

Informe Técnico – Technical Report

DPTOIA-IT-2002-005

Marzo, 2002

MODELADO DE UN FERMENTADOR MEDIANTE REDES NEURONALES

Mario Francisco Sutil

Pastora Isabel Vega Cruz



Departamento de Informática y Automática
Universidad de Salamanca

Revisado por:

Dra. Belén Pérez Lancho
Área de Ingeniería de Sistemas y Automática
Departamento de Informática y Automática
Facultad de Ciencias – Universidad de Salamanca
Plaza de la Merced s/n – 37008 Salamanca

Dr. Ángel Luis Sánchez Lázaro
Área de Lenguajes y Sistemas Informáticos
Departamento de Informática y Automática
Facultad de Traducción y Documentación
C/ Fco. Vitoria 6-16 - 37008 Salamanca

Aprobado en el Consejo de Departamento del 27 de marzo de 2002

Información de los autores:

Mario Francisco Sutil
Área de Ingeniería de Sistemas y Automática
Departamento de Informática y Automática
E.T.S. de Ingeniería Industrial de Béjar – Universidad de Salamanca
Av. Fernando Ballesteros, 2 – 37700 Béjar (España)
mfs@usal.es

Dra. Pastora Isabel Vega Cruz
Área de Ingeniería de Sistemas y Automática
Departamento de Informática y Automática
E.T.S. de Ingeniería Industrial de Béjar – Universidad de Salamanca
Av. Fernando Ballesteros, 2 – 37700 Béjar (España)
pvega@usal.es

Este documento puede ser libremente distribuido.

© 2002 Departamento de Informática y Automática - Universidad de Salamanca.

Resumen

En este trabajo se han desarrollado algunos métodos avanzados para modelar un fermentador dedicado a la producción de levaduras. En primer lugar, se describe el proceso y su modelo matemático, explicando las razones que hacen necesario el uso de nuevas técnicas. En segundo lugar, se utiliza una red neuronal en el espacio de estados (RNEE) para modelar el proceso globalmente, dando buenos resultados para datos de validación típicos. Posteriormente se ha utilizado otra técnica distinta consistente en considerar múltiples modelos lineales locales, en la cual se han estudiado dos métodos para identificar estos modelos lineales locales. En el primero se entrena una RNEE lineal para cada uno de los modelos locales, y en el segundo se linealiza el modelo global obtenido mediante la RNEE alrededor de los puntos donde están colocados los modelos locales. Ambos métodos dan buenos resultados, aunque el comportamiento del segundo es mejor debido a la robustez del modelo RNEE global.

Abstract

In this work, some advanced methods have been developed to model a fermenter for baker's yeast production. Firstly, the process and its mathematical model are described, explaining the reasons that make necessary the use of new techniques. Secondly, a state space neural network (SSNN) is used to model the process globally, giving good results for typical validation data. Then, a linear local modelling approach is presented, and two methods are considered to identify this local models. The first one allows to train a linear SSNN for every local model, and the second one allows to linearize the global model obtained with the SSNN around the points where local models are placed. Both methods give good results, but the accuracy of the second is better due to the robustness of the SSNN global model.

Tabla de Contenidos

1.	INTRODUCCIÓN.....	1
2.	FERMENTADOR PARA LA PRODUCCIÓN DE LEVADURAS.....	1
3.	MODELADO DEL FERMENTADOR UTILIZANDO LA RNEE	3
4.	MODELADO DEL FERMENTADOR MEDIANTE MÚLTIPLES MODELOS LINEALES LOCALES	6
4.1.	Red de base radial para solapar los modelos locales.....	7
4.2.	Utilización de una red competitiva para inicializar los centros de las gaussianas de la RBFN	8
4.3.	Equivalencia entre la RBFN y un sistema de inferencia difuso.....	9
4.4.	Modelos locales obtenidos mediante una RNEE lineal.	9
4.5.	Modelos locales obtenidos linealizando a partir del modelo RNEE global.....	12
4.6.	Determinación de las matrices K por ubicación de polos	16
5.	CONCLUSIONES Y TRABAJO FUTURO.....	17
6.	REFERENCIAS	18

1. INTRODUCCIÓN

Los procesos biotecnológicos, en particular los procesos de fermentación, son bastante complicados de modelar debido a la presencia de microorganismos cuyo comportamiento es altamente no lineal y variable en el tiempo. Además, muchos de los parámetros del proceso son inciertos o sólo parcialmente conocidos.

El modelado clásico es el basado en *primeros principios* que describan las leyes internas del sistema: se plantean ecuaciones de balance de masa y energía, ecuaciones cinéticas y relaciones de transferencia de masa. Los modelos matemáticos obtenidos con esta técnica suelen tener poca capacidad de predicción cuando se comparan con los procesos reales debido fundamentalmente al comportamiento no lineal intrínseco del metabolismo celular.

Otra dificultad a la hora de modelar procesos biotecnológicos es que suelen verse sometidos a perturbaciones de carga significativas y a veces es imposible obtener on-line variables importantes del proceso.

Por todo lo anterior, la utilización de redes neuronales es una buena alternativa cuando tratamos de modelar estos sistemas. Las redes neuronales poseen una gran flexibilidad para modelar dinámicas no lineales complejas, se aplican fácilmente a sistemas multivariados y son robustas ante la presencia de fallos. Otra ventaja de la utilización de las redes es su buena capacidad de modelado solamente a partir de datos experimentales del proceso a identificar, que es lo que normalmente tendremos a nuestra disposición.

En este trabajo se obtendrán distintos modelos neuronales para un fermentador semicontinuo dedicado a la producción de levaduras. Partiendo de un modelo neuronal global se obtienen otros basados en modelos lineales locales.

2. FERMENTADOR PARA LA PRODUCCIÓN DE LEVADURAS

El sistema objeto de nuestro estudio es un fermentador (reactor químico biológico) en el cual tienen lugar una serie de reacciones químicas complejas cuyo objetivo es la producción de la levadura *Saccharomyces Cerevisiae*, utilizada posteriormente en las industrias panificadoras. Se trata de un fermentador semicontinuo, es decir, existe un caudal de entrada de forma que el proceso tiene lugar hasta que se alcanza un volumen máximo.

Esta levadura está compuesta esencialmente por células vivas del microorganismo *Saccharomyces Cerevisiae*. Nutricionalmente, las levaduras son microorganismos heterotróficos, es decir, pueden obtener la energía y el carbono necesarios para crecer a partir de compuestos orgánicos, en nuestro caso, a partir de la glucosa que se introduce con el caudal de entrada. La cantidad de oxígeno existente induce alteraciones en el metabolismo de las levaduras, de tal forma que crecen tanto en ausencia de aire (fermentación) como en su presencia (respiración o metabolismo oxidativo).

Las variables de interés en nuestro fermentador son las concentraciones de los siguientes compuestos: biomasa o levadura (X), sustrato o glucosa (S), etanol (E), oxígeno disuelto (O), dióxido de carbono disuelto (C) y concentración de glucosa en el caudal de entrada (S_e). Los tres caminos metabólicos para el crecimiento de la levadura (X) son los siguientes:

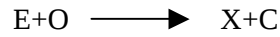
- Crecimiento respiratorio en glucosa (1)



- Crecimiento fermentativo en glucosa (2)



- Crecimiento respiratorio en etanol (3)



Teniendo en cuenta el hecho de que las vías metabólicas que producen la biomasa son distintas según la cantidad de oxígeno disponible en el medio, tenemos dos modos de producción: Cuando hay oxígeno en exceso en el medio, tienen lugar la reacciones (1) y (3), mientras que si el oxígeno existente no es suficiente para oxidar toda la glucosa, tienen lugar (1) y (2).

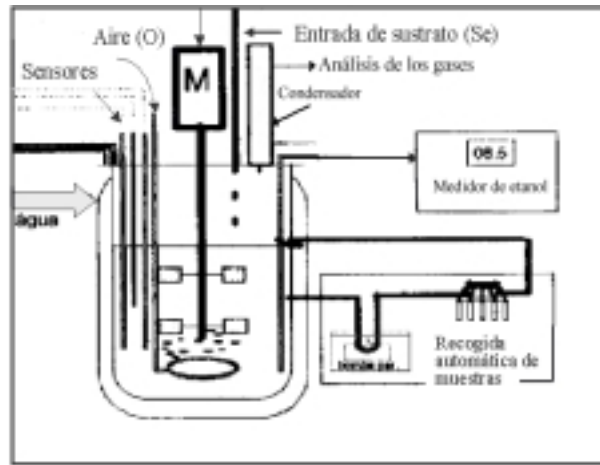


Figura 1: Esquema del dispositivo experimental del fermentador

El modelo matemático del fermentador basado en primeros principios se obtiene a partir de los balances de masa para todos los componentes, considerando que el reactor está bien agitado. Si no tenemos en cuenta la dinámica de la fase gaseosa ni la del dióxido de carbono, el conjunto de ecuaciones es el siguiente:

$$\begin{aligned} \frac{dX}{dt} &= -DX + (\mu_s^o + \mu_r^s + \mu_e^o)X \\ \frac{dS}{dt} &= -DS + DS_e + \left(-\frac{\mu_s^o}{Y_{x/s}^o} - \frac{\mu_r^s}{Y_{x/s}^r}\right)X \\ \frac{dE}{dt} &= -DE + \left(\frac{\mu_r^s}{Y_{x/e}^r} - \frac{\mu_e^o}{Y_{x/e}^{oe}}\right)X \\ \frac{dO}{dt} &= -DO + OTR + \left(-\frac{\mu_s^o}{Y_{x/o}^o} - \frac{\mu_e^o}{Y_{x/o}^{oe}}\right)X \\ \frac{dV}{dt} &= -DV \end{aligned}$$

$$D = \frac{F}{V}$$

$$OTR = K_L a (O^* - O)$$

donde

V = volumen

D = tasa de dilución

F = caudal de entrada

OTR = tasa de transferencia de oxígeno de la fase gaseosa a la fase líquida

O* = Concentración de saturación de oxígeno disuelto

K_La = Coeficiente global de transferencia de masa

μ = Tasas de crecimiento de la biomasa

Y = coeficientes de crecimiento

Los dos modos de crecimiento de la levadura antes mencionados quedan determinados por distintos valores de las tasas de crecimiento en las ecuaciones del modelo:

$$\begin{aligned} aq_s &\leq q_o \\ \mu_s^o &= Y_{x/s}^o q_s \\ \mu_s^r &= 0 \\ \mu_e^o &= \min(\mu_{e1}^o, \mu_{e2}^o) \end{aligned}$$

Régimen respiratorio

$$\begin{aligned} aq_s &\geq q_o \\ \mu_s^o &= Y_{x/s}^o \frac{q_o}{a} \\ \mu_s^r &= Y_{x/s}^r (q_s - \frac{q_o}{a}) \\ \mu_e^o &= 0 \end{aligned}$$

Régimen respiro-fermentativo

Utilizando este modelo matemático, se ha simulado el sistema mediante el lenguaje ACSL. En las simulaciones se ha fijado en 20 horas la duración de cada lote y se ha hecho un muestreo con un periodo de 0.1 horas obteniendo así 201 datos de cada lote. Estos datos son los que en el resto del trabajo se han considerado como experimentales, es decir, las redes neuronales se han entrenado y validado con estos datos.

3. MODELADO DEL FERMENTADOR UTILIZANDO LA RNEE

En primer lugar se ha realizado la identificación del fermentador utilizando un modelo neuronal, en concreto, la red neuronal en el espacio de estados. Esta red consta de cinco capas tal como se muestra en la Figura 2. La primera capa simplemente recibe las señales de entrada y las distribuye a la siguiente. La segunda capa representa la función no lineal f. La tercera es la capa de estados, donde cada neurona representa un estado del sistema. La cuarta capa representa la función no lineal g, y la capa de salida recibe las señales de la capa anterior y produce la salida. Las ecuaciones de la red son las siguientes:

$$\begin{aligned}\hat{\mathbf{x}}(t+1) &= \mathbf{W}^h \cdot f(\mathbf{W}^r \hat{\mathbf{x}}(t) + \mathbf{W}^i \mathbf{u}(t) + \mathbf{B}^h + \mathbf{W}^e \mathbf{e}(t)) \\ \hat{y}(t) &= \mathbf{W}^o \cdot g(\mathbf{W}^{h2} \hat{\mathbf{x}}(t) + \mathbf{B}^{h2}) \\ \mathbf{e}(t) &= y(t) - \hat{y}(t)\end{aligned}$$

La primera ecuación representa la evolución no lineal de los estados, incluyendo las entradas y un término de corrección (diferencia entre la salida real y la que daría la red) ante posibles señales estocásticas que actúen sobre el proceso. La segunda calcula las salidas como una función no lineal de los estados.

Antes de entrenar la red, lo primero que se debe hacer es elegir adecuadamente las entradas y salidas, así como particularizar su arquitectura para el sistema que nos ocupa. En el caso del fermentador, me interesa obtener el valor de las variables de estado (S,X,O,E,V), luego únicamente se necesitan las tres primeras capas de la RNEE ya que los estados son a su vez las salidas. En cuanto a las entradas, se han seleccionado las dos siguientes: el caudal de entrada al fermentador (F), que podría actuar como variable de control, y la concentración de glucosa en el caudal de entrada (Se), que podría considerarse como perturbación al sistema. Para la capa oculta, 11 neuronas es el número que proporciona mejores resultados. En resumen, la red consta de 3 capas y su topología es 2-11-5. Las funciones de transferencia de las neuronas de las capas 1ª y 3ª son lineales, mientras que en las neuronas de la capa oculta son funciones no lineales, de tipo sigmoide.

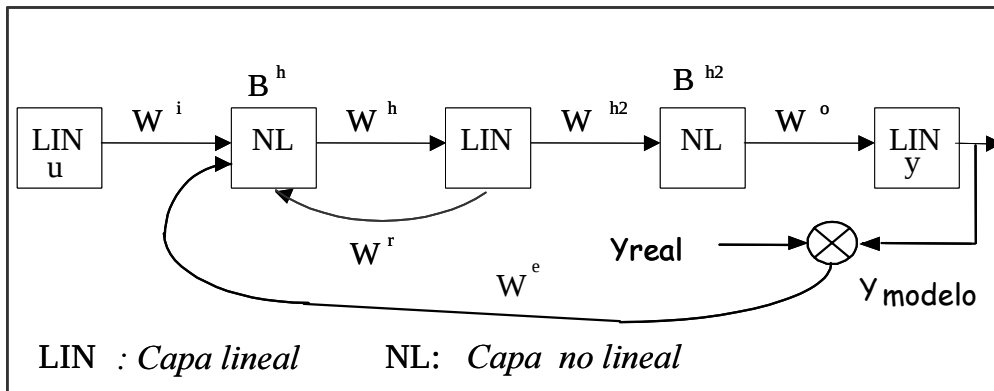


Figura 2: Arquitectura de la red neuronal en el espacio de estados (RNEE)

El entrenamiento de la red ha sido realizado mediante el Método de Optimización Aleatoria Modificado (MOAM) (Baba, 1989), que es una técnica aleatoria de minimización del error. Las entradas utilizadas en el entrenamiento han sido del tipo ‘trenes de saltos’ de distintas frecuencias y con distintos valores iniciales. En todos los casos, el periodo de muestreo es $T = 0.1$ h, el tiempo de operación del fermentador son 20 h, ya que se trata de un proceso por lotes, y el número de iteraciones para entrenar ha sido 50.000. Dado que es una red recurrente, la inicialización se hace tomando los valores reales de los estados en $t=0$.

Tras haber entrenado la red, se consigue que para los datos de entrenamiento las salidas que nos proporciona la red (en rojo) y las reales del sistema (en azul) sean muy similares, como se observa en la figura 3.

En la figura 4 se muestran las salidas que proporciona la red comparadas con las reales para entradas distintas a las de entrenamiento (datos de validación). De la figura se puede observar que la red se comporta bastante bien con respecto a entradas desconocidas para ella, es decir, el

error entre la salida que proporciona la red y la salida real del sistema no es muy grande en relación con las magnitudes absolutas de las variables. Sólo se observa un pequeño offset que se puede corregir si al simular se introduce como término de corrección el error anterior.

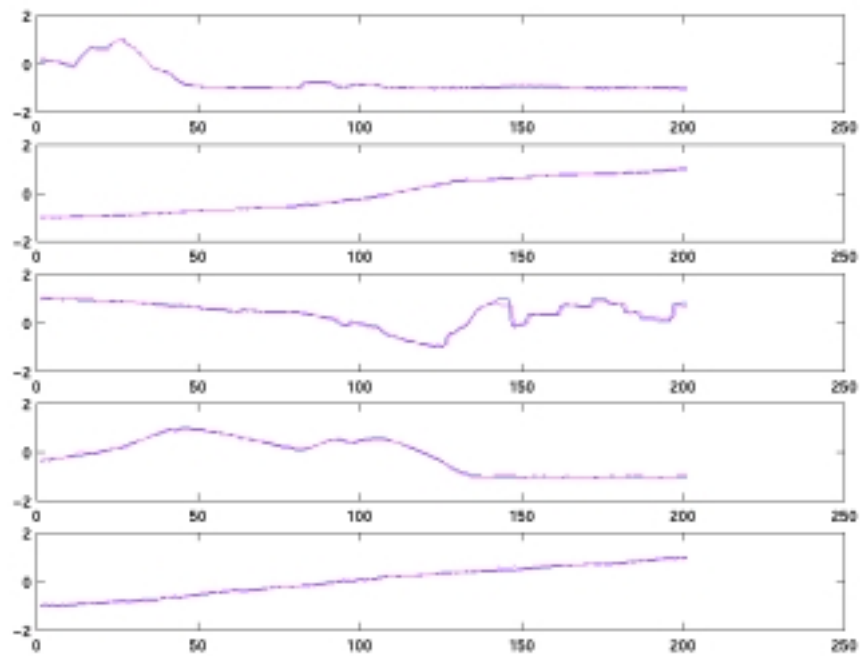


Figura 3: Salidas de la red (rojo) y reales (azul) normalizadas (S,X,O,E,V) para los datos de entrenamiento.

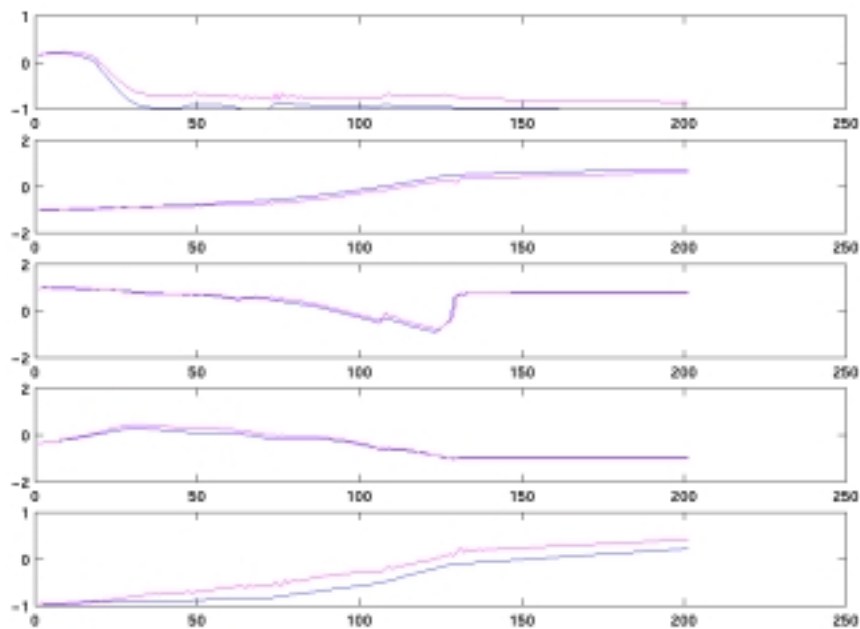


Figura 3: Salidas de la red (rojo) y reales (azul) normalizadas (S,X,O,E,V) para los datos de validación.

4. MODELADO DEL FERMENTADOR MEDIANTE MÚLTIPLES MODELOS LINEALES LOCALES

Otra posibilidad a la hora de modelar el fermentador es considerar modelos locales que se solapen de algún modo para obtener las salidas del proceso. Debido a la relativa sencillez para identificar modelos lineales, se utilizarán modelos locales lineales. Principalmente se han utilizado dos técnicas para identificar los modelos locales:

- Linealizando a partir del modelo global obtenido mediante la RNEE en determinados puntos.
- Entrenando una RNEE lineal para identificar cada uno de los modelos locales.

Antes de entrar en detalles respecto a estos dos puntos, vamos a describir la metodología general seguida en la obtención del modelo del fermentador.

Como para cualquier proceso no lineal, las ecuaciones del fermentador son de la forma siguiente, donde x es el vector de estados, u es el vector de entradas, y es el vector de salidas, y f, g son dos funciones no lineales:

$$\begin{cases} \dot{x} = f(x, u) \\ y = g(x) \end{cases}$$

Nuestra intención es desarrollar un modelo global que esté formado por la contribución de modelos lineales locales de la siguiente forma, donde A, B, C son las matrices en el espacio de estados:

$$\begin{cases} \dot{x} = Ax + Bu \\ y = Cx \end{cases}$$

En cuanto a la transición entre modelos locales, la validez de uno u otro modelo no está marcada por fronteras rígidas sino que existe solapamiento entre ellos. Las ecuaciones del modelo global son entonces una suma ponderada de las contribuciones de cada modelo local:

$$\begin{cases} \dot{x} = \sum_{i=1}^k (A_i x + B_i u) \cdot \varphi_i(x) \\ y = Cx \end{cases}$$

La función $\varphi_i(x)$ ($i = 1, 2, \dots, k$) proporciona el grado de validez entre 0 y 1 del i -ésimo modelo para cada punto del espacio de estados, de forma que la salida 'y' en un determinado punto x normalmente tendrá parte de varios modelos locales. En el caso del fermentador seguiremos considerando que las salidas son los estados, o lo que es lo mismo, $C = Id$, por lo que la ecuación de salida no se tendrá en cuenta.

4.1. Red de base radial para solapar los modelos locales

La forma de la función φ_i es importante para determinar cómo se solapan los modelos locales. En nuestro caso φ_i viene determinada por una red de funciones de base radial (RBFN) (ver figura 5) en la que cada neurona oculta representa un modelo local.

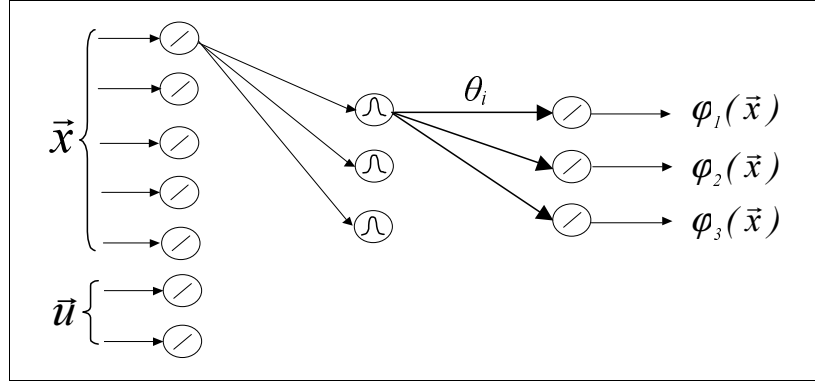


Figura 4: Red de base radial para 3 modelos locales

Las funciones base f_i que hemos utilizado son gaussianas cuyos centros son los vectores c_i . Los parámetros θ_i son los pesos de la capa de salida de la red y $\| \cdot \|$ es la norma Euclídea. La ecuación matemática de la red es la siguiente:

$$\varphi_i(\vec{x}) = \sum_{i=1}^k \vartheta_i \cdot f_i(\|\vec{c}_i - \vec{x}\|)$$

La entrada de la red son los distintos vectores de estado x , que no son otra cosa que los datos de salida del fermentador (variables de estado: S, X, E, O, V), más las entradas correspondientes a ese vector de estado (caudal de entrada: F, concentración de sustrato en el caudal de entrada: Se). Las salidas de la red son los grados de pertenencia de ese dato a cada uno de los k modelos locales, luego el número de neuronas en la capa de salida es siempre igual al número de neuronas ocultas. Tanto los vectores de entrada a la red como los centros de las gaussianas, se encuentran en un espacio de 7 dimensiones. Básicamente, los grados de pertenencia vendrán determinados por la proximidad o lejanía del dato a cada uno de los centros debido a la forma de las gaussianas. Como caso extremo, si un determinado vector de entrada es igual a uno de los c_i , por ejemplo c_1 , la correspondiente salida $\varphi_1(x)$ será 1, mientras que el resto de las $\varphi_i(x)$ valdrá 0. Esto quiere decir que para ese dato la validez del modelo representado por c_1 es total, mientras que el resto de los modelos tienen validez nula.

Los pesos θ_i no se entrenan como en otras redes neuronales, sino que se calculan para que las salidas estén normalizadas, es decir, la suma de los grados de pertenencia a cada uno de los modelos sea igual a la unidad para todos los datos presentados a la entrada. En cuanto a las anchuras de las gaussianas, se han tomado siempre varianzas de 0.5 para todas las dimensiones. Se han realizado pruebas entrenando dichas varianzas, y los resultados obtenidos son siempre peores debido al sobreentrenamiento del modelo.

4.2. Utilización de una red competitiva para inicializar los centros de las gaussianas de la RBFN

Para colocar inicialmente los centros es necesario disponer de datos experimentales, que nosotros llamaremos datos de entrenamiento y son los datos de un lote determinado. Estos datos forman una nube de 201 puntos en el espacio de estados de 7 dimensiones (ver figuras 5 y 6 para cuatro de las cinco variables de estado). Mediante una red competitiva se realizan agrupaciones de estos datos en k grupos, colocándose los pesos de la red competitiva en los lugares de mayor densidad de puntos. Estos vectores de pesos, una vez entrenados, son los que se toman como centros de las gaussianas de la RBFN, es decir, son los puntos en los que un determinado modelo local tiene máxima validez. En las figuras 5 y 6 se observan dichos centros representando los modelos locales utilizados para el modelado global del fermentador.

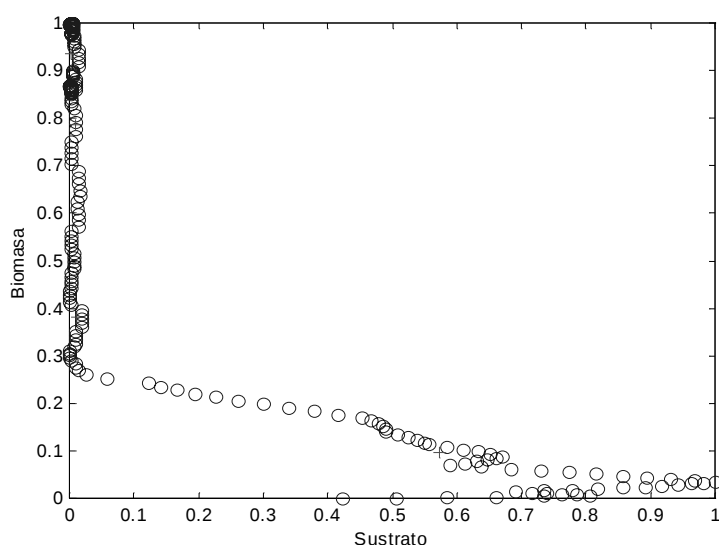


Figura 5: Espacio de estados normalizado para las variables S y X (círculos azules) y centros de las gaussianas de la RBFN (cruces rojas)

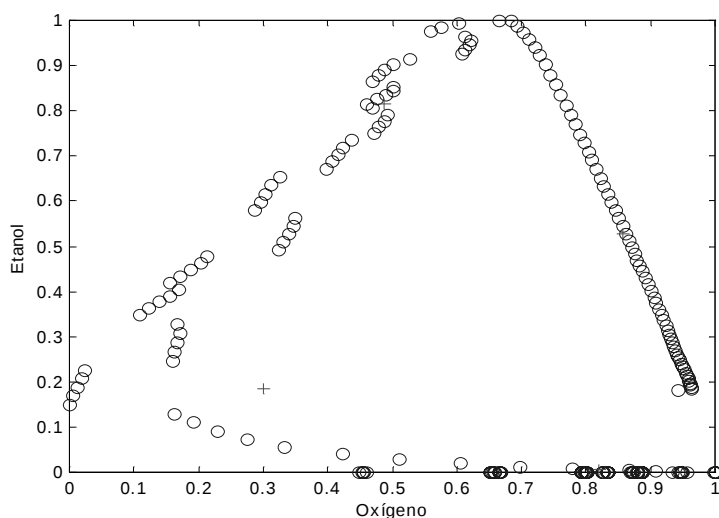


Figura 6: Espacio de estados normalizado para las variables O y E (círculos azules) y centros de las gaussianas de la RBFN (cruces rojas)

Como características concretas del entrenamiento de la red competitiva, decir que los datos se han normalizado previamente para solucionar posibles problemas derivados de las grandes diferencias de dimensiones de las distintas variables involucradas. El número de iteraciones ha sido de 4000 con un paso de aprendizaje de 0.1.

4.3. Equivalencia entre la RBFN y un sistema de inferencia difuso

La utilización de una red de base radial equivale a un sistema de inferencia difuso, si se tienen en cuenta los puntos siguientes:

- Cada una de las neuronas ocultas se corresponde con una regla difusa if-then.
- La salida de cada regla borrosa es una constante.
- Las funciones de pertenencia utilizadas en las reglas son gaussianas.
- El operador utilizado para computar el disparo de cada regla es la multiplicación.
- Tanto la RBFN como el sistema de inferencia difuso usan el mismo método para obtener las salidas globales (en este caso, una suma ponderada).

Todo esto es interesante para poder realizar una interpretación lingüística de la red, al contrario de lo que sucede normalmente con otro tipo de redes en las cuales los pesos no tienen ninguna interpretación física (P. Ascencio, 1998).

4.4. Modelos locales obtenidos mediante una RNEE lineal

Una vez estudiada la forma de trabajar con los modelos lineales locales, veamos cómo obtener cada uno de ellos a partir de los datos experimentales disponibles. En primer lugar vamos a identificar cada modelo utilizando una RNEE lineal (ver figura 7), que nos proporcionará una representación en el espacio de estados para cada uno de ellos. Las ecuaciones de la red lineal utilizada y la equivalencia con un modelo en el espacio de estados (con un término proporcional al error) es la siguiente:

$$\hat{x}(t+1) = W^h \cdot (W^r \hat{x}(t) + W^i u(t) + W^e e(t))$$

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

$$\hat{y}(t) = \hat{x}(t)$$

$$\hat{x}(t+1) = A\hat{x}(t) + Bu(t) + Ke(t)$$

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

$$\hat{y}(t) = \hat{x}(t)$$

$A = W^h \cdot W^r$ $B = W^h \cdot W^i$ $K = W^h \cdot W^e$

A continuación se expone el proceso seguido para obtener cada uno de esos modelos:

- 1) Primero se calcula el grado de pertenencia de cada dato a cada modelo local utilizando la red de base radial como se ha explicado en apartados anteriores.
- 2) Se forman tantos grupos de datos como modelos locales tengamos, agrupando los datos cuyo grado de pertenencia a un modelo sea mayor que a los restantes.
- 3) Con cada uno de los grupos se entrena una RNEE lineal, obteniendo así un modelo lineal local en el espacio de estados. En concreto, se han tomado 5 neuronas lineales en la capa oculta para todos los modelos locales. Para el entrenamiento, el número de iteraciones ha sido de 2.500 y se ha utilizado el método MOAM.

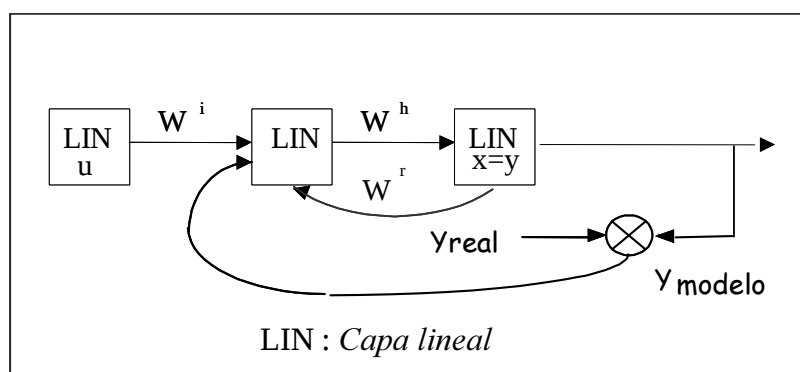


Figura 7: RNEE lineal para el caso en el que las salidas son igual a los estados

Con los datos de entrenamiento se colocan los centros de los modelos y se realiza su identificación mediante una RNEE lineal. Con los datos de validación se observa cómo se comporta el modelo global para datos no vistos con anterioridad, sin cambiar los centros ni los modelos locales. En la figura 8 se pueden ver las entradas con las cuales se han obtenido los datos de entrenamiento. En las figuras 9,10 y 11 se presentan los resultados obtenidos para 3 y 6 modelos locales siguiendo esta metodología, incluyendo el error cuadrático medio que se comete. Los datos se encuentran todos normalizados entre 0 y 1.

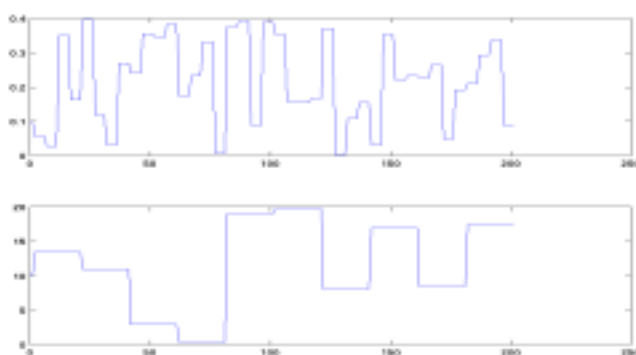


Figura 8: Entradas para el entrenamiento: F (arriba, en l/h), Se (abajo, en g/l)

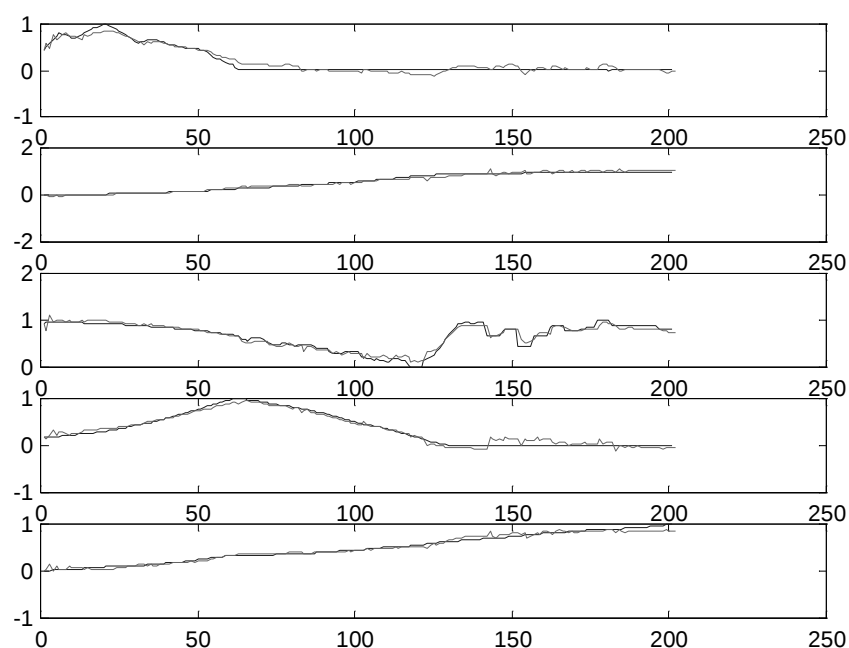


Figura 9: Salidas reales (azul) y del modelo formado por 3 modelos locales (rojo), para los datos de entrenamiento. (Error = 0.001752)

De la figura anterior se puede deducir que el comportamiento para los datos de entrenamiento es bastante bueno, pero para los datos de validación el error es mayor aunque más o menos se sigue la tendencia, como se observa en las figuras siguientes.

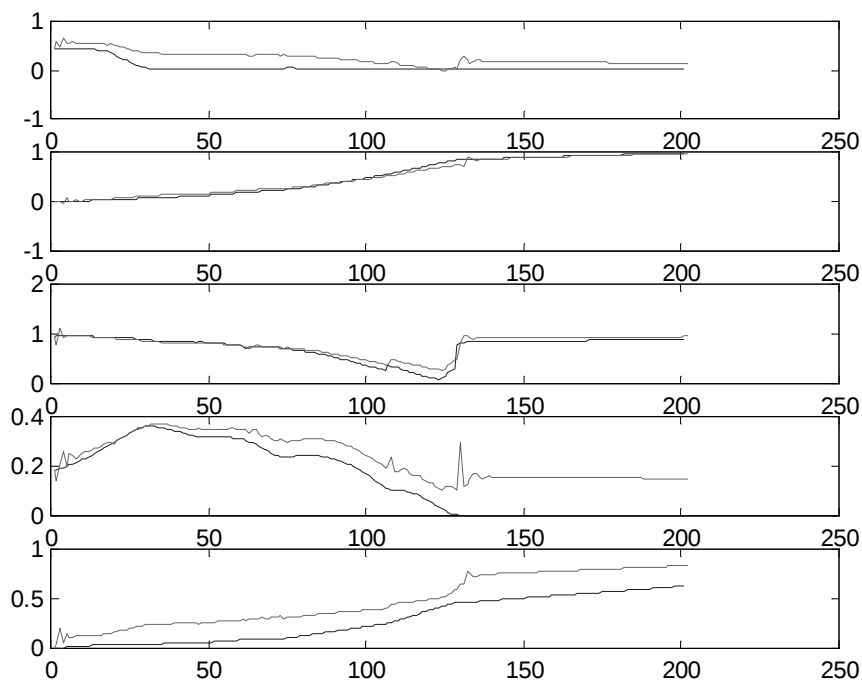


Figura 10: Salidas reales (azul) y del modelo formado por 3 modelos locales (rojo), para los datos de validación. (Error = 0.010015)

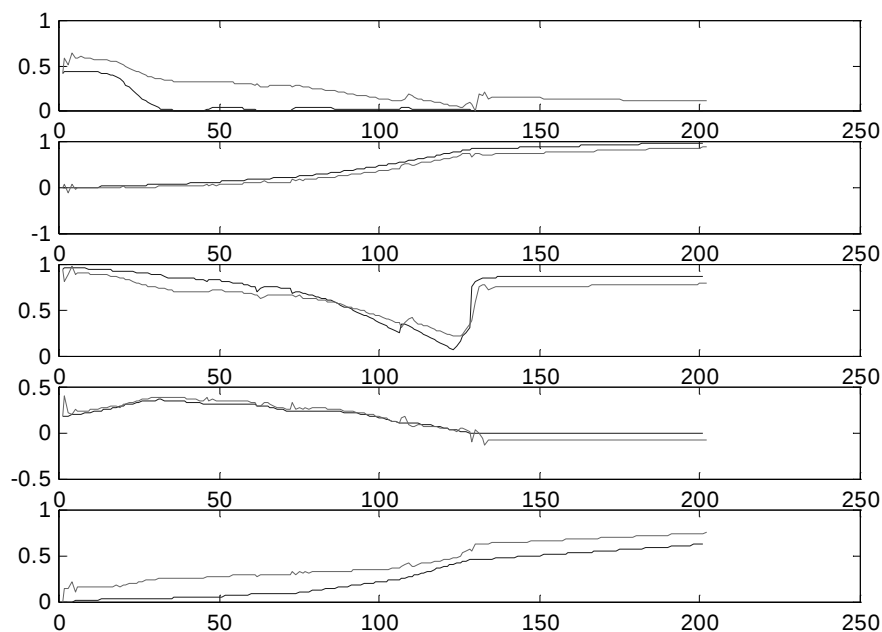


Figura 11: Salidas reales (azul) y del modelo formado por 6 modelos locales (rojo), para los datos de validación. (Error = 0.008473)

4.5. Modelos locales obtenidos linealizando a partir del modelo RNEE global

En el apartado 3 se desarrolló un modelo global para el fermentador utilizando la RNEE. Linealizando en torno a los puntos que hemos fijado como centros de los modelos se pueden obtener los modelos lineales locales, que se solaparán mediante las gaussianas para proporcionar la salida del sistema global.

El estudio se ha llevado a cabo utilizando desde 1 hasta 8 modelos locales, observándose que a mayor número de modelos el error proporcionado es menor. Como datos de entrenamiento para colocar los centros de las neuronas de la RBFN se han utilizado las entradas de la figura 8.

Para estos datos, se han comparado las salidas reales con las que proporciona nuestro modelo cuando se consideran 4 modelos locales (ver figura 12). Como se puede observar, el comportamiento del modelo es muy similar al del sistema, y lo mismo ocurre cuando consideramos distinto número de modelos locales, debido a que son datos de entrenamiento. En las figuras 13-15 se observa lo que ocurre para datos de validación, es decir, datos que no se han utilizado en el desarrollo del modelo.

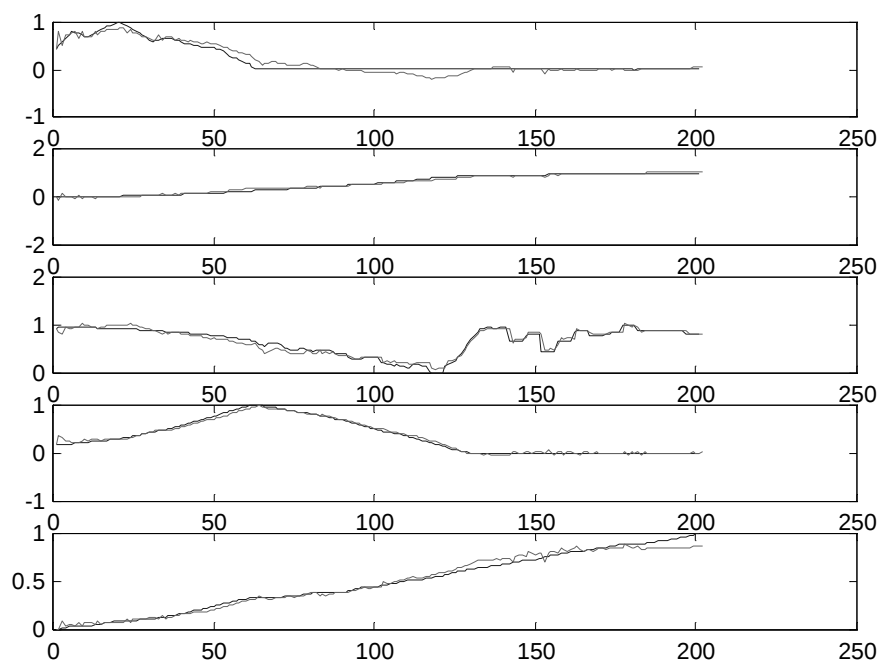


Figura 12: Salidas para los datos de entrenamiento (reales y modelo formado por 4 modelos lineales locales)

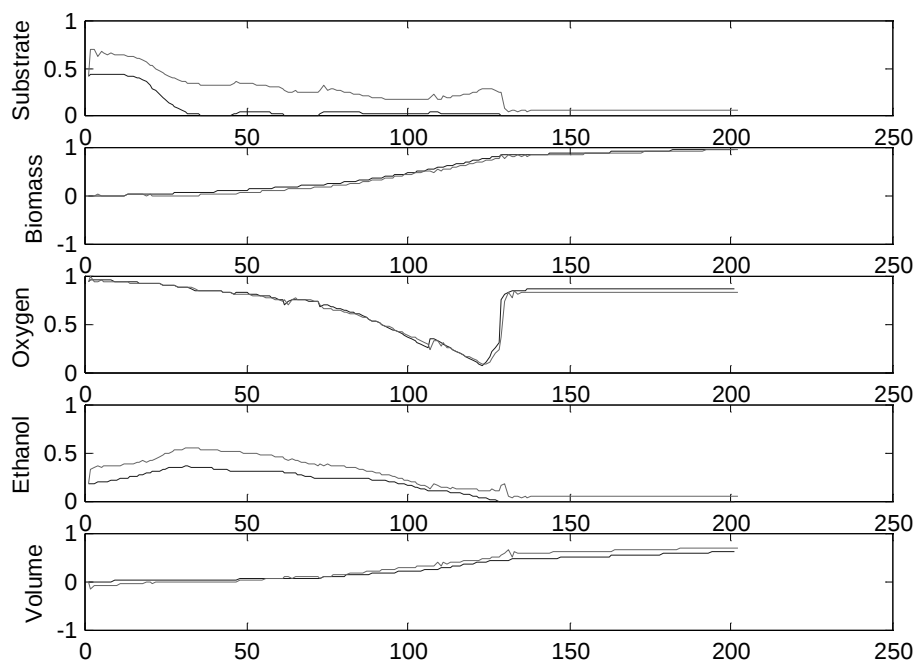


Figura 13: Salidas reales (azul) y del modelo formado por 2 modelos locales (rojo), para los datos de validación. (Error = 0.005993)

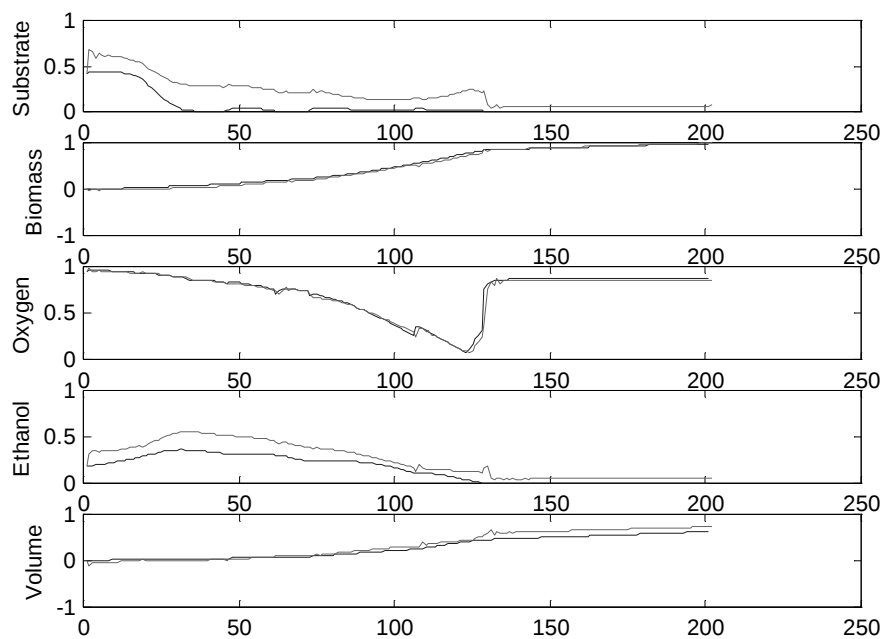


Figura 14: Salidas reales (azul) y del modelo formado por 4 modelos locales (rojo), para los datos de validación. (Error = 0.004723)

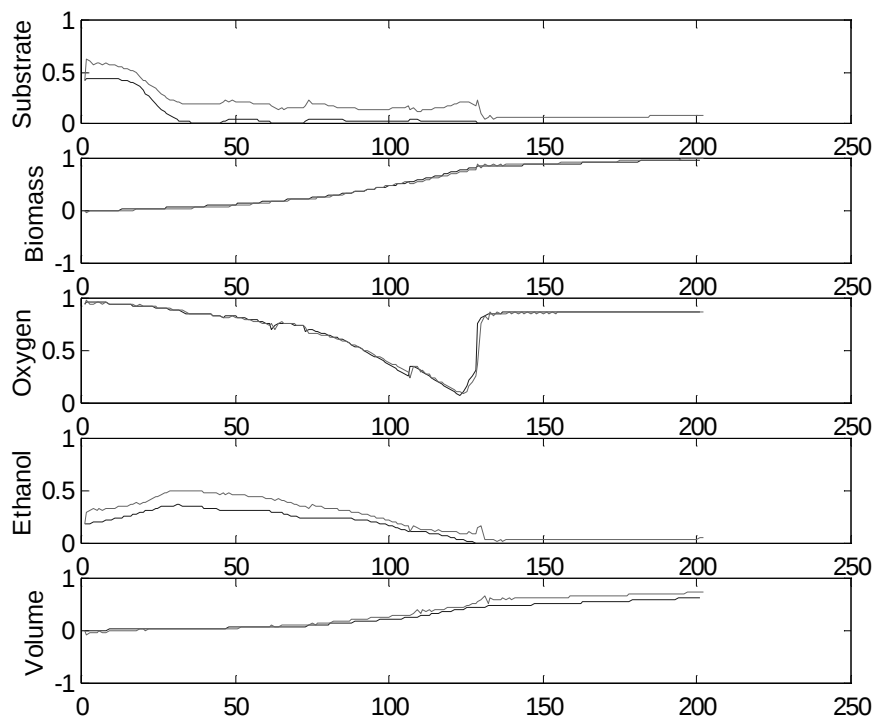


Figura 15: Salidas reales (azul) y del modelo formado por 6 modelos locales (rojo), para los datos de validación. (Error = 0.003039)

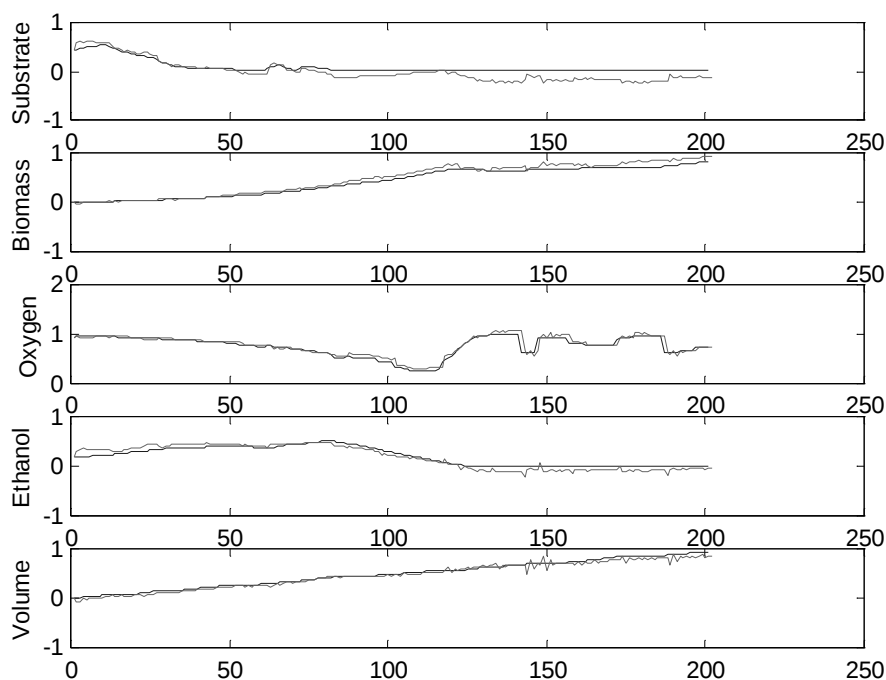


Figura 16: Salidas reales (azul) y del modelo formado por 6 modelos locales (rojo), para datos de validación distintos a los de las figuras anteriores. (Error = 0.003123)

De los resultados anteriores se puede deducir que a mayor número de modelos locales, el desempeño del modelo global es mejor (el error cuadrático medio para los datos de validación es menor). No obstante, existe un offset que habría que corregir en cada periodo de muestreo con el error anterior. Si se sigue aumentando el número de modelos locales, el comportamiento no mejora e incluso es peor para los datos de validación. Esto es debido a que los modelos locales ya son suficientes para cubrir todo el espacio de estados. Aunque en todos los casos se toman como condiciones iniciales los valores reales en $t = 0$, el comportamiento es también bueno si se parte de otros valores ya que el error se corrige muy rápidamente.

Si comparamos las figuras 11 y 15, ambas considerando 6 modelos locales, podemos observar que en el caso de los modelos locales obtenidos a partir del modelo RNEE global el resultado es mejor (figura 15) que cuando se ha identificado cada modelo con una RNEE lineal (figura 11). Esto puede ser debido a que en el primer caso los modelos locales se obtienen a partir de un modelo global que ya ha sido validado, mientras que en el otro caso los modelos parciales se han obtenido únicamente utilizando datos de entrenamiento, sin ningún tipo de validación cruzada simultánea.

Por último, en la figura 17 se presentan los resultados que se obtienen si no se utiliza la RBFN para ponderar la validez de los distintos modelos locales, es decir, si en cada punto todos los modelos locales tienen la misma validez. Comparando con la figura 15 se observa obviamente que los resultados son peores debido a que cada modelo debe tener su zona de aplicación y no considerar a todos por igual.

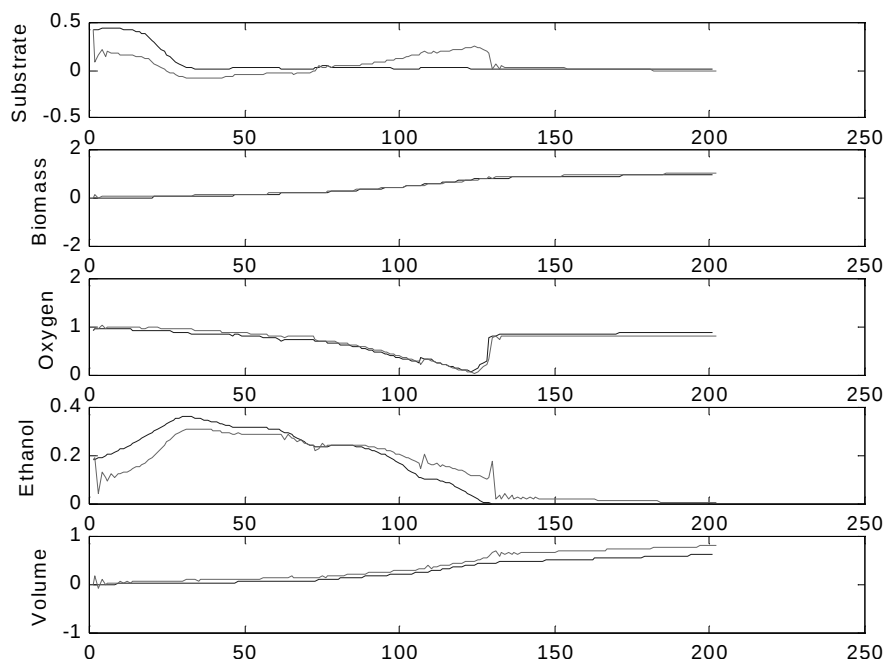


Figura 17: Salidas reales (azul) y del modelo formado por 6 modelos locales (rojo), para datos de validación y cuando cada modelo local contribuye 1/6 a la salida total. (Error = 0.003560)

4.6. Determinación de las matrices K por ubicación de polos

Hasta este momento, las matrices K que aparecen en el término de corrección proporcional al error, han sido determinadas bien a partir de una RNEE lineal o bien a partir de la linealización del modelo RNEE global. En este punto proponemos su determinación utilizando la técnica de colocación de polos, de tal manera que los polos de los observadores locales estén alejados de los del sistema local y del límite de estabilidad.

Considerando 6 modelos lineales locales, tendremos 6 modelos en el espacio de estados con sus matrices A , B y K . Mediante la técnica de colocación de polos se obtienen matrices K tales que los autovalores de $A-K$ (matriz del observador) están alejados del círculo unidad, que es el límite de la estabilidad para sistemas discretos como el que estamos simulando, con periodo de muestreo de 0.1 horas. En cuanto a las matrices A y B se han seguido utilizando las mismas que en el apartado 4.5, es decir, linealizando el modelo global RNEE.

En la figura siguiente se muestran los resultados cuando los polos del observador se colocan en 0.1, 0.1, -0.1, -0.1 y 0.2, todos ellos reales. El resultado obtenido es peor que con las técnicas anteriores para el mismo número de modelos locales.

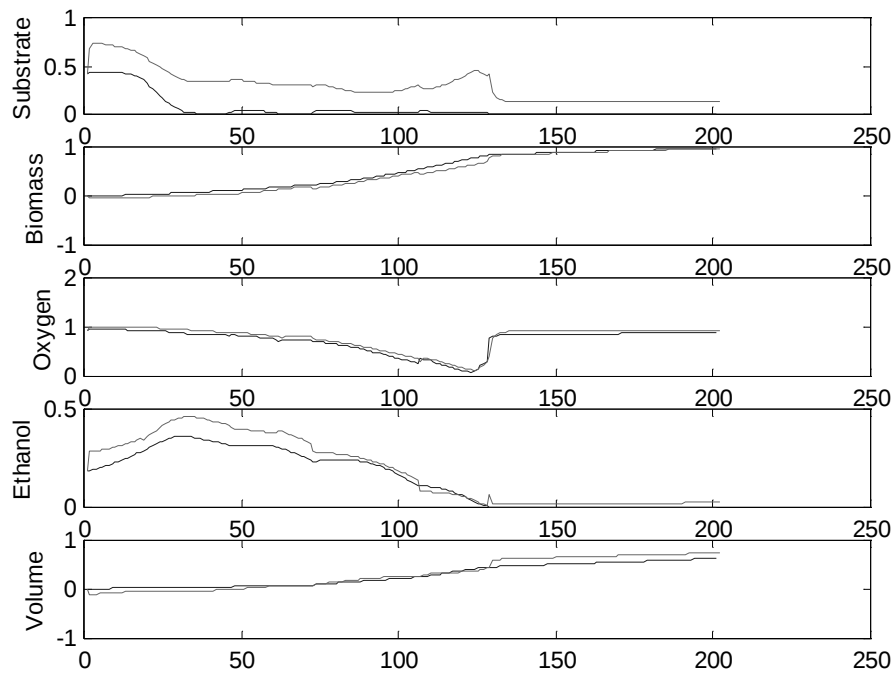


Figura 18: Salidas reales (azul) y del modelo formado por 6 modelos locales (rojo), para datos de validación y con colocación de polos para el cálculo de las K. (Error = 0.007835)

Como comparativa a la figura anterior, si colocamos los polos en la zona próxima a la inestabilidad, por ejemplo en -0.5 , 0.5 , $0.7+0.7i$, $0.7-0.7i$ y -0.3 , observamos que el comportamiento es aún peor, casi oscilante en las variables S y X, tanto para los datos de entrenamiento como para los de validación (figura 19).

5. CONCLUSIONES Y TRABAJO FUTURO

En este trabajo se han utilizado tres técnicas para desarrollar un modelo formado por modelos locales para un fermentador semicontinuo. En todos los casos se trata de modelos representados en el espacio de estados y solapados mediante una red de base radial. Debido al término de corrección $K(e(t))$, todos los modelos son de predicción a un paso. Esto es así porque dichos modelos utilizan el valor real de la salida en cada instante de tiempo t para compararlo con el valor estimado, calcular el error $e(t)$, y con todo ello calcular la salida en $t+1$.

Sería conveniente realizar todo este estudio para el caso en el que la matriz C de salida fuera distinta de la identidad, es decir, considerando que los estados S y X no son medibles en línea y deben ser obtenidos a partir de medidas de O, E y V.

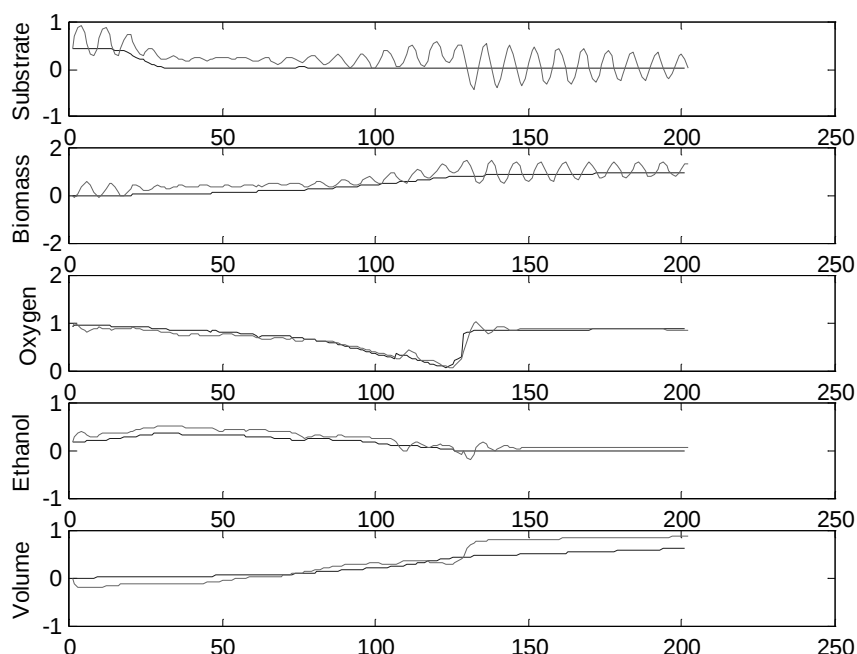


Figura 19: Salidas reales (azul) y del modelo formado por 6 modelos locales (rojo), para datos de validación y utilizando colocación de polos próximos a la inestabilidad para el cálculo de las matrices K.

6. REFERENCIAS

- J. M. Zamarreño, P. Vega, L. D. García, M. Francisco, State space neural network for modelling, prediction and control. *Control Engineering Practice*, Vol. 8 , 2000. pp. 1063-1075.
- M. Francisco, P. Vega, Modelado de un Fermentador mediante Red neuronal en el espacio de estados. *XX Jornadas de Automática (Salamanca)*, 1999.
- J. M. Zamarreño, Identificación y control predictivo basado en modelos mediante red neuronal en el espacio de estados, *Tesis Doctoral. (Universidad de Valladolid)*, 1996.
- F.R. Oliveira, Monitorização e controlo de fermentadores. Aplicação ao fermento de padeiro. *Tesis Doctoral. (Universidad de Oporto)*, 1997.
- P. Ascencio, Sistemas neuro-difusos en modelación y control de procesos no lineales. *Memoria de título de Ingeniero Civil Electrónico. (Universidad de Concepción)*, 1998.
- A. Banerjee, Y. Arkun, B. Ogunnaike, R. Pearson, Estimation of nonlinear systems using linear multiple models. *AIChE Journal*, Vol. 43, n°5, 1997. pp. 1204-1225.

- J. Moreno, A. Vargas, Nonlinear observer design and observability of an aerobic bioreactor using respirometry, *European Control Conference*, 1999.
- M. Maher, G. Roux, B. Dahhou, A method for estimating the state variables and parameters of fermentation systems, *J. Chem. Tech. Biohecnol.*, Vol. 63, 1995, pp. 153-159.