la constante de Henry experimental
Co2in = 1*0.21/H
-- kla modelo: (kla = m*((T-n)^0.5)*Fo2^a)
kla = m*((T-n)**0.5)*Fo2**a
-- Variables manipuladas
-- Flujo de O2
Fo2c1 = u[1]
Fo2c2 = u[2]

```

-- ECUACIONES DIFERENCIALES

```

-- Balance de materia en fase líquida
V*Co2out' = kla*V*(Co2sat-Co2out) + F*(Co2in-Co2out)

-- Trayectoria de referencia
ref_int[1]' = (refy - ref_int[1]) / alfa[1]

```

-- Error (BIAS)

```
y[1] = Co2out + Error
```

```

-- Índice de coste
J' = (y[1]-ref_int[1])**2 \
+ 0.1*(u[1]-uant[1])**2 \
+ 0.1*(u[2]-uant[2])**2 \
+ 500*condicionFo2max\
+ ind_penal*R1sup*(Lsup[1]-y[1])**2 \
+ ind_penal*R1inf*(Linf[1]-y[1])**2

```

```
R1sup = ZONE (y[1] > Lsup[1]) 10000.0
        OTHERS 0.0
R1inf = ZONE (y[1] < Linf[1]) 10000.0
        OTHERS 0.0
```

```
-- Calcular las restricciones no lineales (expresiones  $g(x) \leq 0$ )
-- Mantener la concentración dentro de los límites.
-- Si se trabaja con estos valores, entonces en la expresión de J la variable
-- ind_penal se considera 0.
val_func[1] = Linf[1] - minabs(y[1], ymin[1])
val_func[2] = maxabs(y[1], ymax[1]) - Lsup[1])

condicionFo2max= maxabs(u[1]+u[2]-4.4, 0)
```

```
END COMPONENT
```

Código del archivo control.el:

```
-----
-- EXTERNAL FUNCTION
-----
```

```
"C++" FUNCTION NO_TYPE ini_control (IN INTEGER nr_ent, IN INTEGER Nu[],
                                   IN INTEGER n_res,
                                   IN REAL L_est_inf[], IN REAL L_est_sup[],
                                   IN REAL D_est_inf[], IN REAL D_est_sup[],
                                   IN REAL f_prec) IN "optim_control.lib"
```

```
"C++" FUNCTION NO_TYPE optim_control (FUNC_PTR fcn, IN REAL u_est[]) IN
                                     "optim_control.lib" "nagcd.lib"
```

```
-----
-- CLASS INHERITED FROM coste_default TO GENERATE AN OPTIMIZATION
-----
```

```
CLASS optimiz IS_A coste_default
```

```
DECLS
```

```
-- Horizonte de predicción
INTEGER Hp = 30
-- Periodo de muestreo
REAL per_muestreo = 0.2 "min"
-- Número de variables manipuladas
INTEGER nr_inputs = 2
-- Horizonte de control para cada entrada
CONST INTEGER H_con[2] = {3, 3}
-- Utilizar el mismo horizonte para todas las entradas
CONST INTEGER H_con_comun = 3
-- Restricciones de camino que se incluirán en el problema de optimización
INTEGER nr_path_constr = 0      -- Un 0 indica que las restricciones
                                -- sobre los estados se expresan a través de
                                -- funciones de penalización

-- Límites en las variables manipuladas
REAL Usup[2] = {4.4, 4.4}
REAL Uinf[2] = {0.4, 0.4}
-- Límites en los incrementos
REAL Dsup[2] = {0.1, 0.1}
REAL Dinf[2] = {-0.1, -0.1}

REAL f_prec = 1.e-15 -- precisión en el cálculo de la función objetivo

--Valor de la predicción
REAL y_prediccion = 0
```

```
METHODS
```

```
METHOD NO_TYPE ini ()
```

```
  BODY
```

```
    ini_control      (nr_inputs, H_con, nr_path_constr,
                    Uinf, Usup, Dinf, Dsup, f_prec)
```

```
END METHOD
```

```
-----
-- CALLBACK METHOD TO BE PASSED TO OPTIMIZATION ROUTINE
-----
```

```

METHOD NO_TYPE objetivo (IN INTEGER n_est, IN REAL var_est[], IN REAL func_res[])
  DECLS

  BODY
    RESET_VARIABLES()
    RESTORE_STATE("estado_actual.txt")
    EXEC_INIT()
    -- Pasar el horizonte de control al modelo
    setValueReal("NU[1]", H_con[1])
    setValueReal("NU[2]", H_con[2])

    setArray2D("uf", var_est, nr_inputs, H_con_comun)

    IF (nr_path_constr == 0) THEN
      setValueInt("ind_penal",1)
    ELSE
      setValueInt("ind_penal",0)
    END IF

    TIME = 0.0
    CINT = (Hp-1)*per_muestreo
    TSTOP = Hp*per_muestreo
    REL_ERROR = 1.e-09    -- 0.01*f_prec
    INIT_INTEG_STEP = 0.000001
    INTEG()
    -- Recoger el valor de la función objetivo
    func_res[1] = getValueReal("J")
    -- Recoger los valores de las funciones g(x)
    -- que definen las restricciones de camino
    func_res[2] = getValueReal("val_func[1]")
    func_res[3] = getValueReal("val_func[2]")

    y_prediccion = getValueReal("y_predic[1,2]")

  END METHOD

-----
-- MAIN OPTIMIZATION METHOD
-----

METHOD NO_TYPE control (IN REAL con[])
  DECLS

  BODY
    -- Llamar al optimizador
    optim_control (objetivo, con)
  END METHOD

METHOD REAL prediccion()
  DECLS

  BODY
    RETURN y_prediccion
  END METHOD

END CLASS
-- Aquí acaba la definición de la clase optimiz

```

COMPONENT control

DATA

```

REAL F = 0.55          -- Flujo de agua (L/min)
REAL V = 0.84          -- Volumen de líquido (L)
REAL yo2c1 = 0.62     -- Fracción molar de O2 colector 1
REAL yo2c2 = 0.50     -- Fracción molar de O2 colector 2
REAL P_tot = 1         -- Presión total (atm)
-- Parámetros del modelo (a, m, n) ( $kla = m \cdot ((T-n)^{0.5}) \cdot Fo2^a$ )
-- Rango de validez: T de 11 a 17°C y Fo2 de 0.4 a 4.4 L/min.
REAL a = 0.6822, m = 0.7432, n = 7.667

```

```

-- Valores iniciales (estado estacionario)

```

```

REAL Fo2c1inic = 1
REAL Fo2c2inic = 1
REAL Co2outinic = 24.26
REAL Tinic = 15

```

```

-- Periodo de muestreo

```

```

REAL tsamp = 0.2 "min"

```

```

-- Límites de operación

```

```

REAL Fo2c1_max = 4.4
REAL Fo2c1_min = 0.4
REAL Fo2c2_max = 4.4
REAL Fo2c2_min = 0.4

```

DECLS

```

-- Valores máximos y mínimos en el proceso

```

```

REAL Co2out_max
REAL Co2out_min

```

```

REAL Co2in      -- Concentración de O2 a la entrada (ppm)
REAL Co2sat     -- Concentración de O2 en saturación (ppm)
REAL Co2out     -- Concentración de O2 a la salida (ppm)
REAL Fo2c1      -- Flujo de gas colector 1 (L/min)
REAL Fo2c2      -- Flujo de gas colector 2 (L/min)
REAL Fo2        -- Flujo de gas total (L/min)
REAL yo2        -- Pureza de O2 (tanto por 1)
REAL T          -- Temperatura (°C)
REAL H          -- Constante de Henry (atm L/mg)
REAL kla        -- Coeficiente de transferencia de materia (1/min)

```

```

REAL e          -- yref - Co2out

```

```

-- Valores iniciales (estado estacionario)

```

```

REAL Fo2c1i
REAL Fo2c2i
REAL Ti
REAL Co2outi

```

```

-- Variables del controlador

```

```

-- Referencia

```

```

REAL refy

```

```

-- Variables que devuelve el controlador

```

```

DISCR REAL u_new[2]

```

```

-- Activar/desactivar la llamada al controlador cada periodo de muestreo
BOOLEAN sample = TRUE
DISCR REAL Valor_Predicho = 0
DISCR REAL Error = 0

```

OBJECTS

```
optimiz modelo
```

INIT

```

Fo2c1 = Fo2c1inic
Fo2c2 = Fo2c2inic
T = Tinic
Co2out = Co2outinic
Valor_Predicho = Co2outinic

```

```

-- Inicializar el controlador
modelo.ini ()
u_new[1] = Fo2c1inic
u_new[2] = Fo2c2inic
refy = Co2outinic

```

DISCRETE

```

-- Llamada al controlador predictivo
WHEN (sample == TRUE) THEN
-- Guardar los estados, perturbaciones y referencias actuales
    Error = Co2out - Valor_Predicho
    SAVE_VARS("estado_actual.txt", "Fo2c1i Fo2c2i Ti Co2outi refy Error")
-- Recoger la nueva señal de control
    modelo.control (u_new)
    Valor_Predicho = modelo.prediccion()
    Fo2c1 = u_new[1]
    Fo2c2 = u_new[2]

    sample = FALSE
    sample = TRUE AFTER tsamp

```

```
END WHEN
```

CONTINUOUS

```

-- Establecimiento de los valores máximos y mínimos de Co2out:
-- máximo: saturación; mínimo: flujo mínimo en los colectores (0.8)
Co2out_max = Co2sat
Co2out_min = (F*Co2in + V*(m*((T-n)**0.5)*0.8**a*Co2sat))/ \
    (F+V*(m*((T-n)**0.5)*0.8**a))

-- ECUACIONES ANALÍTICAS

-- Constante de Henry experimental
H = 0.0005*T + 0.01315
-- Balance global
Fo2 = Fo2c1 + Fo2c2
-- Balance de materia al O2
Fo2*yo2 = (Fo2c1*yo2c1+Fo2c2*yo2c2)
-- Concentración de saturación a partir de la constante de Henry experimental
Co2sat = P_tot*yo2/H
-- Concentración de entrada a partir de la constante de Henry experimental
Co2in = P_tot*0.21/H
-- kla modelo: (kla = m*((T-n)^0.5)*Fo2^a)

```

$$kla = m*((T-n)**0.5)*Fo2**a$$

-- Referencia - valor de la variable controlada

$$e = refy - Co2out$$

-- ECUACIONES DIFERENCIALES

-- Balance de materia en fase líquida

$$V*Co2out' = kla*V*(Co2sat-Co2out) + F*(Co2in-Co2out)$$

-- Asignación de variables actuales para pasarlos al coste

$$Fo2c1i = Fo2c1 \quad \text{-- Flujo de gas colector 1 (L/min)}$$

$$Fo2c2i = Fo2c2 \quad \text{-- Flujo de gas colector 2 (L/min)}$$

$$Ti = T \quad \text{-- Temperatura (°C)}$$

$$Co2outi = Co2out \quad \text{-- Concentración de O2 a la salida (ppm)}$$

END COMPONENT

Código del archivo controlador.el:

```

-----
-- EXTERNAL FUNCTION
-----
"C++" FUNCTION NO_TYPE ini_control      (IN INTEGER nr_ent, IN INTEGER Nu[], IN INTEGER n_res,
                                         IN REAL L_est_inf[], IN REAL L_est_sup[], IN REAL
                                         D_est_inf[], IN REAL D_est_sup[], IN REAL f_prec)
                                         IN "optim_control.lib"
"C++" FUNCTION NO_TYPE optim_control   (FUNC_PTR fcn, IN REAL u_est[])
                                         IN "optim_control.lib" "nagcd.lib"

-- Horizonte de predicción
INTEGER Hp = 30
-- Horizonte de control
INTEGER H_con_comun = 3

-----
-- CLASS INHERITED FROM coste_default TO GENERATE AN OPTIMIZATION
-----
CLASS optimiz IS_A coste_default

DECLS

    -- Periodo de muestreo
    REAL per_muestreo = 0.2 "min"
    -- Número de variables manipuladas
    INTEGER nr_inputs = 2
    -- Horizonte de control para cada entrada
    CONST INTEGER H_con[2] = {3, 3}
    -- Utilizar el mismo horizonte para todas las entradas
    CONST INTEGER H_con_comun = 3
    -- Restricciones de camino que se incluirán en el problema de optimización
    INTEGER nr_path_constr = 0      -- Un 0 indica que las restricciones
                                    -- sobre los estados se expresan a través de
                                    -- funciones de penalización

    -- Límites en las variables manipuladas
    REAL Usup[2] = {4.4, 4.4}
    REAL Uinf[2] = {0.4, 0.4}

    REAL Dsup[2] = {0.1, 0.1}      -- límites en los incrementos
    REAL Dinf[2] = {-0.1, -0.1}

    REAL f_prec = 1.e-15          -- precisión de cálculo en el cálculo
                                    -- de la función objetivo

METHODS

    METHOD NO_TYPE ini ()

        BODY
            ini_control      (nr_inputs, H_con, nr_path_constr,
                            Uinf, Usup, Dinf, Dsup, f_prec)

        END METHOD

```

```

-----
-- CALLBACK METHOD TO BE PASSED TO OPTIMIZATION ROUTINE
-----
METHOD NO_TYPE objetivo (IN INTEGER n_est, IN REAL var_est[],
                        IN REAL func_res[])
  DECLS

  BODY

    RESET_VARIABLES()

    RESTORE_STATE("estado_actual.txt")

    EXEC_INIT()

    -- Pasar el horizonte de control al modelo
    setValueReal("NU[1]", H_con[1])
    setValueReal("NU[2]", H_con[2])

    setArray2D("uf", var_est, nr_inputs, H_con_comun)

    IF (nr_path_constr == 0) THEN
      setValueInt("ind_penal",1)
    ELSE
      setValueInt("ind_penal",0)
    END IF

    TIME = 0.0
    CINT = (Hp-1)*per_muestreo
    TSTOP = Hp*per_muestreo
    REL_ERROR = 1.e-09      --0.01*f_prec

    INIT_INTEG_STEP = 0.000001
    INTEG()

    -- Recoger el valor de la función objetivo
    func_res[1] = getValueReal("J")
    -- Recoger los valores de las funciones g(x)
    -- que definen las restricciones de camino
    func_res[2] = getValueReal("val_func[1]")
    func_res[3] = getValueReal("val_func[2]")

  END METHOD

-----
-- MAIN OPTIMIZATION METHOD
-----
METHOD NO_TYPE control (IN REAL con[])
  DECLS

  BODY

    -- Llamar al optimizador
    optim_control (objetivo, con)

  END METHOD

```

END CLASS

-- aquí acaba la definición de la clase optimiz

COMPONENT controlador

DATA

```
REAL Fo2c1inic = 1
REAL Fo2c2inic = 1
REAL Co2outinic = 24.26
REAL Tinic = 15
```

DECLS

```
DISCR REAL set_point
-- Variables que devuelve el controlador
DISCR REAL u_new[2], Fo2c1, Fo2c2

-- Variables que se leen del proceso (vía OPC)
REAL OPC_Co2out, OPC_Fo2c1, OPC_Fo2c2, OPC_T

-- Parámetros del control predictivo
INTEGER OPC_N2, OPC_Nu

-- Variables correspondientes al estado actual que se tienen que enviar al modelo
DISCR REAL Fo2c1i, Fo2c2i, Ti, Co2outi, refy

-- Periodo de muestreo
REAL tsamp = 0.2 "min"

-- Activar/desactivar la llamada al controlador cada periodo de muestreo
BOOLEAN sample = TRUE
INTEGER activate = 0
```

OBJECTS

optimiz modelo

INIT

```
modelo.ini ()

u_new[1] = Fo2c1inic
u_new[2] = Fo2c2inic

Fo2c1 = Fo2c1inic
Fo2c2 = Fo2c2inic

set_point = Co2outinic

OPC_Co2out = Co2outinic
OPC_Fo2c1 = Fo2c1inic
OPC_Fo2c2 = Fo2c2inic
OPC_T = Tinic

OPC_N2 = 30
```

```
OPC_Nu = 3
```

```
DISCRETE
```

```
-- Llamada al controlador predictivo
```

```
WHEN (sample == TRUE) THEN  
    activate = 0  
    sample = FALSE  
    sample = TRUE AFTER tsamp  
END WHEN
```

```
WHEN (activate == 1) THEN  
    -- Enviar al modelo el estado actual del proceso  
    Co2outi = OPC_Co2out  
    Fo2c1i = OPC_Fo2c1  
    Fo2c2i = OPC_Fo2c2  
    Ti = OPC_T  
    refy = set_point  
  
    Hp = OPC_N2  
    H_con_comun = OPC_Nu  
  
    SAVE_VARS("estado_actual.txt", "Co2outi Fo2c1i Fo2c2i Ti refy")  
  
    activate = 2  
END WHEN
```

```
WHEN (activate == 3) THEN  
    -- Recoger la nueva señal de control  
    modelo.control (u_new)  
  
    Fo2c1 = u_new[1]  
    Fo2c2 = u_new[2]  
  
    activate = 4  
END WHEN
```

```
END COMPONENT
```

Código del archivo coste MHE.el:

-- Modelo transferencia de O2

-- Función de coste para la estimación de la perturbación no medible (kla)

COMPONENT coste_MHE

DATA

REAL F = 0.55

REAL V = 0.84

REAL yo2c1 = 0.62

REAL yo2c2 = 0.5

REAL P_tot = 1

-- Parámetros del modelo (a, m, n) ($kla = m \cdot ((T-n)^{0.5}) \cdot Fo2^a$)

-- Rango de validez: T de 11 a 17°C y Fo2 de 0.4 a 4.4 L/min.

REAL a = 0.6822

REAL m = 0.7432

REAL n = 7.667

-- Valores iniciales (estado estacionario)

REAL C_O2_i = 24.26 -- INIT

-- Variables del controlador predictivo que tomarán los valores

-- recibidos desde el código del controlador

-- Periodo de muestreo

REAL tsamp = 0.5 "min"

-- Control futuro para minimizar y del modelo menos y medido

REAL uf_Fo2c1[4] = {1,1,1,1} -- viene de la clase estimacion (hija de
-- coste_MHE_default): vec_Fo2c1

REAL uf_Fo2c2[4] = {1,1,1,1}

-- Perturbación medible

REAL per_T[4] = {15,15,15,15}

-- Estado Medido

REAL per_CO2[4] = {24.26,24.26,24.26,24.26}

--Variable decisión

REAL d_i = 1 -- INIT

DECLS

REAL Co2in -- Concentración de O2 a la entrada (ppm)

REAL Co2out -- Concentración de O2 a la salida (ppm)

REAL Co2sat -- Concentración de saturación (ppm)

DISCR REAL Fo2c1 -- Flujo de gas colector 1 (L/min)

DISCR REAL Fo2c2 -- Flujo de gas colector 2 (L/min)

REAL Fo2 -- Flujo de gas total (L/min)

REAL yo2 -- Pureza (tanto por 1)

DISCR REAL T -- Temperatura (°C)

REAL H -- Constante de Henry (atm L/mg)

REAL kla -- Coeficiente de transferencia de materia (1/min)

DISCR REAL d -- se multiplica a kla

-- El índice que se minimiza

DISCR REAL J

-- Variable de estado medida

DISCR REAL ValorMedido

-- Variables bucles

INTEGER z

```

    BOOLEAN Sample = TRUE

INIT
    Co2out = C_O2_i -- viene de la clase estimación, hija de coste_MHE_default
    d = d_i -- viene de modelo_control (al final se pasa el estado actual del modelo)
    J = 0.0
    z = 0

DISCRETE
    WHEN (Sample) THEN
        z=z+1
        IF (z>4) THEN
            -- Función de coste:
            J = J + (ValorMedido - Co2out)**2
        END IF

        IF (z<=4) THEN
            Fo2c1 = uf_Fo2c1[z]
            Fo2c2 = uf_Fo2c2[z]
            T = per_T[z]
            ValorMedido = per_CO2[z]
        END IF
        Sample = FALSE
        Sample = TRUE AFTER tsamp
    END WHEN

CONTINUOUS
    -- ECUACIONES ANALÍTICAS
    -- Constante de Henry experimental
    H = 0.0005*T*(P_tot**(-2.34)) + 0.01315 -- H = 0.0005*T + 0.01315
                                           -- Hasta P = 0.1 barg

    -- Balance global
    Fo2 = Fo2c1 + Fo2c2
    -- Balance de materia al O2
    Fo2*yo2 = Fo2c1*yo2c1+Fo2c2*yo2c2
    -- Concentración de saturación a partir de la constante de Henry experimental
    Co2sat = P_tot*yo2/H
    -- Concentración de entrada a partir de la constante de Henry experimental
    Co2in = 1*0.21/H
    -- kla modelo: (kla = m*((T-n)^0.5)*Fo2^a)
    kla = d*(m*((T-n)**0.5)*Fo2**a)
    -- ECUACIONES DIFERENCIALES
    -- Balance de materia en fase líquida
    V*Co2out' = kla*V*(Co2sat-Co2out) + F*(Co2in-Co2out)

END COMPONENT

```

Código del archivo estimacion.el:

```

CLASS estimacion IS_A coste_MHE_default
  DECLS
    -- Estimación de parámetros
    REAL matrizValores[4,5]={{0,0,0,0,0},{0,0,0,0,0},{0,0,0,0,0},{0,0,0,0,0}}
    -- Periodo de muestreo
    REAL per_muestreo = 0.5 "min" -- 30 segundos
    -- Vectores
    REAL vec_Fo2c1[4]
    REAL vec_Fo2c2[4]
    REAL vec_T[4]
    REAL vec_CO2[4]

    -- Optimización
    -- Número de variables manipuladas
    INTEGER nr_inputs = 1 -- d
    -- Horizonte de control para cada entrada
    CONST INTEGER H_con[1] = {1}
    -- Utilizar el mismo horizonte para todas las entradas
    CONST INTEGER H_con_comun = 1
    -- Restricciones de camino que se incluirán en el problema de optimización
    INTEGER nr_path_constr = 0      -- Un 0 indica que las restricciones
                                   -- sobre los estados se expresan a través de
                                   -- funciones de penalización

    -- Límites en las variables manipuladas
    REAL Usup[1] = {1.5}      -- límite superior en el valor del control
    REAL Uinf[1] = {0.5}     -- límite inferior en el valor del control
    REAL Dsup[1] = {0.05}    -- límite superior en los incrementos
    REAL Dinf[1] = {-0.05}   -- límite inferior en los incrementos

    REAL f_prec = 1.e-05     -- precisión de cálculo en el cálculo
                                -- de la función objetivo

  METHODS
    METHOD NO_TYPE Actualizar (IN REAL Fo2c1_estim, IN REAL Fo2c2_estim, IN
                              REAL T_estim, IN REAL Co2_estim, IN INTEGER Posicion, IN
                              INTEGER TamanoVentana)

    DECLS
      BODY
        matrizValores[1,Posicion] = Fo2c1_estim
        matrizValores[2,Posicion] = Fo2c2_estim
        matrizValores[3,Posicion] = T_estim
        matrizValores[4,Posicion] = Co2_estim
    END METHOD

    METHOD NO_TYPE DesplazarVentana (IN REAL Fo2c1_estim, IN REAL
                                     Fo2c2_estim, IN REAL T_estim, IN REAL Co2_estim, IN
                                     INTEGER TamanoVentana)

    DECLS
      INTEGER p
      BODY
        IF (TamanoVentana>1) THEN
          FOR (p=1;p<=TamanoVentana;p=p+1)
            matrizValores[1,p] = matrizValores[1,p+1]
            matrizValores[2,p] = matrizValores[2,p+1]
          END FOR
        END IF
      END BODY
    END DECLS
  END METHODS

```

```

        matrizValores[3,p] = matrizValores[3,p+1]
        matrizValores[4,p] = matrizValores[4,p+1]
    END FOR
END IF
matrizValores[1,TamanoVentana] = Fo2c1_estim
matrizValores[2,TamanoVentana] = Fo2c2_estim
matrizValores[3,TamanoVentana] = T_estim
matrizValores[4,TamanoVentana] = Co2_estim
END METHOD

METHOD NO_TYPE ini ()
    BODY
        ini_control      (nr_inputs, H_con, nr_path_constr,
                        Uinf, Usup, Dinf, Dsup, f_prec)
    END METHOD

METHOD NO_TYPE objetivo (IN INTEGER n_est, IN REAL var_est[], IN REAL func_res[])
    DECLS
    INTEGER eme
    BODY
    RESET_VARIABLES()
    setValueReal("C_O2_i",matrizValores[4,1])
    -- Variable de decisión. Perturbación no medible
    setValueReal("d_i",var_est[1])
    EXEC_INIT()

    FOR (eme=2;eme<=5;eme=eme+1)
        vec_Fo2c1[eme-1] = matrizValores[1,eme]
        vec_Fo2c2[eme-1] = matrizValores[2,eme]
        vec_T[eme-1] = matrizValores[3,eme]
        vec_CO2[eme-1] = matrizValores[4,eme]
    END FOR

    setArray1D("uf_Fo2c1",vec_Fo2c1,2)
    setArray1D("uf_Fo2c2",vec_Fo2c2,2)
    setArray1D("per_T",vec_T,2)
    setArray1D("per_CO2",vec_CO2,2)

    ABS_ERROR=1E-5
    REL_ERROR=1E-5
    TIME = 0.0
    CINT = 0.0005
    TSTOP = 4*per_muestreo
    INTEG()
    -- Recoger el valor de la función objetivo
    func_res[1] = getValueReal("J")
END METHOD

METHOD NO_TYPE EstimacionPerturbacion (IN REAL con[])
    DECLS
    BODY
    -- Llamar al optimizador
    optim_control (objetivo, con)
END METHOD

END CLASS

```

Código del archivo modelo control.el:

```

-----
-- CLASS INHERITED FROM coste_default TO GENERATE AN OPTIMIZATION
-----
CLASS optimiz IS_A coste_default
  DECLS
    -- Horizonte de predicción
    INTEGER Hp = 30
    -- Periodo de muestreo
    REAL per_muestreo = 0.5 "min"
    -- Número de variables manipuladas
    INTEGER nr_inputs = 2
    -- Horizonte de control para cada entrada
    CONST INTEGER H_con[2] = {3, 3}
    -- Utilizar el mismo horizonte para todas las entradas
    CONST INTEGER H_con_comun = 3
    -- Restricciones de camino que se incluirán en el problema de optimización
    INTEGER nr_path_constr = 0      -- Un 0 indica que las restricciones
                                   -- sobre los estados se expresan a través de
                                   -- funciones de penalización

    -- Límites en las variables manipuladas
    REAL Usup[2] = {4.4, 4.4}
    REAL Uinf[2] = {0.4, 0.4}
    REAL Dsup[2] = {0.1, 0.1} -- límites en los incrementos
    REAL Dinf[2] = {-0.1, -0.1}
    REAL f_prec = 1.e-15      -- precisión de cálculo en el cálculo
                                   -- de la función objetivo

    -----
    -- Compensación de las perturbaciones no medibles
    REAL Perturbacion_NoMedible = 1
    -- Estado actual para calcular la acción de control
    REAL Co2out_estado = 24.26 -- Luego se asigna a Co2outi, a coste
    REAL Fo2c1_estado = 1
    REAL Fo2c2_estado = 1
    REAL T_estado = 15
    REAL refy_estado = 24.26

    -----
  METHODS
    METHOD NO_TYPE ini ()
      BODY
        ini_control      (nr_inputs, H_con, nr_path_constr,
                          Uinf, Usup, Dinf, Dsup, f_prec)
      END METHOD

    -----
    -- CALLBACK METHOD TO BE PASSED TO OPTIMIZATION ROUTINE
    -----
  METHOD NO_TYPE objetivo      (IN INTEGER n_est, IN REAL var_est[],
                              IN REAL func_res[])
    BODY
      RESET_VARIABLES()
      -- Se pasa el estado actual
      setValueReal("Co2outi",Co2out_estado)
      setValueReal("Fo2c1i",Fo2c1_estado)
      setValueReal("Fo2c2i",Fo2c2_estado)

```

```

setValueReal("Ti",T_estado)
setValueReal("refy",refy_estado)
setValueReal("Perturbacion",Perturbacion_NoMedible)
EXEC_INIT()
-- Pasar el horizonte de control al modelo
setValueReal("NU[1]", H_con[1])
setValueReal("NU[2]", H_con[2])
setValueReal("N2", Hp)
setArray2D("uf", var_est, nr_inputs, H_con_comun)
IF (nr_path_constr == 0) THEN
    setValueInt("ind_penal",1)
ELSE
    setValueInt("ind_penal",0)
END IF
ABS_ERROR=1E-5
REL_ERROR=1E-5
TIME = 0.0
CINT = (Hp-1)*per_muestreo
TSTOP = Hp*per_muestreo
INTEG()
-- Recoger el valor de la función objetivo
func_res[1] = getValueReal("J")
-- Recoger los valores de las funciones g(x)
-- que definen las restricciones de camino
func_res[2] = getValueReal("val_func[1]")
func_res[3] = getValueReal("val_func[2]")

END METHOD
-----
-- MAIN OPTIMIZATION METHOD
-----
METHOD NO_TYPE control (IN REAL con[])
    DECLS
    BODY
        -- Llamar al optimizador
        optim_control (objetivo, con)
    END METHOD
END CLASS

-----
-- PLANTA + LLAMADA AL CONTROLADOR
-----

COMPONENT modelo_control
    DATA
        -----
        INTEGER tamañoVentana = 5
        -----
        REAL F = 0.55
        REAL V = 0.84
        REAL yo2c1 = 0.62
        REAL yo2c2 = 0.5
        REAL P_tot = 1
        -- Parámetros del modelo (a, m, n) (kla = m*((T-n)0.5)*Fo2^a)
        -- Rango de validez: T de 11 a 17°C y Fo2 de 0.4 a 4.4 L/min.
        REAL a = 0.6822
        REAL m = 0.7432
        REAL n = 7.667

```

-- Valores iniciales (estado estacionario)

REAL Fo2c1inic = 1
 REAL Fo2c2inic = 1
 REAL Co2outinic = 24.26
 REAL Tinic = 15

 REAL dinic = 1 -- en INIT se pasa a d_new

-- Periodo de muestreo

REAL tsamp = 0.5 "min"

DECLS

REAL Co2in -- Concentración de O2 a la entrada (ppm)
 REAL Co2out -- Concentración de O2 a la salida (ppm)
 REAL Co2sat -- Concentración de O2 en saturación (ppm)
 REAL Fo2c1 -- Flujo de gas colector 1 (L/min)
 REAL Fo2c2 -- Flujo de gas colector 2 (L/min)
 REAL Fo2 -- Flujo de gas total (L/min)
 REAL yo2 -- Pureza (tanto por 1)
 REAL T -- Temperatura (°C)
 REAL H -- Constante de Henry (atm L/mg)
 REAL kla -- Coeficiente de transferencia de materia (1/min)
 DISCR REAL d -- Perturbación no medible
 REAL Co2out_max, Co2out_min, e

-- Valores iniciales (estado estacionario)

REAL Fo2c1i
 REAL Fo2c2i
 REAL Co2outi
 REAL Ti
 REAL d_i = 1 -- perturbación no medible

-- Referencias

REAL refy

-- Variables que devuelve el controlador

DISCR REAL u_new[2]
 DISCR REAL d_new[1]

-- Activar/desactivar la llamada al controlador cada periodo de muestreo

BOOLEAN sample = TRUE

--Estimación de parámetros

INTEGER posicion
 BOOLEAN ActivacionEstimacion

OBJECTS

estimacion Estimacion
 optimiz modelo

INIT

Fo2c1 = Fo2c1inic
 Fo2c2 = Fo2c2inic
 T = Tinic
 d_new[1] = dinic
 Co2out = Co2outinic
 u_new[1] = Fo2c1inic

```

u_new[2] = Fo2c2inic
refy = Co2outinic

-- Estimación de parámetros
posicion = 1

DISCRETE
-- Llamada al controlador predictivo
WHEN (sample == TRUE) THEN
-----
/*ESTIMACIÓN DE ESTADO Y PERTURBACIÓN NO MEDIBLE*/
-----
-- Estimación de parámetros
IF (posicion<=tamañoVentana) THEN
    Estimacion.Actualizar(Fo2c1i,Fo2c2i,Ti,Co2outi,posicion,tamañoVentana)
    posicion=posicion+1
ELSE
    ActivacionEstimacion=TRUE
    Estimacion.DesplazarVentana(Fo2c1i,Fo2c2i,Ti,Co2outi,tamañoVentana)
END IF
-- Cuando ya tengo la ventana para la estimación permito el control
IF (ActivacionEstimacion) THEN
    Estimacion.ini ()
    Estimacion.EstimacionPerturbacion(d_new)
    -----
    /*CONTROLADOR PREDICTIVO NO LINEAL*/
    -----
    --Llamada al controlador predictivo
    modelo.Perturbacion_NoMedible = d_new[1]
    modelo.Co2out_estado = Co2outi
    modelo.Fo2c1_estado = Fo2c1i
    modelo.Fo2c2_estado = Fo2c2i
    modelo.T_estado = Ti
    modelo.refy_estado = refy

SAVE_VARS("estado_actual.txt", "Fo2c1i Fo2c2i Ti Co2outi refy Error P_tot d_i")
-- Inicializar el controlador
modelo.ini()
modelo.control (u_new)
Fo2c1=u_new[1]
Fo2c2=u_new[2]
END IF
sample = FALSE
sample = TRUE AFTER tsamp
END WHEN

CONTINUOUS
-- Establecimiento de los valores máximos y mínimos
Co2out_max = Co2sat
Co2out_min = (F*Co2in + V*(m*((T-n)**0.5)*0.8**a*Co2sat))/\
(F+V*(m*((T-n)**0.5)*0.8**a))
-- ECUACIONES ANALÍTICAS
-- Constante de Henry experimental
H = 0.0005*T*(P_tot**(-2.34)) + 0.01315 -- H = 0.0005*T + 0.01315
-- Hasta P = 0.1 barg
-- Balance global
Fo2 = Fo2c1 + Fo2c2

```

```

-- Balance de materia al O2
Fo2*yo2 = (Fo2c1*yo2c1+Fo2c2*yo2c2)
-- Concentración de saturación a partir de la constante de Henry experimental
Co2sat = P_tot*yo2/H
-- Concentración de entrada a partir de la constante de Henry experimental
Co2in = 1*0.21/H
-- kla modelo: (kla = m*((T-n)^0.5)*Fo2^a)
kla = (m*((T-n)**0.5)*Fo2**a)

e = refy - Co2out

-- ECUACIONES DIFERENCIALES
V*Co2out' = kla*V*(Co2sat-Co2out) + F*(Co2in-Co2out)

/*GUARDAMOS EL ESTADO ACTUAL DEL PROCESO*/
Fo2c1i = Fo2c1 -- Flujo de gas colector 1(L/min)
Fo2c2i = Fo2c2 -- Flujo de gas colector 2 (L/min)
Ti = T -- Temperatura (°C)
d_i = d -- Perturbación no medible
Co2outi = Co2out -- Concentración de O2 a la salida (ppm)

```

END COMPONENT

Código del archivo controlador2.el:

```

-----
-- EXTERNAL FUNCTION
-----
"C++" FUNCTION NO_TYPE ini_control      (IN INTEGER nr_ent, IN INTEGER Nu[], IN INTEGER n_res,
                                        IN REAL L_est_inf[], IN REAL L_est_sup[], IN REAL
                                        D_est_inf[], IN REAL D_est_sup[], IN REAL f_prec)
                                        IN "optim_control.lib"
"C++" FUNCTION NO_TYPE optim_control  (FUNC_PTR fcn, IN REAL u_est[])
                                        IN "optim_control.lib" "nagcd.lib"

-- Horizonte de predicción
INTEGER Hp = 30
-----
-- CLASS INHERITED FROM coste_default TO GENERATE AN OPTIMIZATION
-----
CLASS optimiz IS_A coste_default
  DECLS
    -- Periodo de muestreo
    REAL per_muestreo = 0.5 "min"
    -- Número de variables manipuladas
    INTEGER nr_inputs = 2
    -- Horizonte de control para cada entrada
    CONST INTEGER H_con[2] = {3, 3}
    -- Utilizar el mismo horizonte para todas las entradas
    CONST INTEGER H_con_comun = 3
    -- Restricciones de camino que se incluirán en el problema de optimización
    INTEGER nr_path_constr = 0      -- Un 0 indica que las restricciones
                                   -- sobre los estados se expresan a través de
                                   -- funciones de penalización

    -- Límites en las variables manipuladas
    REAL Usup[2] = {4.4, 4.4}
    REAL Uinf[2] = {0.4, 0.4}
    REAL Dsup[2] = {0.1, 0.1} -- límites en los incrementos
    REAL Dinf[2] = {-0.1, -0.1}
    REAL f_prec = 1.e-15      -- precisión de cálculo en el cálculo
                                   -- de la función objetivo

    -- Compensación de las perturbaciones no medibles
    REAL Perturbacion_NoMedible=1

  METHODS
    METHOD NO_TYPE ini ()
      BODY
        ini_control      (nr_inputs, H_con, nr_path_constr,
                          Uinf, Usup, Dinf, Dsup, f_prec)

  END METHOD
-----
-- CALLBACK METHOD TO BE PASSED TO OPTIMIZATION ROUTINE
-----
METHOD NO_TYPE objetivo (IN INTEGER n_est, IN REAL var_est[], IN REAL func_res[])
  DECLS
  BODY
    RESET_VARIABLES()
    RESTORE_STATE("estado_actual.txt")
    EXEC_INIT()

```

```

--Pasar la perturbacion estimada
setValueReal("Perturbacion",Perturbacion_NoMedible)
-- Pasar el horizonte de control al modelo
setValueReal("NU[1]", H_con[1])
setValueReal("NU[2]", H_con[2])
setArray2D("uf", var_est, nr_inputs, H_con_comun)
IF (nr_path_constr == 0) THEN
    setValueInt("ind_penal",1)
ELSE
    setValueInt("ind_penal",0)
END IF
TIME = 0.0
-- Especificar el intervalo de comunicación del experimento
-- que simula la función objetivo
CINT = (Hp-1)*per_muestreo
TSTOP = Hp*per_muestreo
REL_ERROR = 1.e-09 --0.01*f_prec
INIT_INTEG_STEP = 0.000001
INTEG()
-- Recoger el valor de la función objetivo
func_res[1] = getValueReal("J")
-- Recoger los valores de las funciones g(x)
-- que definen las restricciones de camino
func_res[2] = getValueReal("val_func[1]")
func_res[3] = getValueReal("val_func[2]")
END METHOD
-----
-- MAIN OPTIMISATION METHOD
-----
METHOD NO_TYPE control (IN REAL con[])
    DECLS
    BODY
        -- Llamar al optimizador
        optim_control (objetivo, con)
    END METHOD
END CLASS

COMPONENT controlador2
    DATA
        REAL Fo2c1inic = 1
        REAL Fo2c2inic = 1
        REAL Co2outinic = 24.26
        REAL Tinic = 15
        REAL yo2c2inic = 0.5
        REAL P_totinic = 1

        -- Periodo de muestreo
        REAL tsamp = 30

        INTEGER tamañoVentana = 5

    DECLS
        DISCR REAL set_point
        -- Variables que devuelve el controlador
        DISCR REAL u_new[2], Fo2c1, Fo2c2
        DISCR REAL d_new[1]

```

```

-- Variables que se leen del proceso (vía OPC)
REAL OPC_Co2out, OPC_Fo2c1, OPC_Fo2c2, OPC_T, OPC_yo2c2, OPC_P
-- Parámetros del control predictivo
INTEGER OPC_N2, OPC_Nu
-- Variables correspondientes al estado actual que se tienen que enviar al modelo
DISCR REAL Fo2c1i, Fo2c2i, Ti, Co2outi, refy, yo2c2, P_tot
-- Activar/desactivar la llamada al controlador cada periodo de muestreo
BOOLEAN sample = TRUE
INTEGER activate = 0

BOOLEAN ActivacionEstimacion
INTEGER posicion

```

OBJECTS

```

optimiz modelo
estimacion Estimacion

```

INIT

```

u_new[1] = Fo2c1inic
u_new[2] = Fo2c2inic

d_new[1] = 1

Fo2c1 = Fo2c1inic
Fo2c2 = Fo2c2inic

set_point = Co2outinic

OPC_Co2out = Co2outinic
OPC_Fo2c1 = Fo2c1inic
OPC_Fo2c2 = Fo2c2inic
OPC_T = Tinic
OPC_yo2c2 = yo2c2inic
OPC_P = P_totinic

OPC_N2 = 30
OPC_Nu = 3

ActivacionEstimacion=FALSE

```

DISCRETE

```

-- Llamada al controlador predictivo
WHEN (sample == TRUE) THEN
    activate = 0
    sample = FALSE
    sample = TRUE AFTER tsamp
END WHEN

WHEN (activate == 1) THEN
-- Enviar al modelo el estado actual del proceso
    Co2outi = OPC_Co2out
    Fo2c1i = OPC_Fo2c1
    Fo2c2i = OPC_Fo2c2
    Ti = OPC_T
    refy = set_point
    yo2c2 = OPC_yo2c2

```

```

P_tot = OPC_P
Hp = OPC_N2
H_con_comun = OPC_Nu
-----
-- Estimación de parámetros
IF (posicion<=tamanoVentana) THEN
Estimacion.Actualizar(Fo2c1i,Fo2c2i,Ti,Co2outi,posicion,tamanoVentana)
posicion=posicion+1
ELSEIF(posicion>tamanoVentana) THEN
ActivacionEstimacion=TRUE
Estimacion.DesplazarVentana(Fo2c1i,Fo2c2i,Ti,Co2outi,tamanoVentana)
END IF
-----
SAVE_VARS("estado_actual.txt", "Co2outi Fo2c1i Fo2c2i Ti refy yo2c2 P_tot")
activate = 2
END WHEN

WHEN (activate == 3) THEN
-----
IF (ActivacionEstimacion) THEN
    Estimacion.ini ()
    Estimacion.EstimacionPerturbacion(d_new)
    modelo.Perturbacion_NoMedible = d_new[1]
END IF
-----
-- Recoger la nueva señal de control
modelo.ini()
modelo.control (u_new)
Fo2c1 = u_new[1]
Fo2c2 = u_new[2]
activate = 4
END WHEN

END COMPONENT

```

12. BIBLIOGRAFÍA

- [1] EcosimPro by EA Internacional, Dynamic Modeling & Simulation Tool, [On line] <http://www.ecosimpro.com/>. Último acceso: Mayo de 2013.
- [2] Martí, R.; Sarabia, D; De Prada, C. "Construcción, simulación y control de una red de distribución de oxígeno", *Departamento de Ingeniería de Sistemas y Automática, Universidad de Valladolid. Julio de 2011.*
- [3] De Prada, César, "El futuro del control de procesos", *Revista Iberoamericana de automática e informática industrial. CEA-IFAC, Vol-4, pp. 5-20, 2004.*
- [4] Valbuena, M., Sarabia, D., De Prada, C. "Simulación dinámica de un colector de H2 basada en el método de diferencias finitas". *CEA 2009, Universidad de Valladolid.*
- [5] Marcos, Natalia I.; Forbes, J. Fraser; Guay, Martin, "Coordination of Distributed Model Predictive Controllers for Constrained Dynamic Processes", *Computers and Chemical Engineering, pp. 1507-1522, 2008.*
- [6a] Alves, R., Normey-Rico, J., Merino, A., Acebes, L.F., Prada, C., (2006), EDUSCA (Educational Scada): Features and applications, *7th IFAC Symposium in Advances in Control Education.*
- [6b] Manual EDUSCA: Raúl Alves Santos, Julio E. Normey Rico, Alejandro Merino, Cesar de Prada. "EDUSCA (EDUCATIONAL SCADA): Features and Applications", *ACE 2006 7th IFAC Symposium in Advances in Control Education. Madrid, Spain, 2006.*
- [7] Bird, R., Stewart, W., Lightfoot, E., "Transport Phenomena", *2nd Edition, John Wiley & Sons, 2002.*
- [8] Garcia-Ochoa, F., Gomez, E., (2004) "Theoretical prediction of gas-liquid mass transfer coefficient, specific area and hold-up in sparged stirred tanks". *Chemical Engineering Science, 59, 2489-2501.*
- [9] Dudley. J (1995) "Mass transfer in bubble columns: A comparison of correlations". *Water Research, Vol. 29, No. 4, 1129-1138.*
- [10] Kawase, Y., Halard, B., Moo-Young, M. (1987) "Theoretical prediction of volumetric mass transfer coefficients in bubble columns for Newtonian and non-newtonian fluids". *Chemical Engineering Science, Vol.42, No.7, 1609-1617.*
- [11] Perry, R., Green, D., (1997), "Perry's Chemical Engineers' Handbook", *7th Edition, 1997, Ed. McGraw-Hill.*
- [12] MathWorks, Matlab. Curve Fitting Tool [On line] <http://www.mathworks.es/>. Último acceso: Mayo de 2013.
- [13] Camacho, F., Bordons, C., "Model predictive control". *Springer-Verlag London, 2004.*

- [14] Richalet, J., Rault, A., Papon, J. L., Testud, J. "Model predictive 6Heuristic control: applications to industrial processes", *Automatica. Volume 14, Issue 5, September 1978, Pages 413–428.*
- [15] Cutler, C. R. and Ramaker, B. L. "Dynamic matrix control and computer control algorithm", *Joint American Control Conference, 1980.*
- [16] Scattolini, R. "Architectures for distributed and hierarchical Model Predictive Control- A review", *Journal of Process Control. Volume 19, Issue 5, May 2009, Pages 723–731.*
- [17] Camacho, F., Bordons, C., (2004), "Control predictivo: pasado, presente y futuro". 2004 CEA-IFAC.
- [18] NAG, "Numerical Algorithms Group", [On line] <http://www.nag.co.uk/>, 1990. Último acceso: Mayo de 2013.
- [19] Allgöwer, F., Badgwell, T. A., Qin, J. S., Rawlings, J. B., Wright, S. J. "Nonlinear Predictive Control and Moving Horizon Estimation. An Introductory Overview". *Advances in Control, 1999, pp 391-449.*
- [20a] Acebes, L. F., Alves, R., Merino, A., (2004), Prada, C., (2004) "Un entorno de modelado inteligente y simulación distribuida de plantas de proceso", *Revista Iberoamericana de Automática e Informática Industrial. Vol. 1, Núm. 2.*
- [20b] Manual CreaOPC: EcoOPC. Interfaz OPC para experimentos en EcosimPro. Julio 2008. CTA.
- [21] Zamarreño, J. M. (2001). "Entorno de comunicaciones OPC para Ecosim". *Primeras Jornadas de Usuarios de EcosimPro. Madrid, 2001.*