
UNIVERSIDAD POLITECNICA DE VALENCIA

Departamento de Ingeniería de Sistemas y Automática



Tesis Doctoral

**Técnicas de agrupamiento
para identificación y control
por modelos locales**

Autor: D. José Luis Díez Ruano

Directores: D. José Luis Navarro Herrero
D. Antonio Sala Piqueras

Valencia, julio de 2003

TÉCNICAS DE AGRUPAMIENTO
PARA IDENTIFICACIÓN Y CONTROL
POR MODELOS LOCALES.



José Luis Díez Ruano

Depto. de Ingeniería de Sistemas y Automática,
Universidad Politécnica de Valencia.

Apdo. 22012, E-46071 Valencia, ESPAÑA

E-mail: jldiez@isa.upv.es

*A mis padres,
porque el espacio que nos une es infinito.*

*A Violeta,
porque nuestro espacio es circular.*

*A Mario,
porque no hay espacio.*

El límite es el verdadero protagonista del espacio.
Eduardo Chillida.

Agradecimientos

Lo que con mucho trabajo se adquiere, más se ama.
Aristóteles.

La tarea del profesor universitario exige compaginar tres campos: docencia, investigación y gestión. Esto es algo que aunque conocido no resulta fácil, especialmente si se desea obtener resultados dignos en todas las tareas. Además, al exigir resultados inmediatos, la docencia y la gestión acaban acaparando gran parte de los recursos relegando la investigación y dilatando sus resultados en el tiempo. En este marco, el apoyo y colaboración se transforman de convenientes a imprescindibles y, aunque el mejor agradecimiento no se escribe, no puedo dejar pasar esta oportunidad para mostrar mi gratitud a algunas personas.

En primer lugar deseo expresar mi agradecimiento a mis directores de Tesis, los doctores D. José Luis Navarro Herrero y D. Antonio Sala Piqueras, por su apoyo, tiempo y dedicación. Igualmente, deseo agradecerles sus múltiples sugerencias durante la supervisión del documento definitivo. La codirección ha resultado una experiencia muy enriquecedora e interesante, pues no solo de las conversaciones con los directores, sino también de las discusiones habidas entre ellos, han surgido ideas que han ido encaminando este trabajo de investigación.

Quiero agradecer también la ayuda aportada por el Grupo de Investigación de Control de Sistemas Complejos, al cual pertenezco, ya que gracias a los proyectos de investigación en los que participamos, he tenido ocasión de implementar y probar algunas de las aportaciones de la Tesis. Asimismo, agradecer al Departamento de Ingeniería de Sistemas y Automática en general y a los miembros Grupo de Control Predictivo en particular, su desinteresada colaboración en el desarrollo de algunos experimentos.

Deseo expresar un especial agradecimiento a los catedráticos D. Pedro Albertos Pérez y D. Miguel A. Martínez Iranzo por su apoyo y consejos que, dada su buena perspectiva, han sido clave para el desarrollo de mi trayectoria en la Universidad.

Además, quiero dar las gracias por las ideas y sugerencias aportadas a todos aquellos que, en algún momento, han interactuado conmigo en el desarrollo de este trabajo.

No puedo terminar sin agradecer a mis padres su esfuerzo, el cual hizo posible que comenzase mis estudios universitarios y que de ese modo se iniciase el camino que aquí culmina. Del mismo modo no olvido su preocupación en todo el recorrido.

Finalmente deseo dar las gracias a Violeta por su comprensión, paciencia y apoyo continuo a lo largo del desarrollo de esta investigación. Especialmente quiero pedir disculpas tanto a ella como a Mario por las horas robadas, por todas las ausencias, porque sin ellas no habría podido finalizar este trabajo.

Resumen

Una opción interesante para el modelado de sistemas no lineales son los modelos borrosos, sin olvidar que una de las metas del modelado borroso es la interpretabilidad de los modelos. Además, si se pretenden usar los modelos identificados para una posterior aplicación a control, será necesario tener en cuenta algunas consideraciones adicionales que podrían modificar los criterios empleados en identificación para predicción. En esta Tesis, tras resaltar algunos conceptos poco claros relacionados con los criterios de modelado local encontrados en la bibliografía, se proponen las características deseables de las técnicas de modelado e identificación borrosa cuando el propósito es el control por modelos locales.

Tras estudiar el comportamiento de las funciones objetivo de los algoritmos de agrupamiento borroso adecuados en el modelado para control, se presenta un nuevo índice que incorpora algunas características deseables relacionadas con la convexidad y suavidad de las clases finales identificadas, con ventajas respecto a otras metodologías como Gustaffson-Kessel.

A continuación se define una familia de algoritmos de agrupamiento que minimizan la función objetivo previamente definida. Estos algoritmos llevan a una identificación más adecuada de modelos locales, aproximando los modelos locales el modelo linealizado de la planta en sus zonas de validez y, adicionalmente, con características deseables respecto a la interpretabilidad de las funciones de pertenencia (convexidad y suavidad). El algoritmo proporciona simultáneamente modelos locales lineales y agrupamiento de los datos de entrada, siendo especialmente aplicable para identificación de modelos borrosos Takagi-Sugeno y descomposición de sistemas no lineales en modelos locales lineales para control.

Resum

Una opció interessant per al modelatge de sistemes no lineals són els models borrosos, sense oblidar que una de les metes del modelatge borros és la interpretabilitat dels models. A més a més, si es pretenen emprar els models identificats per a una posterior aplicació a control, serà necessari tindre en compte algunes consideracions addicionals que podien modificar els criteris emprats en identificació per a predicció. En aquesta Tesi, després de ressaltar alguns conceptes poc clars relacionats amb els criteris de modelatge local trobats en la bibliografia, es proposen les característiques desitjables de les tècniques de modelatge i identificació borrosa quan el propòsit és el control per a models locals.

Després d'estudiar el comportament de les funcions objectiu dels algorismes borrosos d'agrupació adequats en el modelatge per a control, es presenta un nou índex que incorpora algunes característiques desitjables relacionades amb la convexitat i suavitat de les classes finals identificades, amb avantatges respecte a altres metodologies com Gustaffson-Kessel.

A continuació es defineix una família d'algorismes d'agrupació que minimitzen la funció objectiu prèviament definida. Aquests algorismes porten a una identificació més adequada de models locals, aproximant els models locals el model linealitzat de la planta en les seues zones de validesa i, addicionalment, amb característiques desitjables respecte a la interpretabilitat de les funcions de pertinença (convexitat i suavitat). L'algorisme proporciona simultàniament models locals lineals i agrupació de les dades d'entrada, sent especialment aplicable per a identificació de models borrosos Takagi-Sugeno i descomposició de sistemes no lineals en models locals lineals per a control.

Abstract

A valuable option in complex systems modelling is the use of fuzzy models, but fuzzy modelling should have interpretability of the obtained models as a fundamental goal. Additionally, some considerations must be taken into account when subsequent use of this models is local model control, provided that it might influence the criteria used in identification for prediction. In this Thesis, after elucidating some unclear concepts found in literature regarding local modelling criteria, desirable characteristics of fuzzy modelling and identification techniques for control purposes are proposed.

Behaviour of objective functions of local-model-control-oriented fuzzy clustering algorithms is studied and a new index is presented so that it incorporates some desirable characteristics regarding convexity and smoothness of the final identified clusters, with advantages regarding other methodologies such as Gustaffson-Kessel.

Subsequently, the definition of a family of clustering algorithms for previously defined objective function minimization is done. They lead to a better identification of local models approximating the linearized plant model on their validity zones and with some additional desirable characteristics regarding membership function interpretability (convexity and smoothness). The algorithm simultaneously provided local linear models and input clustering, being specially suitable for Takagi-Sugeno fuzzy models identification and local linear models decomposition of nonlinear systems for control.

Índice general

I	Introducción	1
1.	Introducción	3
1.1.	Motivación	3
1.2.	Objetivos de la Tesis	4
1.3.	Estructura y contenidos de la Tesis	5
II	Estado del arte	9
2.	Modelado e identificación de sistemas no lineales	11
2.1.	Introducción	11
2.2.	Diferentes enfoques del modelado	12
2.3.	Modelado por modelos locales	13
2.4.	Modelado borroso	14
2.5.	Identificación borrosa	15
2.6.	Conclusiones	17
3.	Modelado basado en modelos locales	19
3.1.	Introducción	19
3.2.	Modelos de sistemas no lineales	20
3.3.	Linealización en torno a puntos de operación	21
3.4.	Linealización basada en la velocidad	23
3.5.	Conclusiones	24
4.	Control con modelos locales	25
4.1.	Introducción	25
4.2.	Control por planificación de ganancia	26
4.2.1.	Controladores basados en modelos LPV del sistema en puntos de operación	27
4.2.2.	Controladores basados en modelos LPV del sistema linealizados por velocidad	29
4.3.	Análisis de estabilidad mediante LMIs	30
4.3.1.	Proceso iterativo para diseño de controladores	31
4.3.2.	Diseño directo por realimentación del estado	32
4.3.3.	Estabilidad mediante LDIs	33
4.4.	Conclusiones	34

5. Modelado borroso	35
5.1. Modelos borrosos basados en reglas	35
5.1.1. Modelos lingüísticos (Mamdani)	36
5.1.2. Modelos de Takagi-Sugeno	37
5.2. Inferencia de los modelos TS	37
5.3. Modelos borrosos de sistemas dinámicos	39
5.3.1. Representación con preprocesador	39
5.3.2. Representación en el espacio de estados	40
5.4. Identificación de modelos borrosos basados en reglas	40
5.4.1. Elementos a identificar	41
5.4.2. Métodos de aprendizaje	41
5.5. Conclusiones	43
6. Técnicas de agrupamiento para identificación	45
6.1. Técnicas de agrupamiento	45
6.2. Particiones clásicas y borrosas	46
6.3. Algoritmo de agrupamiento borroso fuzzy c-means	49
6.3.1. Función objetivo c-means	49
6.3.2. Algoritmo fuzzy c-means (FCM)	50
6.3.3. Normas	50
6.3.4. Extensiones al algoritmo FCM	52
6.4. Algoritmo de Gustafson-Kessel (GK)	52
6.5. Algoritmo fuzzy c-regression models (FCRM)	54
6.6. Algoritmos con prototipos mixtos	55
6.6.1. Algoritmo fuzzy c-elliptotypes (FCE)	55
6.6.2. Adaptive fuzzy c-regression models (AFCR)	55
6.7. Proceso de identificación de modelos borrosos	56
6.7.1. Diseño de experimentos	57
6.7.2. Selección de la estructura	57
6.7.3. Agrupación de los datos	59
6.7.4. Selección del número de clases	60
6.7.5. Determinación de un modelo borroso inicial	60
6.7.6. Simplificación del modelo inicial	63
6.7.7. Validación del modelo	63
6.8. Conclusiones	64
III Propuesta de técnicas para identificación y control	67
7. Evaluación del agrupamiento en identificación	69
7.1. Método sistematizado de identificación	69
7.2. Criterios de modelado	74
7.2.1. Modelos globales de sistemas no lineales	75
7.2.2. Modelos locales de sistemas no lineales	76
7.3. Agrupamiento borroso en identificación	78
7.3.1. Algoritmos con distancia adaptativa	79

7.3.2.	Algoritmos con prototipos lineales	80
7.3.3.	Algoritmos con prototipos mixtos	80
7.4.	Índices para la validación de los modelos locales	80
7.4.1.	Índice de compacidad	81
7.4.2.	Índice de solapamiento	81
7.4.3.	Índice de funcionamiento local	82
7.5.	Análisis de algoritmos de agrupamiento	82
7.6.	Conclusiones	91
8.	Propuesta de una técnica mejorada de agrupamiento	95
8.1.	Un índice que favorece la convexidad	95
8.2.	Minimización del nuevo índice de coste	97
8.3.	Algoritmo de agrupamiento para la mejora de la convexidad	98
8.4.	Efecto de los nuevos términos en el algoritmo AFCRC	100
8.5.	Parámetros de ajuste del algoritmo	104
8.6.	Ejemplos de aplicación	106
8.6.1.	Identificación de funciones no-lineales unidimensionales	106
8.6.2.	Identificación de funciones de varias variables	108
8.7.	Conclusiones	113
9.	Integración de técnicas de agrupamiento y control	115
9.1.	Introducción	115
9.2.	Control por planificación de ganancia	116
9.2.1.	Modelos LPV identificados con AFCRC	116
9.2.2.	Aplicabilidad de controladores basados en modelos LPV del sistema en puntos de operación	117
9.2.3.	Aplicabilidad de controladores basados en modelos LPV del sistema linealizados por velocidad	126
9.3.	Control por LMIs	133
9.4.	Conclusiones	135
10.	Ejemplos de aplicación de los algoritmos	137
10.1.	Modelado de un sensor de biomasa	137
10.1.1.	Identificación de modelos locales	137
10.1.2.	Validez de los modelos locales	138
10.2.	Identificación y control de un sistema térmico	141
10.2.1.	Descripción del proceso	141
10.2.2.	Identificación de un modelo no lineal	143
10.2.3.	Diseño de controladores lineales	152
10.2.4.	Control por planificación de ganancia: interpolación de controladores	153
10.2.5.	Rendimiento	154
10.3.	Conclusiones	155

IV Conclusiones	157
11. Conclusiones y cuestiones abiertas	159
11.1. Conclusiones de la Tesis	159
11.2. Cuestiones abiertas	163
V Bibliografía	165
VI Anexos	177
A. Modelos locales	179
A.1. Introducción	179
A.2. Modelos de sistemas no lineales	180
A.3. Linealización en torno a puntos de equilibrio	181
A.4. Linealizaciones en puntos fuera del equilibrio	184
B. Planificación de ganancia	189
B.1. Introducción	189
B.2. Control por planificación de ganancia	190
B.3. Planificación de ganancia desde el diseño de controladores con métodos lineales.	191
C. Algoritmos de agrupamiento	205
C.1. Algoritmos de agrupamiento clásico	205
C.2. Algoritmo de agrupamiento borroso fuzzy c-means	211
C.3. Algoritmos con norma adaptativa	215
C.4. Algoritmos con prototipos lineales	219
C.5. Algoritmos con prototipos mixtos	221
C.6. Algoritmos con prototipos no lineales	223
C.7. Algoritmos robustos	226
C.8. Determinación del número de clases	228
D. Identificación de modelos borrosos	233
D.1. Proceso de identificación de modelos borrosos	233
D.2. Selección de la estructura	235
D.3. Agrupación de los datos	239
D.4. Determinación de un modelo borroso inicial	243
D.5. Simplificación del modelo borroso obtenido	248
D.6. Validación del modelo	249
D.7. Software para la identificación sistematizada de modelos borrosos	250

Parte I
Introducción

Capítulo 1

Introducción

*Todo lo que puede ser mal interpretado,
es mal interpretado siempre.*
Dale Carnegie.

Este capítulo de introducción presenta la motivación investigadora que ha llevado a la realización de esta Tesis, presentando los objetivos planteados y la estructura del trabajo que aquí comienza.

1.1. Motivación

En el campo de la ingeniería de control, el modelado y la identificación son pasos previos fundamentales en el diseño de sistemas de control, supervisión y detección de fallos. Sin embargo, existen muchos sistemas para los cuales no es posible aplicar las técnicas de modelado tradicionales debido a la falta de un conocimiento preciso y formal del sistema, a un comportamiento altamente no lineal del sistema, al alto grado de incertidumbre en los datos, a características cambiantes, etc. Los sistemas flexibles de fabricación, la ingeniería aeroespacial, la ingeniería química y bioquímica, así como las ciencias ecológicas, sociales y financieras, son ejemplos de este tipo de sistemas, a los que llamaremos *sistemas complejos*. De entre todos ellos, esta Tesis se centrará en el estudio de *sistemas dinámicos no lineales*.

Un enfoque muy interesante para el modelado de sistemas no lineales es el que recurre a su descomposición en un conjunto más o menos grande de submodelos con un rango de validez limitado, a los que podemos denominar *modelos locales*. Cada uno de estos submodelos puede tener una estructura distinta (caja negra, caja blanca, caja gris) y el modelo global del sistema se obtendrá a través de la integración de los modelos locales.

Si para la integración de estos modelos locales se emplean técnicas de *inteligencia artificial*, se evita una de las desventajas más comunes entre las técnicas de modelado tradicionales y que consiste en que no pueden usar información adicional como el conocimiento y la experiencia de los ingenieros y operadores, que es a menudo cualitativo e impreciso, ni en el modelado ni en la validación del modelo. Las técnicas de inteligencia artificial utilizan representaciones alternativas como, por ejemplo, el lenguaje

natural, reglas, redes semánticas o modelos cualitativos, y poseen métodos formales para incorporar información adicional.

El *modelado borroso* es un ejemplo típico de técnicas inteligentes, ya que usan el conocimiento humano y procesos deductivos. La característica que se debería explotar para desarrollar un controlador desde un modelo borroso y, en general, desde cualquier técnica de identificación basada en Inteligencia Artificial, no es su capacidad de predicción de la salida del sistema real gracias a su característica de aproximador universal, es decir, que pueden emular cualquier tipo de relación continua entre un grupo de variables de entrada y de salida. Lo que realmente diferencia a los modelos borrosos de otras técnicas clásicas de modelado es su capacidad de incorporar información cualitativa e imprecisa que los ingenieros u operarios tienen sobre el sistema y, además, que la representación final que se obtiene posee un sentido físico comprensible para el usuario del modelo.

En el caso de emplear estos modelos borrosos para el *diseño de controladores*, el usuario es el Ingeniero de Control y la interpretabilidad se mide en el sentido de que los modelos identificados sean realmente lo que parecen representar. Por ejemplo, si los consecuentes de las reglas han sido definidos con estructura de modelos lineales se puede interpretar que corresponden a linealizaciones del proceso. Si esta correspondencia no existe, los controladores (para linealizaciones) diseñados por el Ingeniero no se comportarán como es de esperar. En ciertos casos puede interesar emplear metodologías de identificación de sistemas con ventajas de fácil entrenamiento, elevada precisión y técnicas efectivas de diseño de controladores, pero entonces hay que tener en cuenta que los conceptos “controladores locales”, “punto de operación” o “linealización” pueden perder su sentido original.

Esta Tesis se centrará en la identificación de modelos locales lineales de sistemas no lineales e integrados con la arquitectura de los modelos borrosos, procurando la metodología para que los modelos lineales correspondan a linealizaciones del sistema a identificar. De este modo se podrán emplear para el control del sistema no lineal los métodos de diseño de controladores que se basen en linealizaciones de un proceso.

1.2. Objetivos de la Tesis

El objetivo principal de la Tesis será discutir sobre la idoneidad, en lo que respecta a la interpretabilidad para el usuario y posterior aplicación al diseño de controladores, de los diferentes criterios de modelado que se emplean para el modelado de sistemas no lineales mediante técnicas borrosas (especialmente las de agrupamiento). Las ideas de *modelado para control* han aparecido recientemente en el campo de los sistemas lineales y deben ser extendidas a las técnicas de identificación de modelos borrosos que se empleen posteriormente para control, pues el criterio de modelado puede ser distinto al de los modelos borrosos para predicción. A lo largo de este trabajo se demostrará que esto es así y se propondrán alternativas para la identificación de modelos borrosos de sistemas en las que la interpretabilidad es tenida en cuenta.

Este objetivo general se concreta en:

- Revisar el estado actual de las teorías involucradas en el proceso de diseño de

modelos borrosos para su uso en control,

- Determinar un método sistematizado de identificación basado en técnicas de agrupamiento,
- Analizar (desde un punto de vista tanto teórico como práctico) la idoneidad, desde el punto de vista del uso de los modelos para control, de las funciones de coste de los algoritmos de identificación,
- Determinar el algoritmo de agrupamiento más apropiado para la determinación de modelos locales,
- Proponer soluciones alternativas para los problemas que se encuentren en el proceso de identificación,
- Analizar la aplicabilidad de las técnicas de control para modelos locales a los modelos que se identifiquen
- Aplicar las soluciones que se propongan a diferentes ejemplos de identificación y control.

1.3. Estructura y contenidos de la Tesis

Tras la parte de **Introducción** que se desarrolla en este capítulo, para desarrollar los objetivos que se han planteado en la Tesis, se ha dividido este trabajo en tres partes adicionales que llevan a un total de 11 capítulos. Una primera en la que se revisa el **Estado del Arte** actual de los diferentes campos en los que se enmarca el desarrollo de las técnicas de trabajo propuestas (esta revisión se amplía en una serie de monografías incluidas en los Anexos). Posteriormente se abarca el cuerpo de la Tesis, consistente en una **Propuesta de Técnicas para Identificación y Control** de sistemas borrosos mediante técnicas de agrupamiento. El trabajo finaliza con una parte de **Conclusiones**. Cada una de estas partes se estructura en una serie de capítulos cuyo contenido se comentará brevemente a continuación.

Parte II. Estado del Arte.

Capítulo 2. Modelado e identificación de sistemas no lineales. En este capítulo de carácter introductorio se da una revisión general a los diferentes enfoques que se pueden encontrar para el modelado de sistemas no lineales, destacando las ventajas de los enfoques por modelos locales y los métodos denominados inteligentes.

Para ello, tras una introducción genérica al modelado de sistemas, se comentará en qué consiste el modelado por modelos locales así como, de entre los métodos inteligentes, el fundamentado en la lógica borrosa y los conjuntos borrosos: los modelos borrosos basados en reglas. Dichos modelos permiten combinar las buenas características de las técnicas inteligentes con el enfoque de los modelos locales. Seguidamente se planterán las formas en las que se puede llevar cabo la identificación borrosa para el caso de los sistemas no lineales, para finalizar con una sección de conclusiones.

Capítulo 3. Modelado basado en modelos locales. Este capítulo planteará diferentes formas de realizar particiones de modelos locales de sistemas no lineales, todas ellas mediante linealización (desarrollo en serie) en el entorno de puntos de operación empleando el Jacobiano de la planta.

Aunque el enfoque más habitual (clásico) de partición es aquel en el que los puntos de operación se corresponden con uno o varios puntos de equilibrio, sus limitaciones llevan a que este capítulo se centre en revisar marcos alternativos con técnicas que permiten la linealización en torno a puntos que no son de equilibrio, como la linealización en torno a puntos de operación o la linealización basada en la velocidad. De este modo se proporciona a los modelos borrosos identificados tal y como se perfila en el capítulo 2 y se describirá a lo largo de la Tesis, un marco teórico apropiado para trabajar en el entorno de cualquier punto (de equilibrio o no) y ser empleados para cualquier análisis o diseño.

Capítulo 4. Control con modelos locales. En este capítulo se verán las técnicas de control aplicables cuando se dispone de modelos locales de sistemas no lineales como los descritos en el capítulo 3. Estas técnicas son las que, basándose en diseño de controladores lineales para los modelos locales identificados, consiguen obtener controladores para el conjunto del sistema no lineal. Por ello se revisará el control por planificación de ganancia y los métodos basados en LMIs.

Capítulo 5. Modelado borroso. En este capítulo se hace un repaso a los diferentes tipos de modelos borrosos basados en reglas con los que podemos trabajar a la hora de modelar un sistema, resaltando sus características de interpretabilidad y la adecuación de los modelos de Takagi-Sugeno (TS) para integrar modelos locales como los vistos en el capítulo 3.

Capítulo 6. Técnicas de agrupamiento para identificación. Aquí es donde se revisan, de entre los múltiples algoritmos de agrupamiento borroso existentes, aquellos con capacidades de detección de subespacios lineales dada su idoneidad para la identificación de modelos borrosos TS.

También se hace una exposición estructurada y completa de los distintos pasos a seguir para la identificación del modelo de un sistema no lineal mediante modelos borrosos, identificando su estructura y parámetros con las técnicas de agrupamiento.

Parte III. Propuesta de Técnicas para Identificación y Control.

Capítulo 7. Evaluación de las técnicas de agrupamiento en identificación.

Una vez descritos en el capítulo anterior todos los pasos a seguir, se propone un proceso sistematizado y automatizado de identificación de sistemas. Así se muestran algunas deficiencias que, en la posible aplicación a control de los modelos identificados de forma sistematizada, dan las técnicas de agrupamiento. En este capítulo se analizarán, desde el punto de vista de las deficiencias detectadas, los algoritmos de agrupamiento que podrían ser interesantes para la identificación de sistemas.

Para ello se comenzará con un análisis de la idoneidad de las funciones objetivo de los algoritmos de agrupamiento como criterios de modelado en la identificación para control por modelos locales pasando posteriormente, tras definir unas funciones de validez, a realizar un análisis cuantitativo y cualitativo de los algoritmos de agrupamiento .

Capítulo 8. Propuesta de una técnica mejorada de agrupamiento. En este capítulo, a partir de las características mejorables de las técnicas que proporcionan clases lineales que fueron estudiadas en el capítulo 7, se desarrollará una nueva función objetivo y un nuevo algoritmo de agrupamiento que será adecuado para el modelado de sistemas no lineales con modelos borrosos TS para control. La validación de su funcionamiento se realizará en base a su aplicación a diferentes ejemplos.

Capítulo 9. Integración de técnicas de agrupamiento y control. En este capítulo se evalúa el comportamiento de las técnicas de control aplicables a los modelos identificados con el algoritmo descrito en el capítulo 8. Dado que los modelos identificados desde los datos experimentales se ha procurado que sean realmente linealizaciones del proceso en el entorno de los puntos prototipos de las clases, las técnicas de control apropiadas serán las revisadas en el capítulo 4.

Capítulo 10. Ejemplos de aplicación de los algoritmos. En este capítulo se emplean los algoritmos de identificación de modelos locales propuestos a lo largo de la Tesis a dos casos: el modelado de un sensor y la identificación de un sistema térmico sencillo. En este segundo ejemplo, adicionalmente, se mostrará como desarrollar un controlador por planificación de ganancia adecuado.

Parte IV. Conclusiones.

Capítulo 11. Conclusiones. En el capítulo que cierra la Tesis se resumirán las conclusiones que se han obtenido en el desarrollo del trabajo, resaltando las diferentes aportaciones realizadas. El capítulo concluye con un apartado de cuestiones abiertas, donde se concretan una serie de líneas de trabajo que sugiere la línea de investigación de la Tesis.

Parte VI. Anexos.

A modo de ampliación de los temas comentados en el Estado del Arte se han incluido como anexos, al final del documento, una serie de monografías desarrolladas en el proceso de revisión de técnicas ya existentes que ha sido necesario para la realización de esta Tesis. En esta parte se tratan en mayor profundidad los temas de **Modelos locales**, **Planificación de ganancia**, **Algoritmos de agrupamiento** e **Identificación de modelos borrosos**. Los conocedores de estas técnicas pueden obviar su lectura.

Nota respecto a los Anexos. Con el fin de que las monografías fuesen autocontenidas y sea posible su lectura de forma independiente al resto de la Tesis, se han repetido conscientemente partes de la misma en los Anexos (minimizando así las referencias cruzadas).

Parte II
Estado del arte

Capítulo 2

Modelado e identificación de sistemas no lineales

*Estudia el pasado
si quieres pronosticar el futuro.*
Confucio.

En el presente capítulo de carácter introductorio se da una revisión general a los diferentes enfoques que se pueden encontrar para el modelado de sistemas no lineales, destacando las ventajas de los enfoques por modelos locales y los métodos denominados inteligentes.

Para ello, tras una introducción genérica al modelado de sistemas, se comentará en qué consiste el modelado por modelos locales así como, de entre los métodos inteligentes, el fundamentado en la lógica borrosa y los conjuntos borrosos: los modelos borrosos basados en reglas. Dichos modelos permiten combinar las buenas características de las técnicas inteligentes con el enfoque de los modelos locales. Seguidamente se plantearán las formas en las que se puede llevar a cabo la identificación borrosa en el caso de los sistemas no lineales, para finalizar con una sección de conclusiones.

2.1. Introducción

El proceso de modelado e identificación de modelos matemáticos de sistemas, tema de importancia muy relevante en muchas disciplinas de la ciencia y la ingeniería, se vuelve especialmente complicado al tratar con *sistemas no lineales* (Johansson, 1993). Los modelos, una vez desarrollados, pueden ser usados para la simulación, el análisis del comportamiento del sistema, una mejor comprensión de los mecanismos en los que se basa el sistema, para diseñar nuevos procesos y para controlar sistemas con el fin de realizar un mejor uso de materiales, energía e información.

La elección de la precisión con que el modelo matemático representa la realidad es una tarea muy importante (Ljung, 1999; Cellier, 1991). Si el modelo no es suficientemente preciso, los estudios de análisis, predicción o cálculo de controladores basados en el modelo obtenido no serán desarrollados de forma adecuada. Pero el modelo más preciso y complicado no es siempre el más adecuado. Un modelo adecuado deberá proporcionar

información sólo hasta el nivel que ésta sea relevante, eliminando los detalles cuando no sean necesarios. Si el modelo obtenido es demasiado simple, no se podrán representar de forma adecuada las características de interés del proceso físico en estudio. Por otra parte, un modelo nunca deberá ser tan complejo que nos impida su uso práctico.

2.2. Diferentes enfoques del modelado

El enfoque tradicional del modelado se ha basado en una completa comprensión del sistema bajo estudio y de su comportamiento, que con un tratamiento matemático adecuado da lugar a un modelo más o menos útil. Este enfoque es el llamado modelado de *caja blanca* (físico, mecanístico o basado en principios fundamentales). Sin embargo, la necesidad de una muy buena comprensión de los conceptos físicos relacionados con el sistema para realizar este tipo de modelos, hacen que su aplicación sea muy limitada en el caso de tratar con sistemas no lineales. Los problemas habituales de los modelos de caja blanca vienen por: una falta de comprensión de los fenómenos físicos en los que se basa el funcionamiento del sistema, por valores inadecuados de los parámetros del sistema o por la propia complejidad del modelo resultante. La comprensión completa de los mecanismos que hay tras un sistema real es prácticamente imposible y tener los conocimientos necesarios para la realización de un modelo físico es difícil, caro y requiere mucho tiempo. Aun en el caso de obtener el modelo, es necesario determinar valores precisos de una serie de parámetros. La estimación de estos parámetros desde datos experimentales se realiza con la identificación de sistemas, existiendo una gran variedad de métodos especialmente para sistemas lineales. Sin embargo, la mayoría de procesos reales son no lineales y pueden ser aproximados tan solo localmente por modelos lineales.

Otro enfoque del modelado asume que el proceso en estudio puede ser aproximado por una estructura de *caja negra*, que no es más que un aproximador de funciones, como por ejemplo los tratados en (Söderström and Stoica, 1989) y (Johansson, 1993). En este caso, el problema del modelado se reduce a dar una estructura adecuada del aproximador que capture correctamente la dinámica y la no linealidad del sistema. La identificación en este caso consiste en estimar los parámetros del modelo desde datos experimentales. La mayor desventaja de este enfoque es que ni la estructura ni los parámetros del modelo tienen un significado físico. Por tanto, estos modelos no pueden usarse para analizar el comportamiento del sistema de otra forma que no sea la simulación numérica y no son escalables ni válidos para procesos similares, con lo que su aplicación industrial es menor que en el caso anterior.

Por supuesto, hay una serie de técnicas de modelado que intentan combinar las ventajas de los enfoques de caja blanca y caja negra, en las que las partes bien conocidas del sistema se modelan usando el conocimiento físico y las partes con incertidumbre se modelan con técnicas de caja negra. Estos enfoques son los llamados híbridos, semimecanísticos o de *caja gris* y un esquema básico de este tipo de modelos se puede ver en la figura 2.1.

Las propuestas de identificación de sistemas no lineales se han desarrollado partiendo de estructuras simples como modelos de Hammerstein o de Wiener (Isermann *et al.*, 1992), consistiendo en bloques lineales dinámicos y no lineales estáticos habitualmente descritos por polinomios. Estas propuestas se han extendido a estructuras polinómicas

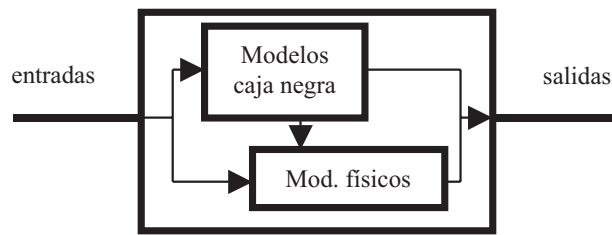


Figura 2.1: Esquema de un modelo de caja gris.

más generales basadas en las series de Volterra o polinomios de Kolmogorov (Haber and Unbehauen, 1990). La tendencia actual es emplear redes neuronales o modelos borrosos, incrementándose así la universalidad de los modelos (Nelles, 1999).

Un enfoque muy interesante para el modelado de sistemas no lineales es el que recurre a su descomposición en un conjunto más o menos grande de submodelos con un rango de validez limitado (Johansen and Murray-Smith, 1997), a los que podemos denominar *modelos locales* (ver capítulo 3). Cada uno de estos submodelos puede tener una estructura distinta (caja negra, caja blanca, caja gris) y el modelo global del sistema se obtendrá a través de la integración de los modelos locales.

La integración de los modelos locales con técnicas de *inteligencia artificial* permite incorporar conocimiento humano en los modelos. Ejemplos de ello son el *modelado borroso* (ver capítulo 5) o las *redes neuronales*, las cuales realizan un aprendizaje y adaptación imitando el funcionamiento de los sistemas neuronales biológicos a un nivel muy simplificado (en (Patterson, 1996) puede verse un estudio detallado de esta técnica).

Además, el empleo de técnicas inteligentes en general y modelos borrosos en particular como alternativa para aquellas partes del modelo donde se vaya a emplear una caja negra (lo que es posible por ser aproximadores universales de funciones, como se demuestra por ejemplo en (Wang, 1997)), siguiendo en lo posible una estructura de modelos locales, puede ayudar a una mejor comprensión del proceso por el paralelismo de estos modelos borrosos con los procesos de razonamiento humano y haciendo que el modelo de caja negra se acerque a un modelo de caja gris.

2.3. Modelado por modelos locales

Dos son las ideas básicas en el modelado por modelos locales: la partición y la interpolación (Johansen and Murray-Smith, 1997).

La *partición* intenta hacer una división del problema complejo (global) en subproblemas (locales) más simples, lo que conlleva ventajas como la existencia de soluciones más sencillas para los problemas locales o/y un incremento en la comprensión del problema global. Esta partición se puede realizar, si es posible, en base a diferentes criterios (Previdi, 1998):

- Descomposición basada en componentes físicos, en donde el problema complejo se descompone en sus componentes y el problema de modelado se resuelve uniendo las partes del sistema.

- Descomposición basada en fenómenos relevantes, cuando el modelo global se obtiene por la descripción que las leyes físicas dan del comportamiento del sistema (por ejemplo, en un sistema termoeléctrico se separa el modelado de la parte eléctrica y la térmica).
- Descomposición basada en objetivos, ya que un sistema complejo suele deber satisfacer numerosos requerimientos.
- Descomposición basada en expansión en series, que es la base de la identificación por caja negra y donde se supone la existencia de una relación funcional entre las variables del sistema.

Este último criterio es el que se empleará en este trabajo, dado que existe la posibilidad de desarrollar modelos locales lineales a los que aplicar posteriormente técnicas de control tradicionales (planificación de ganancia). En cualquier caso, cuanto mayor sea el número de modelos locales elegidos, más sencillo será cada uno de ellos. Sin embargo, esta propiedad se enfrenta a la denominada maldición de la dimensionalidad: una partición uniforme lleva a que el número de particiones crezca de forma exponencial con el número de variables.

El problema de la *interpolación* (combinación) de los modelos locales está altamente relacionado con el anterior de la partición. En algunas aplicaciones puede ser útil el cambio brusco de un modelo local a otro, mientras que otras veces puede interesar que el sistema global sea descrito simultáneamente por varios modelos locales ponderados. En otras situaciones se puede necesitar una transición suave de un modelo local a otro según el sistema se mueve en su espacio de operación. Todas estas posibles transiciones se deben obtener de forma cuidadosa junto a las particiones y pueden variar en un mismo sistema a modelar dependiendo de la aplicación del modelo.

2.4. Modelado borroso

Un sistema estático o dinámico que hace uso de los conjuntos borrosos, introducidos por Zadeh (Zadeh, 1965) en 1965, o la lógica borrosa y de la teoría matemática correspondiente es lo que se denomina un *sistema borroso*. El uso de la teoría de los conjuntos borrosos puede ser tanto en la descripción del sistema como en la especificación de los parámetros o la descripción de las variables (de entrada, de salida, de estado). Los sistemas borrosos pueden ser utilizados para diferentes fines, como el modelado, el análisis de datos, la predicción o el control. Una buena y completa revisión de esta teoría se puede encontrar en (Wang, 1997) o (Bonivento *et al.*, 1998).

El marco de modelado considerado a lo largo de esta Tesis es el fundamentado en los sistemas borrosos basados en reglas, que denominaremos *modelos borrosos*, en los que las relaciones entre las variables se describen por medio de reglas si-entonces (if-then), como:

Si la potencia de calefacción es alta **entonces** la temperatura aumentará rápidamente.

Estas reglas establecen relaciones lógicas entre las variables del sistema relacionando valores cualitativos de una variable (potencia de calefacción es alta) con valores cualitativos de otra variable (temperatura aumentará rápidamente). Los valores cualitativos tienen una interpretación lingüística clara y son llamados *términos lingüísticos* (etiquetas, valores). El concepto de modelado y análisis de sistemas por medio de variables lingüísticas fue introducido por Zadeh (Zadeh, 1973), y ha sido desarrollado en los últimos años. El significado de los términos lingüísticos con respecto a las variables de entrada y salida, que pueden ser numéricas, ha de ser definido por conjuntos borrosos adecuados. De este modo, las funciones de pertenencia de los conjuntos borrosos proporcionan un interfaz entre las entradas y las salidas numéricas y los valores cualitativos en las reglas. La estructura lógica de las reglas facilita la comprensión y el análisis del modelo de una forma cualitativa, cercana a la forma en que los humanos razonan. El proceso de inferencia puede ser visto como una interpolación entre las diferentes reglas.

Las técnicas borrosas tienen varias ventajas con respecto a otras técnicas de modelado inteligente, como las las redes neuronales comentadas anteriormente:

- Los modelos borrosos integran procesado numérico y simbólico, no restringiendo su uso a áreas en las que se requiera el conocimiento de un experto.
- La estructura basada en reglas es muy útil en el análisis de modelos borrosos adquiridos desde datos numéricos, ya que puede dar un conocimiento cualitativo del sistema, que puede ser comparado con el de los expertos.
- Dependiendo del número de valores cualitativos y de reglas consideradas, los modelos tendrán diferentes niveles de abstracción y precisión.

Evidentemente, los modelos borrosos basados en reglas son una posibilidad “inteligente” para realizar un modelado por modelos locales (reglas) de un sistema complejo en general y no lineal en particular.

2.5. Identificación borrosa

La *identificación borrosa* se refiere a las técnicas y algoritmos que permiten construir modelos borrosos desde datos experimentales y podría representarse de forma genérica por el esquema de la figura 2.2. La identificación de estos modelos se realiza en dos etapas (ver sección 5.4). La primera es la *identificación de la estructura* donde diferentes subsistemas o modos de operación son detectados. El segundo paso es la *identificación de parámetros* de los subsistemas anteriores.

Como se comenta en (Wang, 1997) existen tres tendencias, atendiendo a la forma en la que se obtiene la estructura y parámetros del modelo, que permiten la integración del conocimiento y los datos en un modelo borroso:

- La opción más directa para la obtención de un modelo borroso es el llamado *diseño basado en el conocimiento*, que consiste en seguir los siguientes pasos con la ayuda de un experto en el sistema a modelar:

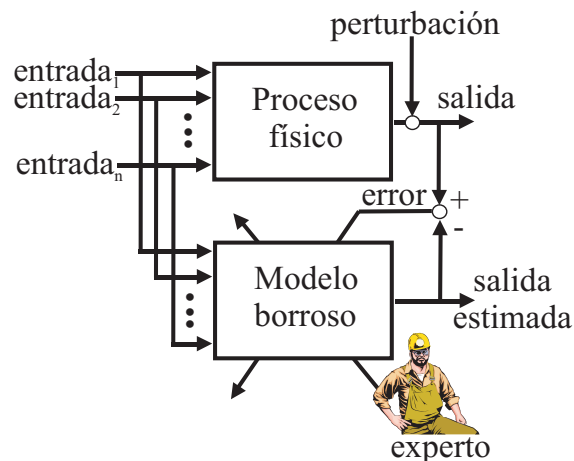


Figura 2.2: Esquema genérico de identificación borrosa.

- Seleccionar las variables de entrada y salida, la estructura de las reglas y los métodos de inferencia y desborrosificación.
 - Elegir el número de términos lingüísticos para cada variable y definir sus funciones de pertenencia.
 - Formular el conocimiento existente en forma de reglas si-entonces.
 - Validar el modelo, y si no cumple con las especificaciones volver a comenzar.
- El conocimiento experto verbal se traduce en un conjunto de reglas del tipo si-entonces. Los parámetros de esta estructura (funciones de pertenencia, peso de las reglas, etc.) se ajustan haciendo uso de los datos experimentales, por ejemplo, por mínimos cuadrados. Si para el ajuste de estos parámetros se hace uso de la propiedad de que los modelos borrosos pueden ser vistos con una estructura similar a las redes neuronales se habla de un enfoque *neuro-borroso* (Jang *et al.*, 1997). En los últimos años han aparecido numerosos métodos basados en la sinergia entre diversas técnicas ya conocidas, denominándose *sistemas híbridos* (ver capítulo 7 de (Bonivento *et al.*, 1998)). De entre estas nuevas técnicas cabría destacar las basadas en algoritmos genéticos (Kinzel *et al.*, 1994).
 - No existe ningún conocimiento previo del sistema y el modelo borroso es construido, tanto en estructura como en parámetros, usando tan solo los datos experimentales. Se prevé en este caso que las reglas puedan dar una interpretación del comportamiento del sistema, que un experto podrá interpretar y mejorar. A este enfoque se le denomina *extracción de reglas*. Para ello existen diversas técnicas como las basadas en algoritmos genéticos (Hopf and Klawonn, 1994), redes neuronales (Jang *et al.*, 1997), en plantillas (partición de los espacios de las variables de entrada y salida y posterior ajuste de las funciones de pertenencia y las reglas generadas (Wang, 1997)) y en técnicas de agrupamiento o clustering (ver capítulo 6).

Como se ha visto existen muy diversas técnicas que permiten, desde datos experimentales del proceso en estudio, la construcción de un modelo borroso basado en reglas

que reproduzca el comportamiento del mismo y permita un estudio en profundidad de los mecanismos que lo gobiernan. Una gran parte de esas técnicas exigen la aplicación del conocimiento heurístico a priori, en la fase de definición de la estructura de modelo, realizándose la identificación tan solo en el sentido de un ajuste fino de los parámetros. Este tipo de técnicas tienen como gran desventaja el cuello de botella que supone la extracción del conocimiento desde los operarios expertos en la fase inicial del proceso de modelado.

Este grave problema se puede evitar con aquellos enfoques que desde los datos experimentales nos dan tanto la estructura del modelo como el ajuste de sus parámetros, requiriendo la participación del usuario experto tan solo en una posterior fase de validación y mejora del modelo, que además es sencilla por la interpretabilidad de las reglas en clave lingüística. De entre todas las técnicas existentes con este enfoque, se dirigirá este trabajo hacia las denominadas *técnicas de agrupamiento* (o *clustering*), que toman formas de trabajo y algoritmos de los campos de reconocimiento de imágenes y patrones que se revisan, por ejemplo, en (Duda *et al.*, 2000).

Si el modelado de sistemas no lineales con el uso de herramientas basadas en la teoría de los conjuntos borrosos (el llamado modelado borroso) se ha revelado como una potente arma en los casos en los que no se tiene un conocimiento profundo del sistema a estudiar, o el que se tiene es heurístico o impreciso, su combinación con las técnicas de agrupamiento (que se ha venido desarrollando y aplicando con éxito en los últimos años) hacen que este aspecto del modelado tenga un futuro muy prometedor.

2.6. Conclusiones

La introducción genérica realizada al modelado de sistemas ha llevado a comentar los conceptos básicos en los que descansan las teorías del modelado por modelos locales y los modelos borrosos basados en reglas (representantes de los métodos inteligentes), así como la estrecha relación entre ellos. La forma de identificar estos modelos desde datos experimentales son las técnicas de identificación borrosa, uno de cuyos enfoques toma las formas de trabajo de los campos de reconocimiento de formas (algoritmos de agrupamiento).

De este modo queda justificada la necesidad de una revisión del estado actual de teorías tan diversas como el modelado basado en modelos locales, el modelado borroso y las técnicas de agrupamiento. Todo ello se realizará, por ese orden, en los capítulos 3, 5 y 6. Igualmente, dado que se pretende emplear los modelos identificados para el control, se revisarán en el capítulo 4 las técnicas de control aplicables a sistemas con modelos locales.

Capítulo 3

Modelado basado en modelos locales

*La matemática es el alfabeto
con el que Dios escribió el mundo.*
Galileo Galilei.

En este capítulo, tras definir los modelos matemáticos de los sistemas que se van a tratar en la Tesis, se plantearán las diferentes formas de realizar particiones de modelos locales de sistemas no lineales, todas ellas mediante linealización (desarrollo en serie) en el entorno de puntos de operación empleando el Jacobiano de la planta.

Aunque el enfoque más habitual (clásico) de partición es aquel en el que los puntos de operación se corresponden con uno o varios puntos de equilibrio, su limitación de que cualquier análisis o diseño basado en esos modelos solo es válido cuando se opera cerca del equilibrio, lleva a que este capítulo se centre en revisar marcos alternativos de linealización con técnicas que permiten la linealización en torno a puntos que no son de equilibrio, como la linealización en torno a puntos de operación o la linealización basada en la velocidad. De este modo se proporciona a los modelos borrosos identificados, tal y como se perfila en el capítulo 2 y se describirá a lo largo de la Tesis, un marco teórico apropiado para trabajar en el entorno de cualquier punto (de equilibrio o no) y ser empleados para análisis o diseño.

3.1. Introducción

La estrategia que propone el modelado (y control) basado en modelos locales (Johansen and Murray-Smith, 1997) consiste en dividir un sistema complejo en subsistemas más sencillos que puedan ser analizados de forma individual. La clave, por tanto, consiste en la elección de aquellos parámetros adecuados que permitan realizar dicha partición. Por otra parte, se hace necesario un elemento supervisor que sea de algún modo capaz de integrar de nuevo todos estos subsistemas en un sistema global. Si se eligen modelos lineales como modelos locales (todos ellos con la misma estructura, manteniéndose así las variables de estado), la integración de todos ellos nos lleva a un *modelo lineal de parámetros variables* LPV (linear parameter-varying), los cuales son la base de las técnicas clásicas para el desarrollo de controladores por planificación de ganancia y

que se revisan, por ejemplo, en (Shamma and Athans, 1990; Rugh, 1991; Aström and Wittenmark, 1995).

Para obtener un modelo LPV, la opción más común está basada en la linealización del *Jacobiano* de la planta no lineal en el entorno de una familia de puntos de operación, que consiste en aplicar la teoría del desarrollo en serie en torno a esos puntos, olvidando los términos de orden superior. Lo más habitual (por la gran cantidad de literatura existente al respecto, como se observa en las revisiones de (Leith and Leithead, 2000) o (Rugh and Shamma, 2000)) es que los puntos de operación se correspondan con una trayectoria deseada o un punto de equilibrio, con la desventaja de que el desarrollo en serie solo es válido en el entorno de esa trayectoria específica o ese punto de equilibrio. Esta condición se puede relajar desarrollando familias de linealizaciones en torno a conjuntos de trayectorias deseadas o puntos de equilibrio, con lo que se aumenta el rango de operación del sistema.

La limitación común a las opciones descritas anteriormente (y revisadas en el anexo A) sobre linealización en el entorno de puntos de equilibrio es que, generalmente, cualquier análisis o diseño basado en esos modelos solo será válido cuando se opere cerca del equilibrio. Afortunadamente, existen algunas técnicas que permiten la linealización en torno a puntos que no son de equilibrio, como la linealización en torno a puntos de operación (Hunt and Johansen, 1997) o la linealización basada en la velocidad (Leith and Leithead, 2000; Rugh and Shamma, 2000). Dichas técnicas son las más apropiadas en el marco de esta Tesis y son comentadas en este capítulo.

En cualquier caso (se elija la forma de linealización que se elija) se llega a una familia parametrizada de plantas linealizadas y es la base de lo que se llama planificación por linealización. Generalmente, la parametrización corresponde a valores fijos de las variables de planificación o funciones de ellas, que son variables internas de la planta y/o señales exógenas.

3.2. Modelos de sistemas no lineales

El modelo que se va a considerar de los sistemas no lineales que se desean estudiar (identificar y controlar) será el de la planta genérica no lineal dada por:

$$\begin{aligned} \dot{x} &= f(x, u), \\ y &= g(x, u). \end{aligned} \tag{3.1}$$

donde x es el vector de estado $n \times 1$, u es el vector de entrada $m \times 1$, y es el vector de salida $p \times 1$ medible para el controlador, y se ha asumido que x y u incluyen todas las dependencias con variables exógenas. Finalmente, $f(\cdot)$ y $g(\cdot)$ son funciones vectoriales suaves, es decir, con derivadas continuas (primera y siguientes) en cualquier instante.

Si se considera (como ocurre en muchas aplicaciones reales para control) que no hay dependencia directa de la salida con la entrada (Rugh and Shamma, 2000), la variable u desaparece de los argumentos de g en (3.1). Finalmente, con ánimo de simplificar, muchas veces se asume que el vector de estado es medible en su totalidad. Teniendo en cuenta estas dos últimas consideraciones (Johansen and Foss, 1992), la ecuación de

salida resultante es la identidad y la representación del sistema se reduce tan solo la ecuación:

$$\dot{x} = f(x, u), \quad (3.2)$$

con $u \in R^m$, $x = y \in R^n$ medible y siendo $f(\cdot)$ una función suave, continua y derivable. Esta ecuación queda en el caso discreto:

$$x_{k+1} = f(x_k, u_k). \quad (3.3)$$

Si el estado no está disponible en (3.3), se puede reconstruir desde un vector de salidas (anteriores) y entradas (actual y anteriores) que sea lo suficientemente amplio y que cumpla las condiciones de observabilidad (Söderström and Stoica, 1989). De este modo se obtiene el conocido modelo no lineal autoregresivo con entrada exógena NARX (nonlinear autoregressive model with exogenous input) de un sistema:

$$y_{k+1} = f(y_k, \dots, y_{k-n_y}, u_k, \dots, u_{k-n_u}). \quad (3.4)$$

3.3. Linealización en torno a puntos de operación

Dado el sistema no lineal (3.1), el enfoque habitual es considerar una familia de puntos de equilibrio del sistema y aproximarlos localmente por un sistema lineal variante en el tiempo, como se describe en el apartado A.3.3.

Una interpretación de esta dinámica clásica linealizada es que aproxima la dinámica del sistema no lineal para valores iniciales y entradas que se desvían ligeramente de x_e y u_e (valores de x y u en el equilibrio, respectivamente). La razón más importante de la popularidad de la linealización clásica es la existencia de una teoría completa y perfectamente desarrollada para el análisis de las propiedades de estabilidad local en el equilibrio de los sistemas no lineales en términos de sus versiones linealizadas, así como la existencia de la conocida teoría de control basada en la linealización.

Supongamos ahora que la dinámica del sistema no lineal se aproxima en el entorno de un punto de operación $(x_o, u_o) \in R^{n+m}$, el cual no es necesario que sea un punto de equilibrio (Wang *et al.*, 1996; Hunt and Johansen, 1997; Shorten *et al.*, 1999). Es decir, los posibles puntos de operación del sistema vienen dados por:

$$\psi = \{(x, u) \in R^{n+m}\} \quad (3.5)$$

con lo que la linealización del sistema no lineal en torno al punto (x_o, u_o) es:

$$\begin{aligned} \dot{x} &= \nabla_x f(x_o, u_o)\delta x + \nabla_u f(x_o, u_o)\delta u + f(x_o, u_o), \\ y &= \nabla_x g(x_o, u_o)\delta x + \nabla_u g(x_o, u_o)\delta u + g(x_o, u_o), \end{aligned} \quad (3.6)$$

donde $\delta x = x - x_o$, $\delta u = u - u_o$, por ejemplo, $\nabla_x f$ indica la matriz Jacobiana de f respecto de x (lo mismo es aplicable al resto de ∇). Evidentemente, $f(x_o, u_o)$ y $g(x_o, u_o)$ serían nulas si el punto de operación fuese un punto de equilibrio. Esta ecuación se puede

reescribir de una forma más adecuada para una posterior extensión a un conjunto de puntos de operación:

$$\begin{aligned}\dot{x} &= \nabla_x f(x_o, u_o)x + \nabla_u f(x_o, u_o)u + d_f(x_o, u_o), \\ y &= \nabla_x g(x_o, u_o)x + \nabla_u g(x_o, u_o)u + d_g(x_o, u_o),\end{aligned}\tag{3.7}$$

donde

$$\begin{aligned}d_f(x_o, u_o) &= f(x_o, u_o) - \nabla_x f(x_o, u_o)x_o - \nabla_u f(x_o, u_o)u_o, \\ d_g(x_o, u_o) &= g(x_o, u_o) - \nabla_x g(x_o, u_o)x_o - \nabla_u g(x_o, u_o)u_o.\end{aligned}\tag{3.8}$$

Desde un punto de vista matemático, esta linealización en torno a puntos de operación (Hunt and Johansen, 1997) corresponde a aproximar las funciones f y g por sus planos tangentes en el punto (x_o, u_o) . Sin embargo, la interpretación del sistema (3.7) es conceptualmente diferente a la clásica, ya que (x_o, u_o) solo puede ser un estado transitorio y nunca un punto de equilibrio. Una interpretación de (3.7) es que aproxima la (posiblemente transitoria) dinámica del sistema no lineal cuando la trayectoria está cerca a (x_o, y_o) . Esta aproximación no es de inmediata utilidad para el análisis de estabilidad, pero es útil para el diseño de control estable de altas prestaciones por planificación de ganancia. El beneficio final de esta forma de trabajo es que la dinámica del transitorio (fuera del equilibrio) del sistema de control puede ser mejorada significativamente (Hunt and Johansen, 1997).

Esta linealización en un punto de operación fuera del equilibrio se puede extender a un conjunto de puntos Ψ del tipo (3.5). Todos estos puntos de operación se pueden parametrizar en función de una variable de planificación ϕ , que puede incluir todas las variables medibles del sistema (internas o externas, estén en ψ o no) con la única restricción de que $\phi(t)$ siempre esté disponible (Rugh and Shamma, 2000), ya que podría ser usada posteriormente para el diseño del control. A $\phi \in \Phi$ se le llama régimen de operación y al conjunto de todos ellos, Φ , conjunto de operación.

Eliminando los términos de orden superior, el sistema (3.1) puede ser aproximado por el sistema lineal variante en el tiempo que forma la familia de linealizaciones en el entorno de puntos de operación:

$$\begin{aligned}\dot{x} &= \nabla_x f(x_o(\phi), u_o(\phi))x + \nabla_u f(x_o(\phi), u_o(\phi))u + d_f(x_o(\phi), u_o(\phi)), \\ y &= \nabla_x g(x_o(\phi), u_o(\phi))x + \nabla_u g(x_o(\phi), u_o(\phi))u + d_g(x_o(\phi), u_o(\phi)),\end{aligned}\tag{3.9}$$

donde $\phi \in \Phi$, $d_f(\cdot)$ y $d_g(\cdot)$ siguen la definición de (3.8) y, por ejemplo, $\nabla_x f$ indica la matriz Jacobiana de f respecto de x (lo mismo es aplicable al resto de ∇). La interpretación es que para cada valor de linealización ϕ , el sistema (3.9) describe el comportamiento local de la planta no lineal en el entorno del punto de operación correspondiente.

La versión discreta de esta aproximación viene dada por:

$$\begin{aligned}x_{k+1} &= \nabla_x f(x_o(\phi), u_o(\phi))x_k + \nabla_u f(x_o(\phi), u_o(\phi))u_k + d_f(x_o(\phi), u_o(\phi)), \\ y_k &= \nabla_x g(x_o(\phi), u_o(\phi))x_k + \nabla_u g(x_o(\phi), u_o(\phi))u_k + d_g(x_o(\phi), u_o(\phi)),\end{aligned}\tag{3.10}$$

donde $\phi \in \Phi$, $d_f(\cdot)$ y $d_g(\cdot)$ siguen la definición de (3.8) y, por ejemplo, $\nabla_u g$ indica la matriz Jacobiana de g respecto de u (lo mismo es aplicable al resto de ∇).

En (Johansen and Foss, 1992; Hunt and Johansen, 1997) se pueden encontrar ejemplos en los que se aplican los modelos descritos en este apartado.

3.4. Linealización basada en la velocidad

Se puede obtener una representación alternativa a (3.7) derivando a ambos lados de la igualdad y, haciendo el cambio de variable $\dot{x} = v$, queda:

$$\begin{aligned}\dot{x} &= v, \\ \dot{v} &= (\nabla_x f(x_o, u_o) + \nabla_x d_f(x_o, u_o))v + (\nabla_u f(x_o, u_o) + \nabla_u d_f(x_o, u_o))\dot{u}, \\ \dot{y} &= (\nabla_x g(x_o, u_o) + \nabla_x d_g(x_o, u_o))v + (\nabla_u g(x_o, u_o) + \nabla_u d_g(x_o, u_o))\dot{u},\end{aligned}\tag{3.11}$$

En (Leith and Leithead, 1998b) se demuestra que (3.11), la llamada solución a la linealización basada en la velocidad aproxima localmente la solución del sistema no lineal (3.1) en un punto de operación genérico (x_o, u_o) (no necesariamente un punto de equilibrio). La solución de la linealización basada en la velocidad (3.11) es solo una aproximación local, pero se puede encontrar una solución asociada a cada punto de operación de un sistema no lineal, con grado de precisión arbitrario, y parametrizarlas en función de una variable de planificación ρ (con las mismas características de ϕ en (3.9)). Así pues, una familia de linealizaciones basadas en la velocidad parametrizada por la variable de planificación ρ engloba toda la dinámica del sistema no lineal (ρ captura la estructura no lineal del sistema) quedando:

$$\begin{aligned}\dot{x} &= v, \\ \dot{v} &= (\nabla_x f(x_o(\rho), u_o(\rho)) + \nabla_x d_f(x_o(\rho), u_o(\rho)))v + \\ &\quad + (\nabla_u f(x_o(\rho), u_o(\rho)) + \nabla_u d_f(x_o(\rho), u_o(\rho)))\dot{u}, \\ \dot{y} &= (\nabla_x g(x_o(\rho), u_o(\rho)) + \nabla_x d_g(x_o(\rho), u_o(\rho)))v + \\ &\quad + (\nabla_u g(x_o(\rho), u_o(\rho)) + \nabla_u d_g(x_o(\rho), u_o(\rho)))\dot{u},\end{aligned}\tag{3.12}$$

Esta representación tiene la ventaja, con respecto a (3.9), de que las funciones no lineales d_f y d_g han desaparecido. La relación entre la solución de un sistema no lineal y las soluciones de los miembros de la familia de linealizaciones basadas en la velocidad asociada a él, se puede usar para derivar condiciones que relacionen la estabilidad del sistema no lineal con la estabilidad de sus linealizaciones basadas en la velocidad (Leith and Leithead, 1998b). Métodos generales de análisis de estabilidad, como la teoría de Lyapunov, se pueden aplicar para derivar condiciones de estabilidad basadas en la velocidad.

La linealización basada en la velocidad se convierte en una linealización basada en incrementos (Δ) para el caso discreto:

$$\begin{aligned}
\Delta x_k &= x_k - x_{k-1}, \\
\Delta x_{k+1} &= (\nabla_x f(x_o, u_o) + \nabla_x d_f(x_o, u_o))\Delta x_k + (\nabla_u f(x_o, u_o) + \nabla_u d_f(x_o, u_o))\Delta u_k, \\
\Delta y_k &= (\nabla_x g(x_o, u_o) + \nabla_x d_g(x_o, u_o))\Delta x_k + (\nabla_u g(x_o, u_o) + \nabla_u d_g(x_o, u_o))\Delta u_k,
\end{aligned} \tag{3.13}$$

que aproxima (para periodos de muestreo pequeños) al sistema no lineal localmente en un punto de operación (x_o, u_o) , de nuevo con la ventaja de que las funciones d_f y d_g no aparecen.

Evidentemente, se puede encontrar una aproximación local del tipo linealización basada en la velocidad asociada a cada punto de operación del sistema no lineal. Así pues, con grado de precisión arbitrario (dependiente de un elevado número de puntos de operación y un bajo periodo de muestreo), se puede encontrar la aproximación al sistema no lineal mediante la familia de linealizaciones basadas en la velocidad parametrizando otra vez con la variable de planificación ρ , englobando así toda la dinámica del sistema no lineal discreto:

$$\begin{aligned}
\Delta x_k &= x_k - x_{k-1}, \\
\Delta x_{k+1} &= (\nabla_x f(x_o(\rho), u_o(\rho)) + \nabla_x d_f(x_o(\rho), u_o(\rho)))\Delta x_k + \\
&\quad + (\nabla_u f(x_o(\rho), u_o(\rho)) + \nabla_u d_f(x_o(\rho), u_o(\rho)))\Delta u_k, \\
\Delta y_k &= (\nabla_x g(x_o(\rho), u_o(\rho)) + \nabla_x d_g(x_o(\rho), u_o(\rho)))\Delta x_k + \\
&\quad + (\nabla_u g(x_o(\rho), u_o(\rho)) + \nabla_u d_g(x_o(\rho), u_o(\rho)))\Delta u_k,
\end{aligned} \tag{3.14}$$

Algunos ejemplos de linealizaciones basadas en velocidad se pueden revisar en (Leith and Leithead, 1998b).

3.5. Conclusiones

Vistas las diferentes opciones para realizar particiones de modelos locales de sistemas no lineales mediante linealización en el entorno de puntos de operación empleando el Jacobiano de la planta, las opciones que se muestran más interesantes y por tanto se han revisado en este capítulo son las relacionadas con familias de linealizaciones en torno a puntos de operación.

Esto es debido a que cuando los puntos de operación se corresponden con puntos de equilibrio cualquier análisis o diseño basado en esos modelos solo será válido cuando se opere cerca del equilibrio. Sin embargo, la linealización en torno a puntos de operación o la linealización basada en la velocidad permiten la linealización en torno a puntos que no son de equilibrio, siendo por tanto las opciones más generales y que se intentará relacionar con los modelos borrosos y técnicas de clustering que se revisarán en capítulos posteriores.

Capítulo 4

Control con modelos locales

*La originalidad
es la vuelta a los orígenes.*
Antoni Gaudí i Cornet.

En este capítulo se verán las técnicas de control aplicables cuando se dispone de modelos locales de sistemas no lineales como los descritos en el capítulo 3. Estas técnicas son las que, basándose en diseño de controladores lineales para los modelos locales identificados, consiguen obtener controladores para el conjunto del sistema no lineal. Por ello se revisará el control por planificación de ganancia y los métodos basados en LMIs.

4.1. Introducción

Una vez se ha desarrollado (con algún método analítico como los comentados en el capítulo anterior o de identificación como los que se propondrán a partir del capítulo 7) un conjunto parametrizado de modelos locales lo suficientemente preciso de un proceso no lineal, hay dos enfoques principales para realizar el control del proceso.

Una de las opciones más utilizadas últimamente (Tanaka and Sugeno, 1992) para el desarrollo de controladores basados en modelos locales (por ejemplo, borrosos) es la que emplea la resolución de LMIs (*Linear Matrix Inequalities*). Esta técnica tiene la ventaja de asegurar la estabilidad del sistema (Bondía, 1997) independientemente de la idoneidad de las linealizaciones, pero da resultados muy conservadores que pueden llevar a que un conjunto de controladores apropiados no satisfagan las condiciones de estabilidad. Además, convierten el problema del diseño del control en una resolución matemática de ecuaciones, con lo que el Ingeniero de Control pierde la capacidad de emplear su conocimiento de experto para mejorar el comportamiento de los controladores locales así diseñados (apartado 4.3).

Sin embargo, si se desea explotar el sentido físico de los modelos lineales como linealizaciones del proceso, se pueden emplear las mismas técnicas de control que se basan en representar sistemas no lineales por un modelo linealizado que lo aproxime y diseñar un controlador para él. La restricción de que dicho modelo solo puede obtener una representación local del comportamiento del sistema, tradicionalmente se elimina

diseñando diferentes controladores para diferentes modelos lineales en diferentes puntos de operación e interpolando estos controladores, con lo que se llega a un controlador no lineal global. A grandes rasgos, esto es lo que se conoce como *control por planificación de ganancia*.

La planificación de ganancia es una de las técnicas más habituales para el control de procesos no lineales y ha sido empleada satisfactoriamente desde hace años (Källström *et al.*, 1979; Stein *et al.*, 1977) en aplicaciones que van desde la ingeniería aeroespacial al control de procesos. Sin embargo, hasta los años noventa es raro encontrar un tratamiento teórico de la planificación de ganancia (Shamma and Athans, 1990; Rugh, 1991; Aström and Wittenmark, 1995), aunque afortunadamente en los últimos años han aparecido interesantes revisiones completas sobre el tema (Leith and Leithead, 2000; Rugh and Shamma, 2000).

4.2. Control por planificación de ganancia

En términos generales, el diseño de un controlador por planificación de ganancia para una planta no lineal se suele describir como un procedimiento de cuatro pasos (Aström and Wittenmark, 1995; Shamma and Athans, 1990; Hyde and Glover, 1993), con diferentes opciones técnicas en cada uno de ellos (ver anexo B). Estos cuatro pasos son:

- calcular un **modelo lineal de parámetros variables** LPV (linear parameter-varying), con una linealización analítica basada en el Jacobiano de la planta (capítulo 3) o por algún método de identificación que asegure la obtención de modelos de estas características,
- diseñar **controladores mediante métodos lineales**, lo que normalmente lleva a una familia de controladores lineales independientes diseñados para valores aislados (puntos de operación) de las variables de planificación.
- realizar la **planificación de ganancia** propiamente dicha, mediante un proceso de interpolación de la familia de controladores lineales de forma que los coeficientes de los controladores (ganancias) varíen (sean planificados) de acuerdo con los valores actualizados de las variables de planificación, y
- evaluar el **rendimiento** del control diseñado, aunque habitualmente solo se puede realizar un estudio analítico a nivel local de las propiedades de estabilidad y rendimiento del controlador por planificación de ganancia, mientras que la evaluación de su rendimiento no local está basado en simulaciones.

Como regla general (Leith and Leithead, 2000), cuanto mayor sea el número de modelos locales elegidos, más sencillo será encontrar un controlador que funcione adecuadamente para cada uno de los regímenes. De hecho, la familia de linealizaciones puede tener un número infinito de miembros, pero es por supuesto más práctico y atractivo considerar en el proceso de diseño de controladores un pequeño conjunto finito de miembros de la familia de linealizaciones de la planta.

4.2.1. Controladores basados en modelos LPV del sistema en puntos de operación

A pesar de la gran utilización de los controladores por planificación de ganancia, las técnicas inteligentes no han empleado todo su potencial. En este apartado se muestra un enfoque de la planificación de ganancia que evidencia su similitud con el modelado y control borroso (Johansen and Murray-Smith, 1997).

Considérese en este caso el desarrollo de un controlador para cada uno de los c puntos de operación $(x_o, u_o) \in R^{n+m}$ en el entorno de los cuales se ha aproximado la dinámica del sistema. Todos estos puntos de operación ψ , definidos según (3.5), deberán pertenecer a un espacio de operación Ψ (conjunto de puntos operación). Para realizar la interpolación de los controladores existen numerosos métodos (ver anexo B), pero muchos de ellos son soluciones ad-hoc a problemas concretos y no están lo suficientemente justificados como para ser planteados como una metodología genérica (Rugh, 1991). Una alternativa que permite sistematizar este proceso consiste parametrizar los puntos de operación en función de una variable de planificación $\phi \in \Phi$ a la que se le llama régimen de operación (el conjunto Φ de todos ellos es el conjunto de operación), y combinar (ponderar) adecuadamente llegando a la representación de (3.9) en la forma de redes de modelos locales.

Definición de Red de Modelos Locales. *Dados c puntos de operación y sus modelos locales según (3.9) y seleccionadas adecuadamente las funciones μ de validez o pertenencia (normalizadas de la forma $\sum_i \mu_i(\phi) = 1$), se define una Red de Modelos Locales como la interpolación de los c modelos locales dada por el aproximador universal (Hunt et al., 1996)*

$$\begin{aligned} \dot{x} &= \sum_{i=1}^c \mu_i(\phi) f_i(x, u), \\ y &= \sum_{i=1}^c \mu_i(\phi) g_i(x, u), \end{aligned} \tag{4.1}$$

que es una familia de aproximación todo lo precisa que se quiera del sistema no lineal (3.1).

Esta aproximación describe, siguiendo la forma marcada por (3.7), el comportamiento local de la planta no lineal de forma lineal para en el entorno de cada valor de la variable de planificación en la linealización (ϕ_i tal que $i \in [1, c]$) mediante $f_i(x, u)$ y $g_i(x, u)$, el sistema (4.1). (4.1) es además equivalente bajo ciertas condiciones (Hunt et al., 1996) a los modelos borrosos Takagi-Sugeno que se describirán en el capítulo 5.

El resultado será, en lo que respecta al control, un conjunto indexado ($i \in [1, c]$) de controladores lineales, cada uno de los ellos diseñado (matrices J_i, K_i que llevan a la acción de control u) para asegurar un funcionamiento adecuado (error e pequeño o nulo) en bucle cerrado cuando se emplea con la linealización de la planta que le corresponde:

$$u = J_i x + K_i e + u_{o_i}. \tag{4.2}$$

Los términos independientes (offsets) que aparecen en (4.1) han de ser tratados aparte, considerándolos por ejemplo como una perturbación (Hunt and Johansen, 1997) a eliminar con una prealimentación o acción integral.

Para que el conjunto indexado de controladores lineales sea aplicable, se ha de conseguir en base a ellos un controlador no lineal único, consistente en un controlador lineal de parámetros variables parametrizado con ϕ :

$$u = J_i(\phi)x + K_i(\phi)e + u_{oi}(\phi). \quad (4.3)$$

Esta combinación de los controladores se puede realizar mediante una red de controladores locales.

Definición de Red de Controladores Locales. *Dado un conjunto de controladores locales lineales diseñados para los modelos locales de la planta (4.1), se define la Red de Controladores Locales como la combinación de esos controladores independientes mediante:*

$$u = \sum_i^c [\mu_i(\phi)J_i]x + \sum_i^c [\mu_i(\phi)K_i]e + \sum_i^c [\mu_i(\phi)u_{oi}(\phi)], \quad (4.4)$$

donde para la combinación de los controladores se emplea (Hunt and Johansen, 1997) la misma regla de variación que dan las funciones de pertenencia (validez) para el modelo de la planta (4.1).

Esta definición lleva implícita la suposición de que, de algún modo, la variación entre modelos de la planta captura su dinámica. El equivalente discreto es:

$$u_k = \sum_i^c [\mu_i(\phi)J_i]x_k + \sum_i^c [\mu_i(\phi)K_i]e_k + \sum_i^c [\mu_i(\phi)u_{oi}(\phi)]. \quad (4.5)$$

Hay que tener en cuenta que modelos como (4.1) son la combinación de los modelos locales que se desprenden de (3.1), pero la relación entre las soluciones de ambos sistemas no es directa. El desarrollo en serie de (4.1) en el entorno del punto de operación aproxima perfectamente a (3.1), pero contiene términos cruzados de las derivadas de las funciones de pertenencia (Hunt and Johansen, 1997), con lo que en aquellas zonas en las que la derivada de alguna de las funciones de pertenencia tome valores elevados, la dinámica del desarrollo en serie estará muy relacionada con las funciones de pertenencia y poco con el modelo local.

Además, el análisis de los resultados de estabilidad de estos sistemas desarrollados en torno a puntos de operación realizado, por ejemplo, empleando la teoría de tiempo detenido (frozen-time, ver anexo B) para establecer condiciones suficientes de estabilidad del sistema lineal variante en el tiempo (3.9) o (3.10) no tiene mucho sentido (Hunt and Johansen, 1997), ya que (x_o, u_o) puede ser, como ya se ha comentado, un estado transitorio y no un punto de equilibrio. Lo que sí que es posible es llegar a resultados como cotas (Díez and Previdi, 2001) (aplicables a herramientas de control robusto) en las que interviene la granularidad del modelo y la suavidad del cambio entre puntos de operación para asegurar un buen comportamiento del control basado en estos modelos.

En definitiva, la aplicabilidad de la filosofía clásica de control por planificación de ganancia en los controladores basados en modelos LPV del sistema en diferentes puntos de operación se restringe a situaciones en las que hay una combinación de modelos mínima o el sistema es lo suficientemente lento como para que la influencia de los términos cruzados en los que aparecen las derivadas de las funciones de pertenencia es

pequeña (Shorten *et al.*, 1999; Leith and Leithead, 1999a). Por tanto, se puede decir que la eficiencia del controlador desarrollado es muy dependiente de las características del sistema así como de su modelo disponible y hay que recordar que, a todas estas consideraciones, hay que añadir las condiciones de variación lenta asociadas con los métodos de planificación de ganancia clásicos.

4.2.2. Controladores basados en modelos LPV del sistema linealizados por velocidad

Cuando se dispone de una familia de linealizaciones de la planta basada en la velocidad (3.12), se debe determinar la familia de linealizaciones del controlador basado en la velocidad tal que la familia en bucle cerrado alcance los requerimientos de diseño deseados. Como cada miembro (3.11) de los que componen la familia de la planta es lineal (ha desaparecido el término independiente), se pueden emplear métodos de diseño lineales para el diseño de cada uno de los controladores. Una aproximación a la familia de linealización basada en la velocidad de la planta en la que se interpolan un número finito de linealizaciones basadas en la velocidad (modelos locales) es la denominada red de modelos locales por velocidad.

Definición de Red de Modelos Locales por Velocidad. *Dado un conjunto indexado ($i \in [1, c]$) de modelos locales según (3.12) (uno de ellos para cada punto de operación (x_i, u_i) elegido), una Red de Modelos Locales por Velocidad es el modelo global consistente en la ponderación:*

$$\begin{aligned}
 \dot{x} &= v, \\
 \dot{v} &= \sum_i^c \mu_i(\rho) [(\nabla_x f(x_o(\rho), u_o(\rho)) + \nabla_x d_f(x_o(\rho), u_o(\rho)))]v + \\
 &\quad + \sum_i^c \mu_i(\rho) [(\nabla_u f(x_o(\rho), u_o(\rho)) + \nabla_u d_f(x_o(\rho), u_o(\rho)))]\dot{u}, \\
 \dot{y} &= \sum_i^c \mu_i(\rho) [(\nabla_x g(x_o(\rho), u_o(\rho)) + \nabla_x d_g(x_o(\rho), u_o(\rho)))]v + \\
 &\quad + \sum_i^c \mu_i(\rho) [(\nabla_u g(x_o(\rho), u_o(\rho)) + \nabla_u d_g(x_o(\rho), u_o(\rho)))]\dot{u},
 \end{aligned} \tag{4.6}$$

en la que μ sigue la misma definición que en (4.1).

Ahora se podrá diseñar un controlador para cada uno de los modelos locales según:

$$u = J_i v + K_i \dot{e}, \tag{4.7}$$

donde se asume que las derivadas del estado de la planta y del error se pueden medir o son calculables. El último paso del proceso de diseño será la realización de un controlador no lineal desde la familia de controladores linealizados por velocidad. La interpolación entre los controladores lineales lleva a una familia (red) de controladores linealizados por velocidad.

Definición de Red de Controladores Locales por Velocidad. *Dado un conjunto de controladores locales lineales diseñados según (4.7), la combinación de éstos con la misma regla de variación que dan las funciones de pertenencia (validez) para el modelo de la planta (4.6), genera la Red de Controladores Locales por Velocidad:*

$$\begin{aligned} \dot{x} &= v, \\ u &= \sum_i^c [\mu_i(\rho) J_i] v + \sum_i^c [\mu_i(\rho) K_i] \dot{e}, \end{aligned} \quad (4.8)$$

El controlador no lineal definitivo en su forma basada en velocidad se puede obtener directamente permitiendo que la variables de planificación ρ varíe con el punto de operación. La versión discreta será:

$$\begin{aligned} \Delta v_k &= v_k - v_{k-1}, \\ \Delta u_k &= \sum_i^c [\mu_i(\rho) J_i] \Delta v_k + \sum_i^c [\mu_i(\rho) K_i] \Delta e_k. \end{aligned} \quad (4.9)$$

En estos casos, en contraste con las representaciones de las redes de modelos locales, al desaparecer el término independiente (offset) se mantiene una relación directa (Leith and Leithead, 1999a) entre las características dinámicas de (4.6) y de (4.8). Por supuesto, hay que tener un cuidado especial (Leith and Leithead, 1998c; Leith and Leithead, 1999a; Leith and Leithead, 1999b) con las realizaciones basadas en velocidad, pues hay que recordar que sus entradas son derivadas de variables. Además, debido a las operaciones de diferenciación e integración asociadas con los modelos locales en su representación por velocidad, el orden de estos modelos puede ser mayor que el de la representación directa (3.1).

4.3. Análisis de estabilidad mediante LMIs

La estabilidad y rendimiento de los sistemas no lineales diseñados por el método de planificación de ganancia basada en la velocidad, a diferencia de los métodos basados en puntos de operación, puede analizarse mediante cualquier método de propósito general para sistemas no lineales (Leith and Leithead, 2000) como pequeña ganancia, teoría de tiempo detenido (ambos se comentan en el anexo B) o la teoría de Lyapunov cuya aplicación al caso que nos ocupa se comenta a continuación.

Los problemas de optimización numérica llamados desigualdades matriciales lineales o LMI (Linear Matrix Inequalities) han recibido mucha atención recientemente (Boyd *et al.*, 1994). En el campo de la ingeniería de sistemas y de control, la importancia de la optimización con LMI viene por la amplia variedad de problemas que pueden reorganizarse como problemas de LMI, con lo que la resolución del problema LMI es equivalente a encontrar una solución al problema de control original. Aunque tan solo en algunos pocos casos se puede encontrar una solución analítica a estos problemas, siempre se pueden resolver eficientemente de forma numérica.

Definición de LMI. Una LMI es una desigualdad matricial de la forma

$$F(x) = F_0 + \sum_{i=1}^m x_i F_i > 0, \quad (4.10)$$

donde $x^T = (x_1, x_2, \dots, x_m)$ es la variable y las matrices simétricas $F_i = F_i^T \in R^{n \times n}$, $i = 0, \dots, m$ son conocidas. El símbolo de desigualdad > 0 significa que $F(x)$ es definida positiva.

La LMI (4.10) es una restricción convexa de x , es decir, el conjunto $\{x \mid F(x) > 0\}$ es convexo. Aunque (4.10) puede representar una gran variedad de restricciones convexas de x , el caso de interés en el marco de esta Tesis es el que refleja las restricciones que aparecen en la teoría de control, como son las desigualdades de Lyapunov para estabilidad y las desigualdades cuadráticas convexas de matrices. Cuando se tienen múltiples LMIs $F^{(i)} > 0$, $i = 1, \dots, p$, pueden ser expresadas como una única LMI $\text{diag}(F^{(1)}, \dots, F^{(p)}) > 0$.

Es habitual (como se comenta en la aplicación de esta técnica a los modelos borrosos de (Tanaka and Sugeno, 1992)) que las variables en la LMI sean matrices, como el caso de la desigualdad de Lyapunov para sistemas continuos ($\dot{x} = Ax$)

$$A^T P + P A > 0, \quad (4.11)$$

o para sistemas discretos ($x_{k+1} = Ax_k$)

$$A^T P A - P < 0, \quad (4.12)$$

donde $A \in R^{n \times n}$ es conocida y $P = P^T$ es la variable. Las desigualdades anteriores se pueden poner de la forma (4.10), por ejemplo para (4.12) tomando $F_0 = 0$, $F_i = -A^T P_i A + P_i$, donde P_1, \dots, P_m es una base para matrices simétricas $n \times n$.

Definición de Problema LMI. Dada una LMI $F(x) > 0$, el problema LMI es encontrar x^{fact} tal que $F(x^{fact}) > 0$ o determinar si la LMI no es factible. Este es un problema de factibilidad convexa.

4.3.1. Proceso iterativo para diseño de controladores

La aplicación del criterio de Lyapunov para resolver problemas LMI independientes en cada punto de operación como método de diseño de controladores cuando se dispone de modelos como (3.9) (eliminando el término independiente) o (3.12) para un sistema no lineal no es adecuada, ya que el hecho de que cada modelo local sea estable no asegura la estabilidad global para cualquier régimen de trabajo, tal y como muestra Tanaka en (Tanaka and Sugeno, 1992) con un contraejemplo.

Sin embargo se puede aplicar la condición de estabilidad simultánea de Lyapunov (Tanaka and Wang, 2001), que es exactamente un problema LMI: dadas $A_i \in R^{n \times n}$, con $i = 1, 2, \dots, c$, hay que encontrar P que satisfaga la LMI

$$P > 0, \quad A_i^T P A_i - P < 0, \quad i = 1, 2, \dots, c, \quad (4.13)$$

o determinar que P no existe.

Así pues, un posible proceso iterativo de diseño de controladores para un sistema no lineal queda así:

1. Diseñar un controlador para cada punto de operación atendiendo al rendimiento local,
2. Realizar un análisis de estabilidad planteado como problema LMI para comprobar que las condiciones de estabilidad se satisfacen (es factible encontrar P según criterio de Lyapunov),
3. Si el problema LMI no es factible, rediseñar los controladores.

Este método da buenos resultados en múltiples aplicaciones, como se comenta en (Tanaka and Wang, 2001).

4.3.2. Diseño directo por realimentación del estado

Existe una segunda estrategia de diseño de controladores basada en LMI que permite obtener directamente un controlador que asegura la estabilidad del sistema en bucle cerrado mediante la aplicación de realimentación del estado a cada modelo local (como (3.12) o como (3.9) eliminando su término independiente)

$$x_{k+1} = \nabla_{x_i} x_k + \nabla_{u_i} u_k, \quad (4.14)$$

donde, para simplificar la notación, se ha definido $\nabla_{x_i} = \nabla_{x_i} f(x_o, u_o)$ y $\nabla_{u_i} = \nabla_{u_i} f(x_o, u_o)$ para cada uno de los i puntos de operación (x_o, u_o) .

La condición suficiente de estabilidad del sistema en este caso (Carbonell, 1999) es la existencia de una matriz común definida positiva P y unas ganancias K_i tal que verifique para todos los modelos locales la desigualdad matricial lineal

$$(\nabla_{x_i} + \nabla_{u_i} K_i)^T P (\nabla_{x_i} + \nabla_{u_i} K_i) - P < 0 \quad (4.15)$$

verificando al mismo tiempo para todos los pares de modelos locales la desigualdad matricial lineal

$$\left(\frac{(\nabla_{x_i} + \nabla_{u_i} K_j) + (\nabla_{x_j} + \nabla_{u_j} K_i)}{2} \right)^T P \left(\frac{(\nabla_{x_i} + \nabla_{u_i} K_j) + (\nabla_{x_j} + \nabla_{u_j} K_i)}{2} \right) - P < 0 \quad (4.16)$$

Este problema se puede reescribir (Tanaka and Wang, 2001), definiendo $X = P^{-1}$ e $Y_i = K_i X$, como un problema LMI: dadas $\nabla_{x_i}, \nabla_{u_i}$ con $i = 1, 2, \dots, c$, hay que encontrar X y las c Y_i s que satisfagan la LMI

$$X > 0, \quad \nabla_{x_i} X + \nabla_{u_i} Y_i + X \nabla_{x_i}^T + Y_i \nabla_{u_i}^T < 0, \quad i = 1, 2, \dots, c, \quad (4.17)$$

o determinar que X no existe.

Si el problema es factible y se encuentran de forma simultánea X y las Y_i s, se deshacen los cambios de variable despejando las ganancias de realimentación K_i y el valor de P definida positiva. De este modo se dispone de una herramienta sencilla y útil para el diseño de sistemas de control para procesos no lineales en la forma de redes de modelos locales.

4.3.3. Estabilidad mediante LDIs

Como se ha comentado, si bien son directamente aplicables a las redes de modelos locales por velocidad, el término independiente hace que las técnicas de LMIs no se puedan aplicar en las redes de modelos locales en puntos de operación. Una opción interesante en este caso son las representaciones basadas en inclusiones diferenciales lineales (Boyd *et al.*, 1994) o LDIs (Linear Differential Inclusions), ya que pueden ser empleadas para establecer propiedades de sistemas no lineales.

Definición de LDI. Una inclusión diferencial lineal o LDI viene dada por

$$\dot{x} \in \Omega x, \quad x(0) = x_0, \quad (4.18)$$

donde Ω es un subconjunto de $R^{n \times n}$.

La LDIs se puede interpretar como una forma de describir una familia de sistemas lineales variables con el tiempo, ya que cada trayectoria (cada solución de las muchas posibles) de la LDI satisface

$$\dot{x} = A(t)x, \quad x(0) = x_0, \quad (4.19)$$

para algún $A : R_+ \rightarrow \Omega$. Igualmente, para cualquier $A : R_+ \rightarrow \Omega$, la solución de (4.19) es una trayectoria de la LDI (4.18). Una interpretación posible es que la LDI (4.18) puede ser descrita como un sistema incierto lineal variante con el tiempo, siendo el conjunto Ω la descripción de la incertidumbre de la matriz $A(t)$.

Considerando el sistema no lineal (3.1), supongamos que para cada x, u y t existe una matriz $G(x, u) \in \Omega$ tal que

$$\begin{bmatrix} f(x, u) \\ g(x, u) \end{bmatrix} = G(x, u) \begin{bmatrix} x \\ u \end{bmatrix} \quad (4.20)$$

donde $\Omega \subseteq R^{(n_x+n_u) \times (n_x+n_u)}$. De este modo cada trayectoria del sistema no lineal (3.1) es también una trayectoria de la LDI definida por Ω . Así pues, si se prueba que cualquier trayectoria de la LDI tiene alguna propiedad (por ejemplo, que converge a 0), se habrá probado que cualquier trayectoria del sistema no lineal (3.1) tiene esa propiedad.

Las condiciones que garantizan la existencia de G son $f(0, 0) = 0, g(0, 0) = 0$, y

$$\begin{bmatrix} \nabla_x f & \nabla_u f \\ \nabla_x g & \nabla_u g \end{bmatrix} \in \Omega, \quad \forall x, u. \quad (4.21)$$

De este modo y según la teoría de LDI, la aplicación de las técnicas de LMI para estudios de estabilidad de G cumpliendo las condiciones anteriores hará que el sistema original (3.1) herede sus propiedades. En el caso de las redes de modelos locales como (4.1), la LDI se transforma en una G_i (determinada según (4.21)) para cada uno de los c puntos (x_i, u_i) en los que se ha desarrollado el modelo local.

$$G_i = \begin{bmatrix} \nabla_x f(x_i, u_i) & \nabla_u f(x_i, u_i) \\ \nabla_x g(x_i, u_i) & \nabla_u g(x_i, u_i) \end{bmatrix} \in \Omega, \quad i \in 1, \dots, c. \quad (4.22)$$

Esta representación es muy útil para un posterior estudio con las técnicas de LMIs ya que, al derivar, desaparecen los términos independientes.

4.4. Conclusiones

En este capítulo se han presentado diferentes alternativas de control que aplicables a un conjunto de modelos lineales locales que describen un sistema no lineal: las técnicas de planificación de ganancia para familias de modelos LPV y las basadas en LMIs (en combinación o no de con las primeras).

Por su sencillez y filosofía de división del problema en subproblemas conservando la continuidad con los métodos lineales de diseño, son especialmente aplicables los controladores basados en modelos LPV en puntos de operación, aunque el término independiente de los mismos puede conllevar algunos problemas.

El diseño por velocidad mantiene las ventajas anteriores y sin restricciones al respecto de la técnica de diseño de controladores que se haya de emplear (los términos de offset han desaparecido). Además la planificación de ganancia basada en la velocidad proporciona un marco de análisis y diseño con un único tipo de linealización. Adicionalmente, el controlador no lineal que resulta es válido en todos los posibles puntos de funcionamiento de la planta, no solo en la vecindad de los puntos de equilibrio.

Por otra parte, las técnicas basadas en LMIs presentan la ventaja de proporcionar un test de estabilidad de los sistemas de control diseñados para los modelos locales (por ejemplo, mediante las técnicas de planificación de ganancia que de otro modo son algo arriesgadas) e incluso herramientas para determinar las ganancias en un esquema de realimentación del estado que cumplan con el criterio de estabilidad de Lyapunov o la no factibilidad de la resolución del problema. Sin embargo, dado que las funciones de validez o pertenencia de las redes de modelos locales por velocidad no se tienen en cuenta en el análisis (T-M.Guerra and Vermeiren, 2001), se puede llegar a resultados muy conservadores e incluso a no asegurar la estabilidad de sistemas que mediante simulaciones se comprueba que tienen un buen funcionamiento.

Finalmente, los términos independientes deben ser obviados en el caso de emplear redes de modelos locales en combinación con las LMIs y esto podría acarrear problemas, aunque puede ser resuelto considerando las condiciones adicionales planteadas por el análisis de sistemas no lineales mediante LDIs.

Capítulo 5

Modelado borroso

*Clearly,
the future is fuzzy.*
Lofti A. Zadeh.

En este capítulo se hace un repaso a los diferentes tipos de modelos borrosos basados en reglas con los que podemos trabajar a la hora de modelar un sistema, resaltando sus características de interpretabilidad y la adecuación de los modelos de Takagi-Sugeno para integrar modelos locales como los vistos en el capítulo 3.

5.1. Modelos borrosos basados en reglas

Basándose en la teoría de los conjuntos borrosos se han desarrollado diferentes métodos con el fin de modelar sistemas. De entre ellos cabría destacar los sistemas borrosos de regresión lineal (Yang and Ko, 1997), los sistemas borrosos basados en estructuras de celdas (Smith *et al.*, 1993) y, sobre todo, los sistemas borrosos basados en reglas debido a su interpretabilidad.

En los sistemas borrosos basados en reglas, la relación entre las variables se representa por medio de reglas si-entonces con la forma general:

Si antecedente **entonces** consecuente.

La proposición del antecedente es siempre del tipo "x es A" donde x es una variable lingüística y A es un término lingüístico constante. El grado de veracidad de la proposición depende de la similitud entre x y A. Dependiendo de la forma del consecuente podemos encontrar, principalmente, dos tipos de modelos borrosos basados en reglas, que serán brevemente revisados en este apartado:

- Modelos lingüísticos: en ellos tanto el antecedente como el consecuente son modelos borrosos.
- Modelo de Takagi-Sugeno: en los que el antecedente es una proposición borrosa y el consecuente una función real.

5.1.1. Modelos lingüísticos (Mamdani)

El modelo lingüístico borroso se introduce como una forma de capturar la información cualitativa disponible en la forma de reglas si-entonces:

$$R_i: \text{Si } x \text{ es } A_i \text{ entonces } y \text{ es } B_i, i = 1, 2, \dots, c.$$

En el caso de sistemas multivariable (MIMO) las reglas quedarían de la forma:

$$R_i : \text{Si } x_1 \text{ es } A_{i1} \text{ y } \dots \text{ y } x_s \text{ es } A_{is} \\ \text{entonces } y_1 \text{ es } B_{i1} \text{ y } \dots \text{ y } y_t \text{ es } B_{it}, i = 1, 2, \dots, c.$$

pero este tipo de reglas son equivalentes a t reglas en un modelo multivariable solo en la entrada (MISO) como:

$$R_i : \text{Si } x_1 \text{ es } A_{i1} \text{ y } \dots \text{ y } x_s \text{ es } A_{is} \\ \text{entonces } y \text{ es } B_i, i = 1, 2, \dots, c; j = 1, 2, \dots, s. \quad (5.1)$$

que será la notación utilizada, tan solo por simplicidad.

En la estructura anterior, x_j son las s variables lingüísticas de entrada y cada A_{ij} es un término lingüístico constante. De forma similar, y es la variable lingüística de salida y B_i es otro término lingüístico constante. Los valores x_j e y , y los términos lingüísticos A_{ij} y B_i son conjuntos borrosos definidos en sus respectivos dominios. Los primeros serán:

$$x_j \in X_j \subset \mathbf{R}^{pj} \\ y \in Y \subset \mathbf{R}^q$$

Las funciones de pertenencia de los conjuntos borrosos del antecedente y del consecuente son:

$$\mu(x_j) : X_j \rightarrow [0, 1] \\ \mu(y) : Y \rightarrow [0, 1]$$

Los conjuntos borrosos A_{ij} definen regiones borrosas en el espacio del antecedente para las que la proposición del consecuente es válida. Los términos lingüísticos A_{ij} y B_i son seleccionados de entre un conjunto de términos predefinidos como: pequeño, medio, etc. Si llamamos a estos conjuntos A_j y B respectivamente, tendremos:

$$A_{ij} \in A_j \\ B_i \in B$$

La base de reglas $R = \{R_i \mid i = 1, 2, \dots, c\}$ y los conjuntos A y B constituyen la *base de conocimiento* del modelo lingüístico. Para que el modelo lingüístico tenga utilidad necesitamos un algoritmo que permita calcular un valor de salida dado un valor de entrada, que es el llamado *mecanismo de inferencia*.

En (Wang, 1997), por ejemplo, se tratan en detalle estos dos temas (base de conocimiento y mecanismo de inferencia, así como la generalización de los modelos lingüísticos denominada *Modelos relacionales* en la que la relación entre el antecedente y el consecuente es también borrosa.

5.1.2. Modelos de Takagi-Sugeno

El modelo borroso de Takagi-Sugeno (TS) se diferencia del modelo lingüístico descrito anteriormente en que tiene funciones reales en su consecuente en lugar de una proposición borrosa:

$$R_i : \mathbf{Si } x_1 \text{ es } A_{i1} \mathbf{ y } \dots \mathbf{ y } x_s \text{ es } A_{is} \\ \mathbf{ entonces } y_i = f_i(x_1, x_2, \dots, x_s), i = 1, 2, \dots, c; \quad (5.2)$$

Un ejemplo sencillo y útil de este modelo es el caso en el que se toma la función del consecuente lineal en sus parámetros, llegando a reglas:

$$R_i : \mathbf{Si } x \text{ es } A_i \mathbf{ entonces } y_i = a_i^T x + b_i, i = 1, 2, \dots, c; \quad (5.3)$$

donde a_i es un vector de parámetros y b_i un escalar. A este modelo se le llama *modelo afín de Takagi-Sugeno* y proporciona el marco adecuado para la descomposición de un sistema no lineal en modelos lineales localmente válidos, muy apropiado para aplicar técnicas de control clásicas.

Si en lugar de considerar modelos afines en el consecuente se consideran modelos homogéneos ($b_i = 0, \forall i$) la intersección de todos ellos será en el origen, con lo que serán menos apropiados para la aproximación de funciones por no poder aproximar una función y su derivada al mismo tiempo.

Cuando los modelos afines del consecuente corresponden con linealizaciones en el entorno de puntos de operación (no necesariamente de equilibrio), un modelo borroso de este tipo puede ser apropiado para combinar las linealizaciones y llegar a integrarlas en redes de modelos locales o linealizaciones basadas en la velocidad, dada la interpretabilidad del antecedente y la correspondencia directa del consecuente con los modelos locales del capítulo 3. Sin embargo, pueden surgir problemas en la interpolación de los reglas (modelos) propuesta en la inferencia TS, tal y como se muestra en el apartado 5.2.

5.2. Inferencia de los modelos TS

La inferencia y desborrosificación de los modelos TS, una vez determinado el grado de cumplimiento $\omega_i(x)$ de cada regla y dados los datos x , se reduce a la expresión algebraica (Takagi and Sugeno, 1985):

$$f(x) = \frac{\sum_{i=1}^c \omega_i(x) f_i(x)}{\sum_{i=1}^c \omega_i(x)} \quad (5.4)$$

Para ver claramente la relación entre estos modelos y los del capítulo 3 se define el grado de cumplimiento normalizado de cada regla como:

$$\mu_i(x) = \frac{\omega_i(x)}{\sum_{j=1}^c \omega_j(x)} \quad (5.5)$$

con lo que la estructura global del consecuente con modelos afines (5.3) será:

$$y = \sum_{i=1}^c \mu_i(x) y_i(x) = \sum_{i=1}^c \mu_i(x) a_i^T x + \sum_{i=1}^c \mu_i(x) b_i \quad (5.6)$$

Este modelo lineal de parámetros variables es una forma de combinar modelos como los vistos en (3.9) para cada valor de linealización ϕ , ya que a_i^T correspondería a $\nabla_x f$ (la matriz Jacobiana de f respecto de x) y b_i seguiría la definición de (3.8).

La combinación de modelos propuesta en (5.4) para los consecuentes de las reglas tiene ventajas, por su simplicidad para el análisis y cálculo, pero tiene también algunas propiedades negativas que deben tenerse en cuenta para la aproximación de funciones como se muestra en el siguiente ejemplo.

Ejemplo 5.1. Sea el sistema dado por la función $y(x) = x^2$ de la cual se pretende hacer un modelo borroso TS. Mediante alguna técnica de identificación se han elegido los dos puntos de operación del sistema $(x_1, y_1) = (-0.5, 0.25)$ y $(x_2, y_2) = (0.5, 0.25)$, se han obtenido dos modelos locales dados por:

Punto de operación	Modelo local
$(-0.5, 0.25)$	$y_1(x) = -1.37x - 0.27$
$(0.5, 0.25)$	$y_2(x) = 1.37x - 0.27$

Tabla 5.1: Modelos locales en los puntos de operación.

y las correspondientes funciones de pertenencia trapezoidales para cada modelo:

$$A_1(x) = \begin{cases} 1 & x \leq -0,5 \\ -x + 0,5 & x \in [x_1, x_2] \\ 0 & x \geq 0,5 \end{cases} \quad (5.7)$$

$$A_2(x) = 1 - A_1(x)$$

Con todo lo anterior se genera un modelo borroso TS de dos reglas dado por:

$$\begin{aligned} R_1 &: \text{Si } x \text{ es } A_1 \text{ entonces } y_1 = -1,37x - 0,27 \\ R_2 &: \text{Si } x \text{ es } A_2 \text{ entonces } y_2 = 1,37x - 0,27 \end{aligned} \quad (5.8)$$

llevando, tras la inferencia (5.4), a una estructura global del consecuente para la estimación \hat{y} de y:

$$\hat{y}(x) = \mu_1(x)(-1,37x - 0,27) + \mu_2(x)(1,37x - 0,27) \quad (5.9)$$

ya que las funciones de pertenencia se han definido normalizadas, con lo que el cumplimiento $\omega_i(x)$ de la regla i ya normalizado y dado por $\mu_i(x)$ será igual a $A_i(x)$.

La figura 5.1 muestra los problemas de los modelos de TS así construidos: los modelos locales (línea continua) interpolados con el esquema propuesto por TS (5.4) llevan a un modelo (línea discontinua) cuya aproximación a la función inicial (línea continua con puntos) difiere bastante de ella y de lo que intuitivamente se espera (Díez *et al.*, 2001).

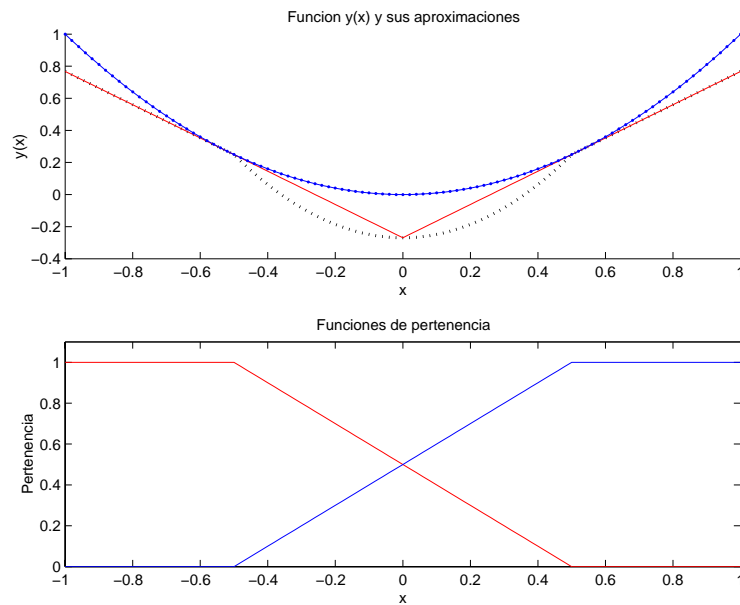


Figura 5.1: Modelos locales (línea continua), aproximación con modelo borroso TS (línea discontinua) y función original (línea continua con puntos) del ejemplo 5.1.

■

El ejemplo anterior pone de manifiesto con un contraejemplo que, aunque el marco de los modelos borrosos de TS parece apropiado para la integración de modelos locales en puntos de operación en redes de modelos locales, la inferencia de los modelos TS no siempre lleva a una interpolación apropiada de los mismos. Esta y otras cuestiones se discuten en mayor profundidad en el apartado 7.2.

5.3. Modelos borrosos de sistemas dinámicos

Existen diferentes opciones para representar un sistema dinámico con modelos borrosos, de entre las que cabría destacar los dos casos siguientes.

5.3.1. Representación con preprocesador

La idea es definir un preprocesador $v = F(u)$ de las variables de entrada u del sistema, donde F consiste en:

- Un filtro o un controlador derivativo o integral, si tratamos con sistemas continuos.
- Tener en cuenta variables retrasadas, si el sistema es discreto.

De esta forma la salida y el sistema dinámico queda completamente definido por:

- Un conjunto de reglas

$$R_{ij} : \text{Si } u \text{ es } U_i \text{ y } v \text{ es } V_j \text{ entonces } y \text{ es } Y_{ij}$$

- El preprocesador:

$$v = F(u)$$

5.3.2. Representación en el espacio de estados

La forma de escribir las reglas para el caso de un sistema continuo será, con el antecedente en función de la salida:

$$R_i : \text{Si } y \text{ es } Y_i \text{ entonces } y' = f_i(x, u), \quad i = 1, 2, \dots, c$$

o bien, si el antecedente está en función del vector de la entrada u :

$$R_i : \text{Si } u \text{ es } U_i \text{ entonces } y' = f_i(x, u), \quad i = 1, 2, \dots, c$$

Para un caso discreto, tomando el antecedente valores de la entrada y la salida, quedará la expresión:

$$R_{ij} : \text{Si } u(k) \text{ es } U_i \text{ y } y(k) \text{ es } Y_j \text{ entonces } y(k+1) \text{ es } Y_{ij} \quad (5.10)$$

que en el caso más genérico incluirá en el antecedente valores de las variables de las que se disponga en instantes (muestras) anteriores. En la figura 5.2 se puede observar el esquema de un modelo de caja negra realizado empleando un modelo dinámico borroso TS que emplea las últimas d muestras del vector de estado x y la salida.

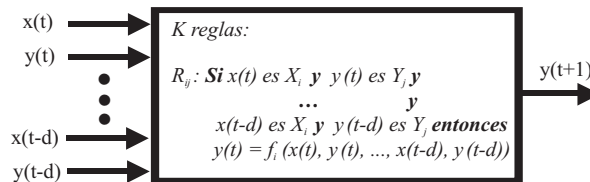


Figura 5.2: Esquema de un modelo dinámico borroso de caja negra.

5.4. Identificación de modelos borrosos basados en reglas

En el diseño de modelos borrosos hay que distinguir dos partes fundamentales: la estructura y los parámetros. La estructura determina la flexibilidad del modelo. Una vez seleccionada se estiman los valores de los parámetros para que el modelo se asemeje a los datos experimentales que existan.

5.4.1. Elementos a identificar

En los modelos borrosos, la selección de la estructura conlleva la elección de:

- *Variables de entrada y salida.* No siempre está claro que variables deben tomarse como entradas al modelo del sistema que se pretende identificar. En el caso de sistemas dinámicos se debe estimar, además, el orden del proceso. Generalmente se hace uso del conocimiento del comportamiento del sistema, aunque también es posible definir algún índice que nos permita comparar elecciones de forma automática, basándose en datos experimentales.
- *Estructura de las reglas.* Hay que elegir el tipo de modelo (lingüístico, relacional, Takagi-Sugeno) y la forma del antecedente de las reglas teniendo en cuenta el propósito del modelo y el conocimiento del que se dispone.
- *Número y tipo de funciones de pertenencia para cada variable.* Con esta elección se determina la granularidad (nivel de detalle) del modelo. En este caso, de nuevo, la clave en la elección es el propósito del modelo, aunque también se pueden usar métodos automáticos que, haciendo uso de datos experimentales, añadan o quiten funciones de pertenencia del modelo.
- *Mecanismo de inferencia, operadores conectivos, método de desborrosificación.* Aunque estas elecciones vienen restringidas por el tipo de modelo elegido, aún queda algo de libertad. Generalmente, si se pretende hacer uso de algún método de optimización del modelo usando datos experimentales, se prefieren los operadores diferenciables (producto, suma) a los habituales de mínimo o máximo.

Una vez la estructura ha sido fijada, los parámetros deben ser ajustados. Estos parámetros ajustables son: las reglas y las funciones de pertenencia de antecedente y consecuente.

5.4.2. Métodos de aprendizaje

Atendiendo a las características de la información disponible, las técnicas de optimización existentes para el diseño de modelos desde datos experimentales son tradicionalmente divididos en dos grandes grupos (Jang *et al.*, 1997) (comparar esta clasificación con la dada en el punto 2.5, más adecuada desde el punto de vista de la identificación borrosa). El primero de ellos es el que incluye los llamados *métodos de aprendizaje (optimización) supervisados* (basados o no en el método del gradiente), en los cuales se dispone de información sobre las respuestas deseadas. El otro gran grupo es el de *métodos de aprendizaje (optimización) no supervisados*, en los que no se tiene ninguna información sobre las respuestas. Por supuesto, el límite entre ambos métodos no está nada definido y existen métodos intermedios, como los de refuerzo, en los que se tiene solo una información parcial de las salidas deseadas (a un nivel de verdadero/falso).

Entre los métodos de aprendizaje supervisado se encuentran las conocidas técnicas de mínimos cuadrados (Wang, 1994) o propagación hacia atrás (backpropagation (Jang, 1993)). Entre los métodos de aprendizaje no supervisado se encuentran los que

utilizan redes neuronales (redes con aprendizaje competitivo (Jang *et al.*, 1997), redes autoorganizadas de Kohonen (Kohonen, 1990), aprendizaje del Hebbiano (Martinetz and Schulten, 1994)), las técnicas de agrupamiento y algunas mixtas como LVQ (Learning Vector Quantization (Chan *et al.*, 1996), donde se inicializa con agrupamiento y se continúa con aprendizaje competitivo).

En cuanto a la aplicabilidad de las diferentes técnicas de identificación a la construcción de un modelo borroso, como se comentó en el apartado 2.5, existen diferentes opciones.

Las *plantillas* (Wang, 1994) constituyen un método heurístico simple y eficaz para diseñar sistemas borrosos desde datos entrada-salida, que se desarrolla en cinco pasos.

Otra opción muy usada e inspirada en el método de propagación del error hacia atrás para redes neuronales (Jang, 1993), consiste en especificar la estructura del sistema (con ayuda de un experto) y ajustar sus parámetros con el método del *gradiente* de forma iterativa hasta que el error entre la salida estimada y la real para cada dato no supere error máximo predefinido.

El método de *mínimos cuadrados recursivo* también puede ser empleado para definir los parámetros del sistema borroso (Wang, 1994), de forma similar al caso anterior pero minimizando en este caso un error global (suma de todos los errores de predicción de los pares entrada-salida).

Los métodos descritos hasta el momento no permiten proponer un método sistemático de identificación, dado que hay numerosos parámetros (sobre todo en la determinación de la estructura, como el número de reglas) que exigen conocimiento del experto y habilidad del ingeniero. Sin embargo, la determinación, de forma directa, de reglas borrosas y funciones de pertenencia adecuadas para modelos como los vistos en el apartado 5.1 es difícil cuando la complejidad del sistema aumenta.

La aplicación de las *técnicas de agrupamiento*, por las que se ha optado en este trabajo como herramienta para determinar modelos borrosos de sistemas no lineales (Díez *et al.*, 2002b) y que se describirán en detalle en el capítulo 6, son una buena forma de evitar este problema, ya que permiten (al ser un método de aprendizaje no supervisado) construir un modelo borroso basado en reglas únicamente desde datos experimentales (incluyendo tanto la selección de la estructura como el ajuste de parámetros), con la ventaja de que la identificación podrían quedar automatizada. Para la parte de estimación de los parámetros de antecedente y consecuente del modelo borroso con la técnica propuesta, aunque se suelen aplicar técnicas de mínimos cuadrados que minimicen el error de predicción del modelo (Babuska, 1996), también es posible plantear problemas de optimización que sean resueltos por algoritmos genéticos, como en (Delgado *et al.*, 1997) y (Kim *et al.*, 1997).

En la bibliografía, es muy común no concretar el trabajo iniciado por métodos de agrupamiento borroso en un modelo borroso definitivo basado en reglas. Muchos trabajos que aparecen sobre modelado borroso de sistemas presentan una tendencia a desarrollar un modelo en dos pasos bien diferenciados: primero un modelo rápido y tosco, para pasar a continuación a un ajuste fino. Mientras que el agrupamiento se emplea para la determinación del modelo inicial, para el paso final del ajuste fino de parámetros, el rango es muy amplio: se aplican las técnicas habituales desarrolladas para redes neuronales si tenemos una estructura ANFIS (Adaptive Network-based-Fuzzy

Inference System, (Jang, 1993)), se puede realizar un ajuste fino con gradiente aplicado al modelo borroso (Kim *et al.*, 1997) o usar el aprendizaje competitivo (Dickerson and Kosko, 1996).

Otra opción, motivada por no existir ningún método que resuelva completamente el problema de identificación, consiste en desarrollar técnicas híbridas (*Neurofuzzy computing*, 1996) que exploten al máximo las posibilidades de cada uno de los métodos de modelado y optimización existentes, utilizando en cada caso la técnica más adecuada. Ejemplo de ello es la arquitectura ANFIS que combina las mejores características de los modelos borrosos y los neuronales. Sin embargo, datos con ruido o poca cantidad de datos pueden llevar a resultados en los que las salidas sean inconsistentes o sin significado alguno (Patterson, 1996).

Finalmente, la equivalencia funcional entre las redes neuronales de base radial (RBFN) hace que los métodos de identificación de RBFN se puedan usar para ANFIS (o modelos borrosos) y viceversa. Como ejemplo del caso contrario al que nos interesa, la aplicación de los métodos de agrupamiento para la determinación de los centros de las funciones de base radial se puede observar en (Sutanto *et al.*, 1997).

A pesar de la sinergia entre las técnicas de los modelos borrosos y la redes neuronales, no se ha encontrado en la bibliografía un aprovechamiento final de la interpretabilidad de la estructura ANFIS que se obtiene, aunque es esta su ventaja respecto a cualquier otra red neuronal. Una opción que se plantea es reconstruir, en un paso final, un modelo borroso basado en reglas para posibilitar la interpretación lingüística de los resultados y determinar la validez del modelo borroso definitivo.

5.5. Conclusiones

En este capítulo se han revisado tres tipos diferentes de modelos borrosos basados en reglas: lingüísticos, relacionales y de Takagi-Sugeno. La diferencia principal entre ellos es que mientras el modelo lingüístico (los modelos relacionales no son más que una extensión de éstos) tiene conjuntos borrosos tanto en antecedente como en consecuente, el modelo de Takagi-Sugeno tiene funciones reales en su consecuente. Los tres tipos de modelos pueden ser usados para modelar sistemas estáticos y dinámicos. Sin embargo, existen diferencias en cuanto a sus propiedades de aproximación, el tipo de conocimiento que pueden representar y los métodos de identificación que emplean.

Los modelos lingüísticos y relacionales son los más apropiados para aquellos casos en los que el conocimiento sea impreciso, heurístico y pueda ser expresado en forma de reglas si-entonces. Los modelos de Takagi-Sugeno proporcionan el marco adecuado para la descomposición de un sistema no lineal en modelos lineales localmente válidos (como los vistos en el capítulo 3), muy apropiado para aplicar técnicas de control clásicas. Es por ello que este trabajo se centrará en este tipo de modelos.

Finalmente, cabría destacar que existe una equivalencia total entre los modelos borrosos de los tres tipos y los modelos neuroborrosos vía la arquitectura ANFIS y su equivalencia funcional con las RBFN (Jang and Sun, 1993). Los modelos de Takagi-Sugeno pueden ser interpretados en clave lingüística-relacional si se toman como variables lingüísticas los parámetros de la función real del consecuente, como por ejemplo:

Si la salida es grande **entonces** la ganancia es grande **y** la cte. de tiempo es pequeña.

De esta forma, métodos para la determinación automática de modelos lingüísticos pasarán a ser válidos para los modelos de Takagi-Sugeno y viceversa e igualmente ocurrirá con los modelos neuroborrosos.

Capítulo 6

Técnicas de agrupamiento para identificación

Divide y vencerás.
Estrategia del Imperio Romano.

En este capítulo se revisan, de entre las técnicas de agrupamiento existentes (ver anexo C), las más apropiadas en la identificación de modelos borrosos como son los algoritmos de agrupamiento borroso. De entre los múltiples algoritmos de agrupamiento borroso existentes, se centrará esta revisión en aquellos con capacidades de detección de subespacios lineales dada su idoneidad para la identificación de modelos borrosos TS.

También se hace una exposición estructurada y completa de los distintos pasos a seguir para la identificación del modelo de un sistema no lineal mediante modelos borrosos, identificando su estructura y parámetros con las técnicas de agrupamiento.

6.1. Técnicas de agrupamiento

Las técnicas de agrupamiento (clustering) tienen como objetivo la clasificación de objetos de acuerdo con sus similitudes, así como la organización de datos en grupos. En concreto, la propiedad que interesa es su facilidad para dividir un conjunto de datos dados en varios grupos, de forma que la similitud entre los elementos dentro de un grupo es mayor que con los de otros grupos.

Por lo tanto, un *cluster* (grupo, clase) no es más que un grupo de objetos que son más similares entre ellos que a miembros de otros grupos. El término *similitud* debe ser entendido matemáticamente, mensurable de alguna forma bien definida. En los espacios métricos, la similitud suele ser definida mediante una norma o distancia. Esta distancia puede estar medida entre vectores de datos, o como distancia a unos vectores de datos prototipo de la clase. Estos prototipos no suelen ser conocidos a priori, sino que los propios algoritmos de agrupamiento los van definiendo a la vez que realizan la partición de los datos.

El aspecto geométrico de las clases puede ser muy variado en cuanto a formas, tamaños y densidades, como se muestra en la figura 6.1 para \mathbf{R}^2 . Las clases pueden ser lineales (6.1.a), esféricas (6.1.b) o incluso huecas como en (6.1.c) y (6.1.d). El mejor o

peor comportamiento de la mayoría de algoritmos de agrupamiento se ve fuertemente influenciado por la forma y densidad de la clase, así como por la relación espacial y distancia entre ellas.

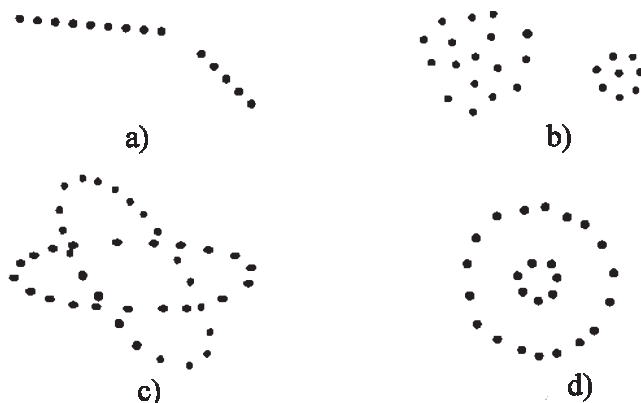


Figura 6.1: Diferentes posibilidades de clases en \mathbf{R}^2 .

Existe una gran diversidad de métodos de agrupamiento en la literatura. Como las clases pueden ser vistas como subconjuntos del conjunto de datos, una primera clasificación podría ser la que tuviese en cuenta si los subconjuntos son clásicos o borrosos. Los *métodos de agrupamiento clásicos* se basan en la teoría clásica de conjuntos y requieren que un objeto pertenezca o no a un conjunto. El método clásico de agrupamiento aplicado a un conjunto de datos Z significa dividir los datos en un número c determinado de subconjuntos mutuamente excluyentes. Los *métodos de agrupamiento borrosos* permiten a los objetos pertenecer a diferentes clases simultáneamente, con diferentes grados de pertenencia, siendo c el número total de subconjuntos borrosos definidos.

Otra posible clasificación de los métodos de agrupamiento podría hacerse con respecto al tipo de algoritmo que el método proponga. Los *métodos jerárquicos* utilizan técnicas de aglomeración y división de los subconjuntos creados, atendiendo a alguna medida de similitud. La mayor ventaja de estos métodos es su sencillez y rapidez. En los *métodos basados en la teoría de grafos* el conjunto de datos Z se toma como un conjunto de nodos. Los enlaces indican similitud entre nodos y existe la posibilidad de dividir el grafo en subgrafos. El tercer tipo de algoritmos de agrupamiento emplea una *función objetivo* para medir la adecuación de las particiones, minimizando una función objetivo global.

6.2. Particiones clásicas y borrosas

El objetivo del agrupamiento es dividir el conjunto de datos $Z = \{z_1, z_2, \dots, z_N\}$ (cada dato puede tener una dimensión n) en c clases. Supongamos que c es conocido (dado, por ejemplo, por la experiencia). Una *partición clásica* de Z se puede definir como la familia de subconjuntos $\{A_i \mid 1 \leq i \leq c\} \subset P(Z)$ ¹ con las propiedades (Bezdek, 1987):

¹ $P(Z)$ es el conjunto de todas las particiones de Z

$$A_i \cap A_j = \emptyset, \quad 1 \leq i \neq j \leq c \quad (6.1)$$

$$\emptyset \subset A_i \subset Z, \quad 1 \leq i \leq c \quad (6.2)$$

$$\bigcup_{i=1}^c A_i = Z, \quad (6.3)$$

La ecuación (6.3) indica que la unión de los subconjuntos A_i contiene todos los datos de Z . Los subconjuntos han de ser disjuntos como se indica en (6.1) y ninguno de ellos es el conjunto vacío ni contiene todos los datos de Z , tal como dice (6.2). En términos de *funciones de pertenencia (o de validez, o características)* las ecuaciones anteriores se pueden expresar como:

$$\mu_{A_i} \wedge \mu_{A_j} = 0, \quad 1 \leq i \neq j \leq c \quad (6.4)$$

$$0 < \mu_{A_i} < 1, \quad 1 \leq i \leq c \quad (6.5)$$

$$\bigvee_{i=1}^c \mu_{A_i} = 1, \quad (6.6)$$

Los valores 0 y 1 indican las funciones cero y uno, respectivamente, y μ_{A_i} es la función característica de A_i (con el fin de simplificar la notación se usará μ_i en lugar de μ_{A_i}) definida como:

$$\mu_{ik} \equiv \mu_i(z_k) = \begin{cases} 1, & z_k \in A_i \\ 0, & z_k \notin A_i \end{cases} \quad (6.7)$$

Una matriz $U = [\mu_{ik}]$ de dimensiones $c \times N$, representa una partición clásica si y solo si sus elementos satisfacen las condiciones:

$$\mu_{ik} \in \{0, 1\}, \quad 1 \leq i \leq c \quad 1 \leq k \leq N \quad (6.8)$$

$$\sum_{i=1}^c \mu_{ik} = 1, \quad 1 \leq k \leq N \quad (6.9)$$

$$0 < \sum_{k=1}^N \mu_{ik} < N, \quad 1 \leq i \leq c \quad (6.10)$$

La i -ésima fila de la matriz de partición U contiene los valores de la función característica del subconjunto A_i de Z . Todo lo visto con respecto a la partición clásica puede quedar resumido en la siguiente definición.

Definición 6.1. Espacio de particiones clásicas. Sea $Z = \{z_1, z_2, \dots, z_N\}$ un conjunto finito de datos y $2 \leq c \leq N$ un entero. El *espacio de particiones clásicas de Z* es el conjunto:

$$M_{hc} = \left\{ U \in R^{c \times N} \mid \mu_{ik} \in \{0, 1\}, \forall i, k; \sum_{i=1}^c \mu_{ik} = 1, \forall k; 0 < \sum_{k=1}^N \mu_{ik} < N, \forall i \right\} \quad (6.11)$$

La generalización de la partición clásica a la partición borrosa (Ruspini, 1970) es tan sencilla como permitir que μ_{ik} tome valores reales en el intervalo $[0,1]$. Las condiciones que debe cumplir una matriz de partición borrosa, análogas a (6.8), (6.9) y (6.10) son:

$$\mu_{ik} \in [0, 1], \quad 1 \leq i \leq c \quad 1 \leq k \leq N \quad (6.12)$$

$$\sum_{i=1}^c \mu_{ik} = 1, \quad 1 \leq k \leq N \quad (6.13)$$

$$0 < \sum_{k=1}^N \mu_{ik} < N, \quad 1 \leq i \leq c \quad (6.14)$$

En este caso se podría resumir en la siguiente definición lo visto con respecto a la partición borrosa.

Definición 6.2. Espacio de particiones borrosas. Sea $Z = \{z_1, z_2, \dots, z_N\}$ un conjunto finito de datos y $2 \leq c \leq N$ un entero. El *espacio de particiones borrosas de Z* es el conjunto:

$$M_{fc} = \left\{ U \in R^{c \times N} \mid \mu_{ik} \in [0, 1], \forall i, k; \sum_{i=1}^c \mu_{ik} = 1, \forall k; 0 < \sum_{k=1}^N \mu_{ik} < N, \forall i \right\} \quad (6.15)$$

La i -ésima fila de la matriz de partición borrosa U contiene los valores de la función de pertenencia del subconjunto borroso A_i de Z . La ecuación (6.13) hace que la suma de cada columna sea 1, con lo que la pertenencia total de cada z_k es 1. La distribución de las funciones de pertenencia entre los c subconjuntos borrosos no está restringida de ningún modo.

Existe la posibilidad de relajar la condición (6.13) no obligando a que la suma de funciones de pertenencia sea la unidad para cada objeto. En (Krishnapuram and Keller, 1993) se propone sustituir la condición (6.13) por:

$$\exists i, \quad \mu_{ik} > 0, \forall k \quad (6.16)$$

con lo que la partición borrosa pasaría a ser una *partición posibilista* cuya definición es:

Definición 6.3. Espacio de particiones posibilísticas. Sea $Z = \{z_1, z_2, \dots, z_N\}$ un conjunto finito de datos y $2 \leq c \leq N$ un entero. El *espacio de particiones posibilísticas de Z* es el conjunto:

$$M_{pc} = \left\{ U \in R^{c \times N} \mid \mu_{ik} \in [0, 1], \forall i, k; \exists i, \quad \mu_{ik} > 0, \forall k; 0 < \sum_{k=1}^N \mu_{ik} < N, \forall i \right\} \quad (6.17)$$

La i -ésima fila de la matriz de partición posibilística U contiene los valores de la función de pertenencia del subconjunto borroso A_i de Z . La ecuación (6.16) hace que la suma de cada columna sea entre 0 y 1, pero ahora la pertenencia total de cada z_k no tiene por que ser 1.

6.3. Algoritmo de agrupamiento borroso fuzzy c-means

En los algoritmos de agrupamiento clásico cada dato debe ser asignado a una clase. El análisis con agrupamiento borroso relaja este requerimiento permitiendo pertenencias graduales y ofreciendo la oportunidad de trabajar con datos que pertenecen a más de una clase al mismo tiempo. La mayoría de las técnicas analíticas de agrupamiento borroso se basan en la optimización de la función objetivo c-means (Bezdek, 1973) o alguna modificación de ésta.

6.3.1. Función objetivo c-means

La función objetivo base de una gran familia de algoritmos de agrupamiento borroso es la siguiente:

$$J(Z; U, C) = \sum_{i=1}^c \sum_{k=1}^N (\mu_{ik})^m \|z_k - c_i\|_B^2 \quad (6.18)$$

donde

$$Z = \{z_1, z_2, \dots, z_N\} \quad (6.19)$$

son los datos que deben ser clasificados,

$$U = [\mu_{ik}] \in M_{fc} \quad (6.20)$$

es una matriz partición borrosa (6.15) de Z ,

$$C = [c_1, c_2, \dots, c_c], c_i \in \mathbf{R}^n \quad (6.21)$$

es el vector de centros (centroides, prototipos) a determinar,

$$D_{ik B}^2 = \|z_k - c_i\|_B^2 = (z_k - c_i)^T B (z_k - c_i) \quad (6.22)$$

es una norma, y

$$m \in [1, \infty) \quad (6.23)$$

es un exponente que determina la "borrosidad" de las clases resultantes. El valor de la función de coste (6.18) es una medida ponderada del error cuadrático que se comete al representar las c clases por los prototipos c_i .

6.3.2. Algoritmo fuzzy c-means (FCM)

La minimización de la función objetivo (6.18) es un problema de optimización no lineal que puede ser resuelto de muchas formas, pero la más habitual es la conocida como algoritmo fuzzy ISODATA o FCM (Fuzzy C-Means, no confundir con el FCM empleado en la bibliografía de redes neuronales para referirse a Fuzzy Cognitive Map (Kosko, 1986) de Kosko).

Los pasos que debe seguir el algoritmo c-means, dado un conjunto de datos como (6.19), y seleccionados un número $c \in [1, N]$ de clases, un exponente de ponderación $m > 1$, una tolerancia $\varepsilon > 0$ (normalmente entre 0.01 y 0.001) y una norma B, son los siguientes:

- Inicializar la matriz de partición borrosa U con valores aleatorios $U^{(0)}$, haciendo $l=1$ (índice de iteración).
- Calcular los centros de las clases:

$$c_i^{(l)} = \frac{\sum_{k=1}^N (\mu_{ik}^{(l-1)})^m z_k}{\sum_{k=1}^N (\mu_{ik}^{(l-1)})^m}, \quad 1 \leq i \leq c$$

- Calcular las distancias:

$$D_{ikB}^2 = (z_k - c_i^{(l)})^T B (z_k - c_i^{(l)}), \quad 1 \leq i \leq c \quad 1 \leq k \leq N$$

- Recalcular la matriz de partición:
 - si $D_{ikB} > 0$ para todo $1 \leq i \leq c, 1 \leq k \leq N$,

$$\mu_{ik}^{(l)} = \frac{1}{\sum_{j=1}^c (D_{ikB} / D_{jkB})^{2/(m-1)}}$$

- en cualquier otro caso:

$$\mu_{ik}^{(l)} = 0 \text{ si } D_{ikB} > 0, \text{ y}$$

$$\mu_{ik}^{(l)} \in [0, 1] \text{ con } \sum_{i=1}^c \mu_{ik}^{(l)} = 1 \text{ para el resto.}$$

- Si $\|U^{(l)} - U^{(l-1)}\| < \varepsilon$, parar. Si no, volver al segundo paso del algoritmo.

6.3.3. Normas

La forma de las clases se determina por la elección de la matriz B en (6.22). Una elección habitual es tomar $B = I$, lo que lleva a la norma Euclidea:

$$D_{ik}^2 = (z_k - c_i)^T (z_k - c_i) \tag{6.24}$$

Esta elección induce clases hiperesféricas (figura 6.2.a). B también puede ser una matriz diagonal de dimensiones $n \times n$ (n es la dimensión de los elementos de Z) que toma diferentes varianzas en cada una de las direcciones de los ejes coordenados de Z :

$$B = \begin{bmatrix} (1/\sigma_1)^2 & 0 & \dots & 0 \\ 0 & (1/\sigma_2)^2 & \dots & 0 \\ \dots & \dots & \dots & \dots \\ 0 & 0 & \dots & (1/\sigma_n)^2 \end{bmatrix} \quad (6.25)$$

Esta matriz induce una norma diagonal en \mathbf{R}^n y una forma hiperelipsoidal en las clases (figura 6.2.b). La última opción es definir B como la inversa de la matriz de covarianzas de las muestras de Z (\bar{z} es la media de los datos):

$$B = R^{-1} \quad \text{con} \quad R = \frac{1}{N} \sum_{k=1}^N (z_k - \bar{z})(z_k - \bar{z})^T \quad (6.26)$$

En este caso se induce la norma de Mahalanobis (Bezdek, 1987), que también da lugar a clases hiperelipsoidales, pero con una orientación arbitraria de sus ejes (figura 6.2.c).

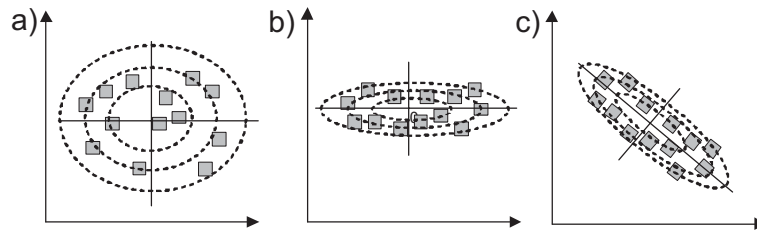


Figura 6.2: Normas usadas en el agrupamiento borroso.

Algunas normas son sensibles a los rangos numéricos de los datos, con lo que una normalización previa puede ser una buena opción. La norma Euclídea, por ejemplo, asigna un mayor peso a aquellas características con rangos amplios que a las de rangos estrechos. Una simple elección de unidades (metros o centímetros) puede influenciar negativamente los resultados del agrupamiento. La literatura de reconocimiento de patrones sugiere normalizar los datos previamente al agrupamiento (Jain and Dubes, 1988). La normalización más simple es la dada por la resta de la media de la característica \bar{z}_j :

$$z_{jk} = z_{jk}^* - \bar{z}_j \quad (6.27)$$

que hace los valores de la característica invariantes a desplazamientos rígidos de las coordenadas. El asterisco denota los datos "sin escalar". Otro tipo de normalización traslada y escala los ejes de forma que todas las características tengan media nula y varianza unitaria:

$$z_{jk} = \frac{z_{jk}^* - \bar{z}_j}{\sigma_j} \quad (6.28)$$

Sin embargo, la normalización no siempre es deseable y puede alterar la separación entre las clases y afectar negativamente los resultados del agrupamiento. Afortunadamente, los algoritmos basados en medidas de distancia adaptativas (apartado C.3) no son sensibles al escalado de los datos, ya que la adaptación de la medida de distancia compensa automáticamente las diferencias de escala.

6.3.4. Extensiones al algoritmo FCM

En algunas aplicaciones el algoritmo FCM no da unos buenos resultados, principalmente por identificar todas las clases con dimensiones y orientación similar y una forma hiperesférica o hiperelipsoidal (Bezdek, 1987).

Sin embargo, existen numerosas extensiones y modificaciones al algoritmo básico fuzzy c-means (ver anexo C) que adaptan (6.22) con el fin de detectar clases de diferentes formas geométricas y tamaños en un mismo conjunto de datos. Algunas de ellas pueden hacer que sus resultados sean más adecuados para otros fines, como por ejemplo el modelado y control por modelos locales. Los métodos de posible interés para aplicaciones en las que la detección de espacios lineales sea de importancia, pueden ser clasificados en dos grandes grupos:

- Algoritmos que utilizan una medida de la distancia adaptativa, tomando una norma diferente para cada clase.
- Algoritmos basados en prototipos lineales, dejando la norma constante y haciendo los prototipos variables.

De entre los algoritmos con distancia adaptativa, cabe destacar el de Gustafson-Kessel (GK, (Gustafson and Kessel, 1979)), dado que las clases hiperelipsoidales que busca detectan de forma bastante correcta los comportamientos quasi-lineales de los diversos regímenes de funcionamiento que pueden existir en un conjunto de datos (apartado 6.4). Otros algoritmos adaptativos con los que también se obtienen buenos resultados (punto C.3 del anexo C) son el Algoritmo de estimación de la máxima probabilidad (FMLE, (Bezdek and Dunn, 1975)) y el Algoritmo de Gath y Geva (GG, (Gath and Geva, 1989)).

Los algoritmos basados en prototipos lineales mantienen la norma constante pero definen prototipos lineales r -dimensionales ($0 \leq r \leq n-1$) en subespacios del espacio de datos. El algoritmo más interesante en el ámbito de esta Tesis es el fuzzy c-regression models (FCRM, (Hathaway and Bezdek, 1991)) (apartado 6.5), aunque el algoritmo fuzzy c-varieties (FCV, (Bezdek, 1987)) también puede ser interesante en algunos casos (apartado C.4 del anexo C).

Los prototipos empleados en los algoritmos también pueden tomar formas no lineales en subespacios de los datos (búsqueda de clusters huecos (Klawonn *et al.*, 1997), punto C.6 del anexo C), pero no son de aplicabilidad en el marco de esta Tesis.

6.4. Algoritmo de Gustafson-Kessel (GK)

Este algoritmo extiende el algoritmo básico fuzzy c-means eligiendo una norma diferente B_i para cada clase, con lo que (6.22) pasa a ser (Gustafson and Kessel, 1979):

$$D_{ik}^2 B_i = \|z_k - c_i\|_{B_i}^2 = (z_k - c_i)^T B_i (z_k - c_i) \quad (6.29)$$

Estas matrices son ahora tomadas como posibles variables para la optimización de la función (6.18), con lo que se adaptará la norma a cada clase según sus características. Sea $B = \{B_1, B_2, \dots, B_c\}$ el vector que contiene las c normas. La nueva función a minimizar será:

$$J(Z; U, C, B) = \sum_{i=1}^c \sum_{k=1}^N (\mu_{ik})^m D_{ik}^2 B_i \quad (6.30)$$

cumpliendo (6.20), (6.21) y (6.23). Para obtener una solución viable B_i debe ser limitada de alguna forma. La forma más habitual es fijar el determinante de B_i , lo que es equivalente a optimizar la forma de la clase manteniendo su volumen constante:

$$|B_i| = \rho_i, \quad \rho > 0 \quad (6.31)$$

con ρ_i constante para cada clase. La expresión que se obtiene para la minimización de B_i con la restricción (6.31) de volumen constante es:

$$B_i = [\rho_i \det(F_i)]^{1/n} F_i^{-1} \quad (6.32)$$

siendo F_i la matriz de covarianzas de clase i definida por:

$$F_i = \frac{\sum_{k=1}^N (\mu_{ik})^m (z_k - c_i)(z_k - c_i)^T}{\sum_{k=1}^N (\mu_{ik})^m} \quad (6.33)$$

Dado un conjunto de datos como (6.19), y seleccionados un número $c \in [1, N]$ de clases, un exponente de ponderación $m > 1$, un valor para cada determinante ρ_i ($\rho_i = 1$ si no hay información previa en ese sentido) y una tolerancia $\varepsilon > 0$ para la terminación, los pasos del algoritmo GK completo son los siguientes:

- Inicializar la matriz de partición borrosa U con valores aleatorios $U^{(0)}$, haciendo $l=1$ (índice de iteración).
- Calcular los centros de las clases:

$$c_i^{(l)} = \frac{\sum_{k=1}^N (\mu_{ik}^{(l-1)})^m z_k}{\sum_{k=1}^N (\mu_{ik}^{(l-1)})^m}, \quad 1 \leq i \leq c$$

- Calcular la matriz de covarianzas de cada clase:

$$F_i^{(l)} = \frac{\sum_{k=1}^N (\mu_{ik}^{(l-1)})^m (z_k - c_i^{(l)})(z_k - c_i^{(l)})^T}{\sum_{k=1}^N (\mu_{ik}^{(l-1)})^m}, \quad 1 \leq i \leq c$$

- Calcular las distancias:

$$B_i = (z_k - c_i^{(l)})^T \left[[\rho_i \det(F_i)]^{1/n} F_i^{-1} \right] (z_k - c_i^{(l)}), \quad 1 \leq i \leq c \quad 1 \leq k \leq N$$

- Recalcular la matriz de partición:

- si $D_{ikB_i} > 0$ para todo $1 \leq i \leq c, 1 \leq k \leq N$,

$$\mu_{ik}^{(l)} = \frac{1}{\sum_{j=1}^c (D_{ikB_i} / D_{jkB_i})^{2/(m-1)}}$$

- en cualquier otro caso:

$$\mu_{ik}^{(l)} = 0 \text{ si } D_{ikB_i} > 0, \text{ y}$$

$$\mu_{ik}^{(l)} \in [0,1] \text{ con } \sum_{i=1}^c \mu_{ik}^{(l)} = 1 \text{ para el resto.}$$

- Si $\|U^{(l)} - U^{(l-1)}\| < \varepsilon$, parar. Si no, volver al segundo paso del algoritmo.

6.5. Algoritmo fuzzy c-regression models (FCRM)

Este algoritmo (Hathaway and Bezdek, 1991) ajusta los parámetros del modelo de regresión:

$$y_k = f_i(x_k; \beta_i) \tag{6.34}$$

con las funciones f_i parametrizadas por $\beta_i \in \mathbf{R}^{p_i}$. El grado de pertenencia $\mu_{ik} \in U$ se interpreta en este caso como la cercanía existente entre el valor predicho por el modelo (6.34) e y_k . El error de predicción suele calcularse como:

$$E_{ik}(\beta_i) = (y_k - f_i(x_k; \beta_i))^2 \tag{6.35}$$

La función objetivo a minimizar con el método que se presenta, definidas $U \in M_{fc}$ y $(\beta_1, \dots, \beta_c) \in \mathbf{R}^{p_1} \times \mathbf{R}^{p_2} \times \dots \times \mathbf{R}^{p_c}$ es:

$$E_m(U, \{\beta_i\}) = \sum_{i=1}^c \sum_{k=1}^N (\mu_{ik})^m E_{ik}(\beta_i) \tag{6.36}$$

Una posibilidad para minimizar (6.36) se presenta en (Hathaway and Bezdek, 1991) y queda resumida en el siguiente algoritmo. Dado un conjunto de datos $Z = \{(x_1, y_1), \dots, (x_N, y_N)\}$, seleccionados un número $c \in [1, N]$ de clases, un exponente de ponderación $m > 1$ y una tolerancia $\varepsilon > 0$ para la terminación:

- Inicializar la matriz de partición borrosa U con valores aleatorios $U^{(0)}$, haciendo $l=1$ (índice de iteración).
- Calcular valores para los parámetros β_i^* del modelo que minimicen la función $E_m(U^{(l)}, \{\beta_i\})$, empleando mínimos cuadrados.
- Recalcular la matriz de partición para todo $1 \leq i \leq c, 1 \leq k \leq N$,

$$\mu_{ik}^{(l)} = \frac{1}{\sum_{j=1}^c (E_{ik}/E_{jk})^{2/(m-1)}}$$

si $E_{ik} = 0$ para algún $i = s$, hacer $\mu_{sk} = 1$ y $\mu_{ik} = 0$ para $\forall i \neq s$.

- Si $\|U^{(l)} - U^{(l-1)}\| < \varepsilon$, parar. Si no, volver al segundo paso del algoritmo.

6.6. Algoritmos con prototipos mixtos

Existe un tercer tipo de algoritmos, que podrían ser llamados algoritmos con prototipos mixtos, que intentan superar los problemas de FCM mediante la combinación de los algoritmos con norma adaptativa y los que emplean prototipos lineales. Estos algoritmos son el algoritmo fuzzy c -elliptotypes (FCE, (Bezdek, 1987)) y el algoritmo adaptive fuzzy c -regression models (AFCR, (Ryoke and Y.Nakamori, 1996)).

6.6.1. Algoritmo fuzzy c -elliptotypes (FCE)

(Davé, 1990) calcula un mínimo local de una función objetivo basada en FCM (6.18) en la que se sustituye D_{ik}^2 (6.22) por una combinación convexa de distancias.

Con este algoritmo se pretenden superar algunos de los problemas del FCV. Para ello, se fuerza a cada clase a tener una centro de gravedad cdg_i y se mide la distancia como combinación de las distancias de FCM y FCV:

$$D_{eik} = \alpha D_{ik} + (1 - \alpha) D_{rik} \quad (6.37)$$

con $\alpha \in [0,1]$. Sustituyendo (6.24) y (C.34) en (6.37) tenemos:

$$D_{eik} = (z_k - cdg_i)^T (z_k - cdg_i) - \alpha \sum_{j=1}^r \langle z_k - cdg_i, s_{ij} \rangle^2 \quad (6.38)$$

El problema en este algoritmo es la elección del α correcto para cada clase.

6.6.2. Adaptive fuzzy c -regression models (AFCR)

El algoritmo AFCR (Ryoke and Y.Nakamori, 1996) proporciona un tratamiento similar a FCRM del que hace FCE con FCV. En este caso el criterio queda:

$$J(Z; U, \beta_1, \dots, \beta_c, c_1, \dots, c_c) = \sum_{i=1}^c \sum_{k=1}^N (\mu_{ik})^m D_{aik} \quad (6.39)$$

y se toma la distancia como combinación de las distancias de FCRM (E_{ik}) y FCM (D_{ik}):

$$D_{aik} = \alpha_i E_{ik}(\beta_i) + (1 - \alpha_i)\eta D_{ik} \quad (6.40)$$

con $\alpha_i \in [0,1]$. El primer término proporciona el mismo criterio que FCRM y el segundo incrementa la capacidad de partición en las variables ya que tiene en cuenta la distancia de los datos al prototipo de las clases. La elección del α_i se hace de forma dinámica en este algoritmo y se acerca a 1 según la estructura descubierta en la clase i es más lineal. Su determinación se basa en

$$\alpha_i = 1 - \frac{\min_l \{\lambda_{il}\}}{\max_l \{\lambda_{il}\}}, \quad i = 1, 2, \dots, c \quad (6.41)$$

con λ_{il} los autovalores de la matriz de covarianzas definida en (6.33) para GK. Este procedimiento es muy importante para evitar mínimos locales no deseados pero, dependiendo de la complejidad de los datos, se podría pensar en otros índices (más simples) que variaran de 0 a 1 según las clases se hiciesen más lineales.

De este modo se conjuga en un solo algoritmo ventajas de FCRM, FCM y GK. El parámetro η sirve de balance entre los términos cuando su tamaño medio es muy diferente y no hay ningún estudio respecto a su determinación.

6.7. Proceso de identificación de modelos borrosos

En este apartado se hace una exposición estructurada y completa de los distintos pasos a seguir para la identificación del modelo de una sistema no lineal mediante modelos borrosos, identificando su estructura y parámetros con técnicas de agrupamiento.

El proceso de identificación de modelos borrosos con las técnicas vistas en el punto 2.5 es iterativo por naturaleza (Díez and Navarro, 1999) consistiendo en la repetición, para diferentes elecciones de diversos parámetros, de los pasos que se muestran en la figura 6.3 con el fin de obtener todos los elementos del modelo descritos en el apartado 5.4.

¿En qué partes del esquema 6.3 pueden ser de ayuda las técnicas de agrupamiento? Obviamente, se podría aplicar en el paso de agrupación de datos con el fin de determinar diferentes comportamientos del sistema en estudio y dar una idea de cuántas reglas son necesarias. Sin embargo, la aplicación del agrupamiento borroso proporciona ventajas adicionales en la identificación borrosa ya que la pertenencia de los datos experimentales a las clases pueden servir para obtener las funciones de validez.

Con todo lo anterior puede parecer que la construcción de un modelo borroso (aunque sea aproximado) desde datos experimentales sea una tarea simple y que puede ser resuelta de forma automática. Sin embargo, tareas como la selección de entradas, la simplificación y validación de modelos no están resueltas. Además, una vez elegido el tipo de modelo borroso y el algoritmo de agrupamiento apropiados al sistema a identificar, se han de definir un buen número de parámetros como: el número de clases c , el parámetro de borrosidad m , la tolerancia ε , la norma B a emplear, etc.

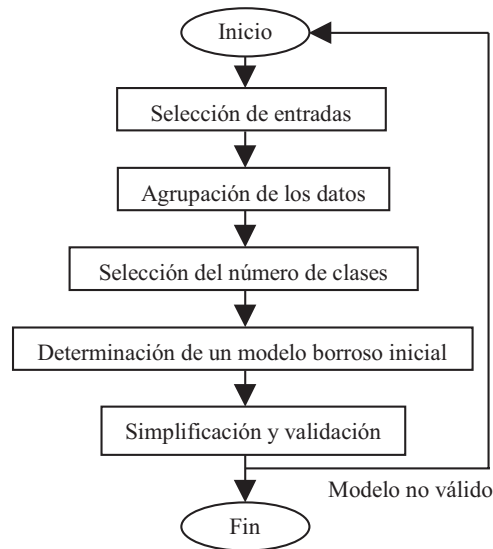


Figura 6.3: Esquema sistematizado de identificación de modelos borrosos.

El primer intento de sistematización de la identificación de modelos borrosos mediante técnicas de agrupamiento (Sugeno and Yasukawa, 1993) (que ha sido utilizado posteriormente por la mayoría de autores) es ampliada en dos pasos en (Babuska, 1996): la simplificación del modelo y la validación del mismo. Aunque no explicitados en (Sugeno and Yasukawa, 1993), estos dos nuevos pasos parecen supuestos en él, ya que su fin era el de conseguir un modelo cualitativo y altamente interpretable, cuya lectura por el experto sería de por sí una forma de validación del modelo. A continuación se detallan cada uno de los pasos en que consiste la identificación del modelo tal y como proponen (Sugeno and Yasukawa, 1993), (Babuska, 1996) y (Emami *et al.*, 1998).

6.7.1. Diseño de experimentos

Este paso inicial es importante en cualquier método de identificación. Los métodos de excitación mediante señales binarias pseudoaleatorias no son apropiados para los sistemas no lineales que se tratan en este trabajo, como los sistemas no lineales (Godfrey, 1990).

Según (Ljung, 1999) lo conveniente en este caso es excitar al sistema a identificar por un rango completo de señales tanto en amplitud como en frecuencia, por lo que las señales con dos valores de amplitud (binarias) no son apropiadas. Una señal de excitación con señales senoidales variables en amplitud y frecuencia o escalones de amplitud y anchura aleatoria pueden ser una buena solución. A la señal de excitación elegida se le suele añadir un ruido blanco de pequeña amplitud para garantizar la adecuada excitación de la dinámica del proceso. En el diseño de experimentos nunca se debe olvidar la elección de un periodo de muestreo adecuado y el tiempo que debe durar el experimento, así como las posibles ventajas que ofrece la normalización de los datos (apartado 6.3.3).

6.7.2. Selección de la estructura

Además de la forma de tratar la dinámica del proceso y la granularidad a emplear en el modelo, en este paso se identifican cuales son las variables de entrada y salida relevantes para el problema de modelado. Aunque el experto en el sistema puede dar las claves en este paso, se puede automatizar la elección de las variables comparando varias propuestas desde el punto de vista de algún índice (ver apartado D.2).

Su resolución no es directa sino que requiere técnicas iterativas, lo que lleva a que sus valores definitivos no sean determinados hasta el último paso de la construcción del modelo borroso, con lo que todo el proceso de obtención del modelo se convierte en un gran problema de iteración, con sus múltiples partes relacionadas entre sí y sin ser posible la determinación aislada y precisa de los parámetros referentes a la estructura.

Elección de las variables de entrada

Aunque muchos de los métodos revisados en la bibliografía ((Nakamori and Ryo-ke, 1994), (Klawonn and Kruse, 1995*b*), (Hong and Lee, 1996), (Genther and Glesner, 1997), (Klawonn and Keller, 1997), (Klawonn and Kruse, 1997), (Delgado *et al.*, 1997) y (Kim *et al.*, 1997)) suponen, para la construcción del modelo borroso de un sistema, que las entradas y salidas son conocidas, generalmente no está claro en los sistemas multivariable qué variables han de ser elegidas como entradas al sistema (Sugeno and Yasukawa, 1993). El conocimiento de los expertos es fundamental en este tema, aunque los métodos estadísticos de correlación también pueden jugar un papel importante.

En las soluciones heurísticas a este problema (apartado D.2) se parte de un número N de posibles candidatos a ser entradas al sistema y se estudia el efecto de las distintas combinaciones de variables de entrada sobre la salida. En general, para un conjunto Z de posibles variables de entrada z_1, z_2, \dots, z_N , el número total de casos a estudiar es $2^N - 1$. Con el fin de no tener que probar todas las posibles combinaciones de entradas, se seleccionan algunas entradas de entre las candidatas y se van incrementando el número de entradas una a una, teniendo en cuenta algún criterio. En (Emami *et al.*, 1998) se emplea un *índice de no-significatividad* de las variables. Otra opción con la que resolver el problema de la selección de entradas, de mayor carga computacional pero más precisa, es el llamado *criterio de regularidad* que se da en (Sugeno and Yasukawa, 1993).

Representación de la dinámica del proceso

En este paso se transforma el problema de identificación de un sistema dinámico en un problema de regresión estática. La idea es realizar alguna transformación desde el dominio temporal al de las variables de estado, de manera que éstas determinen el estado del sistema sin necesidad del tiempo. Los métodos usualmente empleados en el modelado dinámico mediante sistemas borrosos fueron vistos en el apartado 5.3.

En el caso de tratar con sistemas discretos, una solución sencilla y que da buenos resultados consiste en tratar todas las muestras disponibles de una misma variable u como variables independientes ($u(t), u(t-1), u(t-2), \dots$), con lo que todo sistema se convierte en estático. A estas "nuevas" variables se les aplica el método que se desee de selección de variables.

Granularidad del modelo borroso

Este parámetro está relacionado tanto con el número de reglas como con el número de términos lingüísticos para cada variable. Es el paso que en la literatura se le llama normalmente *identificación de la estructura*. El valor que inicializa el proceso de determinación de estos valores, si se usan métodos de agrupamiento, es el número de clases inicial (ver punto 6.7.4). Con los métodos de simplificación de modelos borrosos que se verán en el apartado 6.7.6 se disminuirá, si es posible, el número de conjuntos borrosos en las reglas número de clases final mediante medidas de similaridad y el número de reglas con técnicas de detección de redundancia.

6.7.3. Agrupación de los datos

Una vez se tienen identificadas cuales son las entradas y salidas de importancia para el sistema, se realiza una partición de los datos mediante las técnicas de agrupamiento, con el fin de identificar posibles regímenes de funcionamiento en el espacio de datos elegido. Cada uno de los grupos de datos detectados tiene, generalmente, un comportamiento similar y se podrá obtener en un paso posterior un modelo común a todos ellos, lineal o no. Cada uno de esos regímenes de funcionamiento podría, por ejemplo, ser definido por un modelo lineal. El conjunto de esos submodelos lineales aproximará el modelo no lineal de la forma prevista en el capítulo 3. Sin embargo, este método tiene sus limitaciones (ver D.3).

La primera limitación puede darse por la disposición de los datos con los que se va a trabajar, ya que si el ensayo realizado sobre el sistema no está perfectamente diseñado (no cubre todo el espacio a analizar), la distribución resultante suele ser continua y esto se adecua más a la regresión que al agrupamiento.

Otro problema surge con la decisión del espacio de datos sobre el que realizar el agrupamiento, ya que los datos tienen significado en un espacio que incluye todas las entradas y salidas al sistema. La partición de los datos en, por ejemplo, pares entrada-salida y la posterior proyección de las clases puede llevar a solapamientos y dificultad en la interpretación. De todas formas esto es más una dificultad inherente del proceso de identificación que un escollo insalvable.

Un último problema es la elección del algoritmo de agrupamiento. Las diferencias entre la mayoría de algoritmos de agrupamiento estudiados estriban en el tipo de norma elegida y la forma de la clase (ver tabla 6.1), con lo que el mismo grupo de datos puede dar lugar diferentes agrupaciones si se elige uno u otro método. Ninguno de ellos es, en principio, mejor o peor que los otros, sino que cada uno será apropiado en ciertas circunstancias. La información que a priori se puede tener sobre la forma de las clases puede ser fundamental, ya que se han visto métodos de agrupamiento que buscan elipsoides de tamaños similares, elipsoides de tamaños variables, prototipos lineales, etc. El método más utilizado es el FCM ((Davé and Krishnapuram, 1997), (Hwang and Woo, 1995)), aunque también hay referencias a FMLE (Genther and Glesner, 1997), FCRM (Kim *et al.*, 1997), GK y GG ((Klawonn and Kruse, 1995b), (Klawonn and Keller, 1997), (Klawonn and Kruse, 1997), (Ronen *et al.*, 1998)). Algunos autores desarrollan su propio algoritmo de agrupamiento para la identificación de modelos borrosos en su aplicación concreta, como el agrupamiento hiperelipsoidal (Nakamori and Ryo-

ke, 1994), o modifican algoritmos ya existentes como las variaciones de GK y GG en (Klawonn and Kruse, 1995*b*), (Klawonn and Keller, 1997) y (Klawonn and Kruse, 1997). Concretamente, en el caso de modelado de sistemas no lineales para su posterior aplicación al control del sistema, los métodos de agrupamiento más apropiados si se quieren aplicar técnicas de control lineales a cada grupo de datos, serán aquellos que busquen consecuentes lineales (clases lineales o hiperplanos) como los vistos en el punto C.4. En la tabla 6.1 se hace una primera selección de algoritmos enfocada a los problemas que trata de resolver esta Tesis, pero el capítulo 7 tratará este tema con mayor profundidad desde puntos de vista cuantitativos y cualitativos.

6.7.4. Selección del número de clases

Excepto un número reducido de los algoritmos de agrupamiento (chain-map, max-min y agrupación por eliminación, ver anexo C), el resto necesita saber, con mayor o menor certeza, el número de clases con los que se va a encontrar para comenzar a iterar a la busca de clases.

Una solución obvia al problema del número de clases inicial es realizar un estudio previo de los datos con algún algoritmo capaz de detectarlos como chain-map, max-min o agrupación por eliminación. El valor obtenido en esta etapa de inicialización se introducirá en el algoritmo que se desee emplear. Existen otras soluciones más avanzadas a este problema, que pasan por una iteración de los pasos comentados en los apartados 6.7.2 y 6.7.3 hasta que se llegue a un valor apropiado junto a alguno de los siguientes métodos:

- Comenzar con un número de clases suficientemente elevado y realizar uniones entre clases compatibles con el *algoritmo de unión de clases compatibles* (CCM) (Krishnapuram and Freg, 1992) es una posibilidad a tener en cuenta (ver C.8.1).
- Realizar la agrupación de los datos para diferentes valores de c , y usar posteriormente *medidas de validez* para determinar la bondad de las particiones obtenidas (ver C.8.2).
- Emplear *algoritmos de agrupamiento parcialmente supervisados* si se dispone de datos de alguna clase en concreto (ver C.8.3).

6.7.5. Determinación de un modelo borroso inicial

De algún modo, y basándose en la agrupación de los datos realizada en los pasos anteriores para el espacio de las variables elegidas, se han de derivar reglas para alguno de los tipos de modelo borroso vistos en 5.1.

La clasificación de los datos realizada podría permitir, por ejemplo, determinar un modelo lineal para cada uno de las clases detectados, con lo que se tendría una buena aproximación del sistema no lineal complejo en estudio. Esta determinación de modelos borrosos basados en particiones ha sido muy empleada últimamente (Babuska, 1996). También existe la posibilidad de, por ejemplo, hacer un modelo basado en redes neuronales para cada clase (Ronen *et al.*, 1998) siendo la salida total del sistema una ponderación del resultado obtenido para cada red.

En cuanto a los modelos borrosos de reglas basados en esas particiones, en principio, cualquier tipo de modelo borroso visto en la sección 5.1 (Takagi-Sugeno, lingüístico, relacional) es válido. Se pueden encontrar referencias de todos ellos, aunque es más utilizado el modelo lingüístico ((Sugeno and Yasukawa, 1993), (Hong and Lee, 1996), (Genther and Glesner, 1997), (Klawonn and Keller, 1997), (Klawonn and Kruse, 1997), (Hwang and Woee, 1995)) que el de Takagi-Sugeno ((Nakamori and Ryoike, 1994), (Babuska, 1996), (Kim *et al.*, 1997)).

Dado que la opción que se persigue en el presente trabajo es la de obtener modelos como los vistos en el capítulo 3, se revisará la obtención de modelos borrosos de Takagi-Sugeno cuyos consecuentes sean modelos locales afines. La forma de obtener los lingüísticos es similar y se puede revisar en D.4.2.

En este caso se busca un modelo basado en reglas del tipo (5.2), en el que se va a suponer que el consecuente de la regla es una función lineal. Las reglas que se proponen son:

$$R_i: \text{Si } x \text{ es } A_i \text{ entonces } y_i = a_i^T x + b_i, i = 1, 2, \dots, c;$$

donde a_i es un vector de parámetros y b_i un escalar. A este modelo se le llama *modelo afín de Takagi-Sugeno*, como ya se vio en 5.1. Por tanto, llegados a este punto, será necesario estimar funciones de pertenencia para las variables A_i del antecedente y los parámetros a_i y b_i del consecuente para cada regla de las c que se determinen con los métodos descritos en 6.7.2.

Funciones de pertenencia del antecedente

Desde los resultados del proceso de agrupamiento de los datos, se tienen dos opciones para obtener funciones de pertenencia del antecedente.

Generación de funciones de pertenencia del antecedente por proyección.

En este método se proyectan los conjuntos borrosos definidos punto a punto en la matriz de partición sobre las variables del antecedente, como en (Sugeno and Yasukawa, 1993), (Genther and Glesner, 1997), (Klawonn and Keller, 1997), (Klawonn and Kruse, 1997), (Delgado *et al.*, 1997) y (Hwang and Woee, 1995). Si se proyecta sobre las variables del conjunto de datos inicial, nos encontramos ante una proyección ortogonal, como se puede observar en la figura 6.4.

Con una proyección de este tipo se pueden obtener errores en el sentido de pérdida de información (zona sombreada en la figura 6.4), si los ejes de la clase no son paralelos (o casi paralelos) a los ejes sobre los que se proyecta. Aunque hay mejoras a la proyección (ver D.4.1), se elija el método de generación de funciones de pertenencia que se elija, después del agrupamiento del conjunto de datos Z , para cada dato $z_k^T = [z_{1k}, \dots, z_{nk}, y_k]$ los grados de pertenencia son dados por la matriz de partición borrosa $U = [\mu_{ik}]$. Para z_k estará dado por la k -ésima columna de la matriz de partición $\mu_k^T = [\mu_{1k}, \dots, \mu_{ck}]$. Proyectando la fila i (grado de pertenencia a una clase i de cada dato j) sobre la variable x_j , se obtiene una proyección como la de la figura 6.5.a. Evidentemente, esta no es la forma adecuada de una función de pertenencia. Por supuesto, se puede aproximar por el tipo de función convexa que se desee (como en 6.5.b), aunque lo más habitual en la

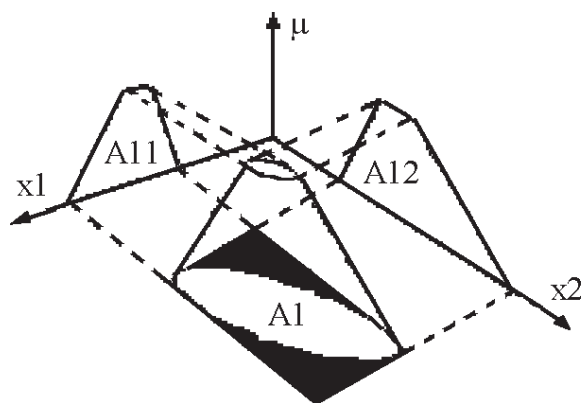


Figura 6.4: Proyección ortogonal de una clase del espacio de entrada tomando como ejes las variables de entrada.

bibliografía es utilizar funciones de pertenencia triangulares o, a lo sumo, trapezoidales, como se muestra en la figura 6.5.c. Evidentemente las proyecciones (o sus ajustes) no cumplen con la restricción de partición borrosa (suma de pertenencia 1) y este extremo también ha de ser tenido en cuenta.

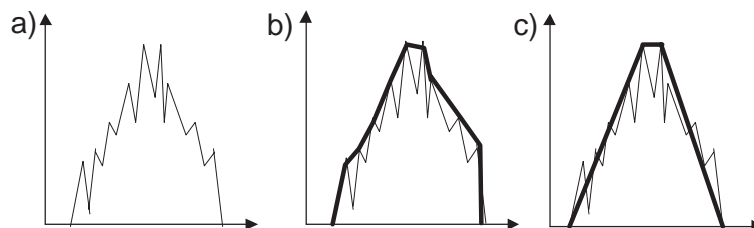


Figura 6.5: Función de pertenencia proyectada y aproximaciones.

Generación de funciones de pertenencia del antecedente de forma directa en el espacio de entradas. En este caso se calculan los grados de pertenencia directamente en el espacio de las variables de entrada (Delgado *et al.*, 1997), usando la norma empleada en el algoritmo de agrupamiento.

La idea es medir la distancia de cada punto al prototipo de cada clase y transformarla en un grado de pertenencia mediante la aplicación de alguna media ponderada de la inversa de la distancia (con expresiones similares a las que aparecen en los algoritmos de agrupamiento), con lo que se da un valor de pertenencia de cada punto a cada clase. Esto conlleva la desventaja de su alta carga computacional pues, para evaluar la pertenencia de nuevos datos, cada vez hay que buscar entre los valores de las funciones de pertenencia identificadas el punto más cercano al cual se pretende evaluar la pertenencia, asignando esa pertenencia o bien interpolando entre varias pertenencias si hay más de un punto a la misma distancia.

Una segunda opción consiste en intentar ajustar los parámetros de alguna superficie con forma similar a las funciones de pertenencia identificadas. De este modo, si la

superficie a la que se decide ajustar no es la correcta se puede perder información y unos buenos modelos lineales pueden llegar a componer un mal modelo no lineal por su incorrecta interpolación.

En ambos casos las reglas serán proposiciones de conjuntos borrosos multidimensionales.

Parámetros del consecuente

Para la estimación de los parámetros del consecuente del modelo borroso de Takagi-Sugeno, la solución habitual (Babuska, 1996) es aplicar a los resultados del agrupamiento técnicas de mínimos cuadrados que minimicen el error de predicción del modelo global o de cada uno de los subespacios aproximados por cada clase. Otra opción bastante común es plantear problemas de optimización que sean resueltos por algoritmos genéticos, como en (Delgado *et al.*, 1997) y (Hwang and Woee, 1995).

Cuando se usa para la agregación de los resultados de los consecuentes de las reglas borrosas la fórmula de la media borrosa (5.4), la estimación de los parámetros del consecuente se puede obtener resolviendo un problema lineal de mínimos cuadrados (ver D.4.1).

Un opción muy interesante es que sea el propio algoritmo de agrupamiento el que genere los antecedentes y consecuentes de la reglas (Díez *et al.*, 2002a). Esta idea será implementada por el algoritmo que se propone en el capítulo 8 de esta Tesis.

6.7.6. Simplificación del modelo inicial

Llegados a este punto, se ha construido un modelo de Takagi-Sugeno cuya inferencia se reduce a evaluar (5.4) o alguna expresión alternativa de características similares siguiendo las recomendaciones del apartado 5.2. Los modelos borrosos determinados desde los datos experimentales suelen ser redundantes en el sentido de tener más funciones de pertenencia y/o reglas de las necesarias, con lo que alguna medida de similaridad deberá ser empleada para simplificar el modelo inicial.

El proceso de simplificación puede ser cedido a un experto, o bien puede ser automatizado. El empleo de CCM (Compatible Cluster Merging, punto C.8.1) puede simplificar el modelo. Pero si lo que se pretende es identificar conjuntos borrosos similares (que se solapan y describen prácticamente la misma región) en el modelo borroso inicial y reemplazarlos por un conjunto borroso común que represente a los originales, se pueden usar medidas de similaridad (similaridad, matemáticamente hablando, se refiere a algún tipo de norma). De este modo se reduce el número de conjuntos borrosos en el modelo e incluso, si la redundancia es alta, puede dar lugar a eliminar reglas similares. Un ejemplo de algoritmo de simplificación de una base de reglas se muestra en D.5.

6.7.7. Validación del modelo

Diferentes métodos de validación pueden ser usados en esta etapa final, con la idea de aceptar el modelo de forma definitiva o comenzar de nuevo el proceso de modelado desde el principio.

La validación final del modelo borroso obtenido puede ser realizada por simulación y/o revisión de las reglas por parte del experto. Lo más habitual es una validación mediante simulaciones numéricas y comparación con datos reales del proceso, como se realiza en la mayoría de la bibliografía (Sugeno and Yasukawa, 1993; Emami *et al.*, 1998; Babuska, 1996).

Un tema de especial importancia es la interpretabilidad del modelo borroso obtenido pues, como se ha comentado, esta es la característica más reseñable de los sistemas inteligentes y no su capacidad de aproximadores universales.

El análisis del espacio de entrada que cubre la base de reglas no debe ser olvidado. Si una base de reglas es incompleta, se pueden añadir reglas por conocimiento del sistema o mediante un modelo lineal del sistema en el entorno que no había sido cubierto por las reglas iniciales.

6.8. Conclusiones

En este capítulo se han visto los fundamentos de los denominados algoritmos de agrupamiento borroso, ya que son los más apropiados para la aproximación de sistemas no lineales complejos desde datos numéricos con modelos borrosos.

Dado que lo que se pretende en esta Tesis es emplear los métodos de agrupamiento para identificar modelos borrosos con consecuentes afines tipo TS (capítulo 5) que integren modelos LPV consistentes en familias de modelos locales (capítulo 3), se ha centrado la revisión en aquellos algoritmos con capacidades de detección de subespacios lineales.

La integración de todas las técnicas se ha hecho mediante una revisión de los pasos necesarios para la construcción de un modelo borroso TS de un sistema no lineal mediante técnicas de agrupamiento. En cada uno de esos pasos se han comentado los métodos existentes en la bibliografía que nos permiten llevarlos a cabo.

Como se ha visto, la aplicación de las técnicas de agrupamiento en la determinación del modelo borroso es muy apropiada, por la relación directa entre las clases formadas en el proceso de agrupamiento, y más concretamente en el de tipo borroso, y las funciones de pertenencia necesarias para la definición de los elementos de antecedentes y consecuentes de las reglas.

A modo de resumen del capítulo, se plantea la tabla 6.1. En ella se enumeran todas las técnicas de agrupamiento borroso revisadas, resaltando sus características más relevantes y eligiendo una técnica de cada tipo para ser validada posteriormente (capítulo 7) en la identificación de modelos borrosos TS.

En el capítulo 7 también se propondrá una técnica concreta para cada uno de los pasos comentados como necesarios en la identificación, llegando a un proceso sistematizado y automatizado para la identificación de modelos borrosos que será evaluado bajo la perspectiva de la posterior aplicación a control.

Técnica	Tipo	Ventajas	Problemas
FCM	Básico	Sencillez	Orientación y forma de las clases
GK	Adaptativo	Poco sensible a inicialización	Clases hiperelipsoidales
FMLE	Adaptativo	Detecta clases diversos	Sensible a inicialización, clases hiperelipsoidales
GG	Adaptativo	Ventajas GK y FMLE	Alta carga computacional, clases hiperelipsoidales
FCV	Lineal	Detecta hiperplanos	Conecta clases lejanas
FCRM	Lineal	Detecta hiperplanos, ajusta a funciones lineales	Conecta clases lejanas
FCE	Mixto	Ventajas FCM y FCV	Elección parámetro alfa
AFCR	Mixto	Ventajas FRCM, FCM y GK	-

Tabla 6.1: Adecuación de las técnicas de agrupamiento borroso revisadas a la identificación de modelos borrosos TS.

Parte III

Propuesta de técnicas para identificación y control

Capítulo 7

Evaluación de las técnicas de agrupamiento en identificación

*Si buscas resultados distintos,
no hagas siempre lo mismo.*
Albert Einstein.

Una vez descritos en el capítulo anterior todos los pasos a seguir en la identificación borrosa, se propone un proceso sistematizado y automatizado (basado en (Sugeno and Yasukawa, 1993), (Emami *et al.*, 1998) y (Babuska, 1996)) aplicable siempre que los estados del sistema en estudio sean medibles.

En la aplicación de la técnica propuesta se pondrán de manifiesto algunas deficiencias que, en la posible aplicación a control de los modelos identificados de forma sistematizada, muestran las técnicas de agrupamiento. Posteriormente se analizarán, los algoritmos de agrupamiento que podrían ser interesantes para la identificación de sistemas.

Para ello se comenzará con un análisis de la idoneidad de las funciones objetivo de los algoritmos de agrupamiento como criterios de modelado en la identificación para control, pasando posteriormente (tras definir unas funciones de validez) a realizar un análisis cuantitativo y cualitativo de los algoritmos de agrupamiento.

7.1. Método sistematizado de identificación

La propuesta metodológica para la identificación de modelos borrosos de sistemas no-lineales mediante técnicas de agrupamiento consistirá en la repetición de los pasos que se vieron en la figura 6.3, particularizados con las técnicas propuestas para cada uno de ellos.

Selección de entradas. Para la selección de entradas entre un conjunto de variables candidatas se propone emplear, por su precisión, el método heurístico propuesto por Sugeno (Sugeno and Yasukawa, 1993) que se basa (tras dividir el conjunto de datos en dos grupos similares, A y B) en ir incrementado el número de entradas una a una y

evaluar el criterio de regularidad (7.1), hasta alcanzar un mínimo local. En el apéndice D.2.1 se puede observar un ejemplo de aplicación de este criterio de regularidad:

$$RC = \frac{\left[\sum_{i=1}^{k_A} (y_i^A - y_i^{AB})^2 / k_A + \sum_{i=1}^{k_B} (y_i^B - y_i^{BA})^2 / k_B \right]}{2} \quad (7.1)$$

donde:

- k_A y k_B son el número de datos de los grupos A y B
- y_i^A e y_i^B son las salidas correspondientes a los datos de A y de B
- y^{AB} es la salida estimada para la salida de los datos de B por el modelo identificado con los datos de A
- y^{BA} es la salida estimada para la salida de los datos de A por el modelo identificado con los datos de B

Determinadas las variables de interés, en cuanto al resto de elementos de la estructura, la representación dinámica será la propuesta en la sección 6.7.2 para sistemas discretos (en estos sistemas se centrará la Tesis) y la granularidad del modelos borrosos vendrá dada por el número de clases.

Número de clases. Para determinar el número óptimo de clases se aplica de forma iterativa el índice de varianza de Sugeno ((7.2), (Sugeno and Yasukawa, 1993)) y el de dispersión de Emami ((7.3), (Emami *et al.*, 1998)) para cada número de clases a estudiar:

$$S_S(C) = \sum_{i=1}^c \sum_{k=1}^N (\mu_{ik})^m (\|z_k - c_i\|^2 - \|c_i - \bar{z}\|^2) \quad (7.2)$$

$$S_E(C) = \sum_{i=1}^c \sum_{k=1}^N (\mu_{ik})^m (\|z_k - c_i\|^2 - \|c_i - \bar{c}\|^2)$$

$$\bar{c} = \frac{1}{\sum_{i=1}^c \sum_{k=1}^N (\mu_{ik})^m} \sum_{i=1}^c \sum_{k=1}^N (\mu_{ik})^m z_k \quad (7.3)$$

$$c_i = \frac{1}{\sum_{k=1}^N (\mu_{ik})^m} \sum_{k=1}^N (\mu_{ik})^m z_k$$

donde: N es el número de datos, c es el número de clases, z_k es el k-ésimo vector de datos, \bar{z} es la media de los datos y c_i el centro del i-ésima clase. En ambos casos, el número de clases se determina incrementando c hasta que S(c) alcanza un mínimo local. A pesar de la carga computacional que conlleva esta evaluación, es interesante realizarla por ser la determinación del número de clases una parte muy importante previa al agrupamiento (ver anexo C.8).

Agrupación de datos. Cuando sea necesario aplicar el algoritmo de agrupamiento, se empleará (en un principio) el algoritmo GK (Gustafson and Kessel, 1979), dadas sus adecuadas propiedades de detección de clases lineales (Babuska, 1996) y que las ventajas parecen mayores que las desventajas (ver tabla 6.1 de la página 65). De todas formas, posteriormente se compararán todos los algoritmos adecuados en identificación según la tabla 6.1. Conviene resaltar además que, al igual que todo algoritmo de agrupamiento, GK necesita que le sean suministrados para su inicialización una serie de parámetros, como son:

- El espacio de datos al que aplicar el agrupamiento. Los mejores resultados en la bibliografía (Delgado *et al.*, 1997) se obtienen (para precisión e interpretabilidad de modelos) con el espacio completo de entradas-salidas, por lo que éste será el elegido.
- Un valor para el parámetro de borrosidad m . Aunque se han propuesto métodos (Emami *et al.*, 1998) para la estimación de éste, no hay ninguna base teórica al respecto. El valor de $m = 2$ es el más comúnmente utilizado y se tomará así en el presente trabajo.

Determinación de un modelo borroso. Para la determinación de un modelo borroso inicial de Takagi-Sugeno con las clases determinadas en el proceso de agrupación de los datos, se definirán funciones de pertenencia borrosas en el espacio n -dimensional de las entradas para los antecedentes de las reglas, con lo que al no proyectar no es tan necesario el preprocesado y se obviará. Los parámetros de las funciones lineales de los consecuentes serán ajustados por mínimos cuadrados.

Simplificación y Validación. Estos pasos, realizados habitualmente por un experto, se harán siguiendo las indicaciones dadas en la sección 6.7.7.

Con el algoritmo propuesto se pueden determinar de forma automática modelos de caja negra como el de la figura 5.2 con estructuras de modelo afín de TS (5.3). Todas las funciones necesarias han sido programadas en Matlab y se comentan en D.7. A continuación, con el fin de poner de manifiesto el comportamiento de la propuesta, se muestra un ejemplo sencillo de aplicación.

Ejemplo 7.1. Caso 1: función bidimensional estática no lineal. Este ejemplo es muy adecuado para comprobar la efectividad del método propuesto en la identificación de sistemas no lineales, ya que es profusamente utilizado en la literatura (benchmark que se usa, por ejemplo, en (Sugeno and Yasukawa, 1993) y (Emami *et al.*, 1998)) y será sencilla la comparación de resultados. Aunque la función identificada es estática, dado que los sistemas dinámicos son tratados para la identificación borrosa como funciones multidimensionales estáticas (Sugeno and Yasukawa, 1993), los resultados mostrarán igualmente la adecuación de la metodología en la identificación de sistemas dinámicos.

Se pretende identificar la función bidimensional (figura 7.1)

$$y = (1 + x_1^{-2} + x_2^{-1.5})^2 \quad \text{con} \quad 1 \leq x_1, x_2 \leq 5 \quad (7.4)$$

de la que existe una lista de 50 pares de datos para x_1 y x_2 en (Sugeno and Yasukawa, 1993) para identificar el sistema. Adicionalmente se proporcionan datos de dos variables x_3 y x_4 "falsas" para comprobar que se eligen correctamente las variables que influyen en el sistema a modelar.

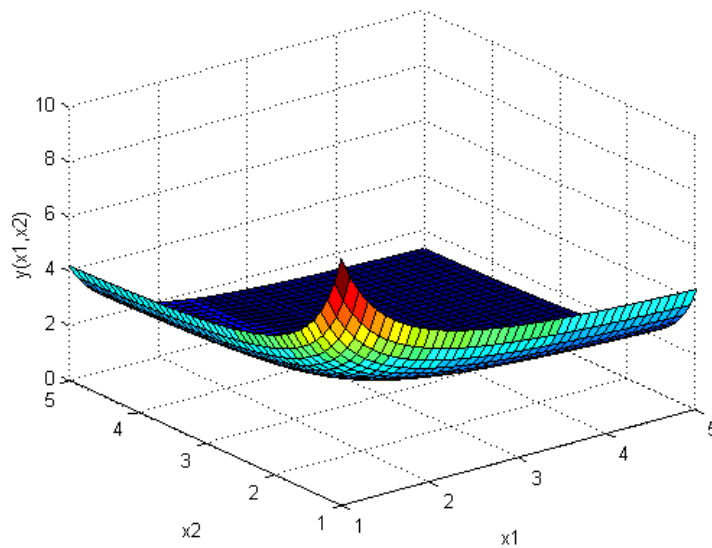


Figura 7.1: Relación entradas-salida del sistema a modelar.

Paso 1. Selección entradas. Para ello (empleando las funciones *bestdata* en D.7) se toma la entrada que muestre un menor RC (7.1), hasta que éste aumente.

- 1 entrada: $RC(x_1)=0.4327$, $RC(x_2)=0.3791$, $RC(x_3)=0.8278$, $RC(x_4)=0.9758 \Rightarrow$ Elijo entrada x_2
- 2 entrada: $RC(x_2, x_1)=0.3491$, $RC(x_2, x_3)=3.1228$, $RC(x_2, x_4)=3.7115 \Rightarrow$ Elijo entrada x_1
- $RC(x_2, x_1, x_3)=11.0159$, $RC(x_2, x_1, x_4)=0.9970$ \Rightarrow RC aumenta

Ya se tienen las entradas "buenas", y estas son x_1 y x_2 , como era de esperar y como se obtiene en (Sugeno and Yasukawa, 1993) y (Emami *et al.*, 1998).

Paso 2. Determinación del número de clases. Dado el espacio entradas-salida (x_1, x_2, y) , se toman los métodos dados en (7.2) y (7.3) (funciones *nclust_em* y *nclust_sy* del anexo D.7), obteniendo el mismo resultado: 4 clases. Este valor es el determinado por (Sugeno and Yasukawa, 1993); en cambio (Emami *et al.*, 1998) estima necesarios 8 clases.

Paso 3 y 4. Obtención y Simulación de un modelo. Una vez determinado el modelo (función f_{mid} del anexo D.7) se debe simular (función f_{msim} en el anexo D.7) para su validación.

Una forma habitual para evaluar la bondad del modelo obtenido de forma objetiva, es emplear un índice de funcionamiento (IF). Para poder comparar los resultados con los de la bibliografía se ha optado por el error cuadrático medio entre la salida real y la del modelo (7.19).

El resultado de la evaluación del IF para el modelo obtenido, así como la comparación con los procesos de identificación de (Sugeno and Yasukawa, 1993) y (Emami *et al.*, 1998) para el mismo conjunto de datos, se muestra en la tabla 7.1.

	Sugeno	Emami	I-FMID
IF (RMS)	0.318	0.171	0.1106

Tabla 7.1: IF del modelo obtenido y de los modelos de la literatura.

Los resultados son prometedores para la estructura de identificación propuesta en cuanto al IF. Sin embargo, conviene estudiar que ocurre en lo que respecta a la interpretabilidad de las clases identificadas.

La figura 7.2 muestra como los pares de datos de entrada son asignados a las clases (se ha considerado la mayor pertenencia de un punto a una clase para realizar la figura) por la aplicación de GK. Para favorecer la interpretación de la figura 7.2, se ha dibujado el prototipo de la clase (+) y los límites de las tres clases convexas en la parte izquierda, mientras que la clase no interpretable se muestra con un rayado en la parte derecha.

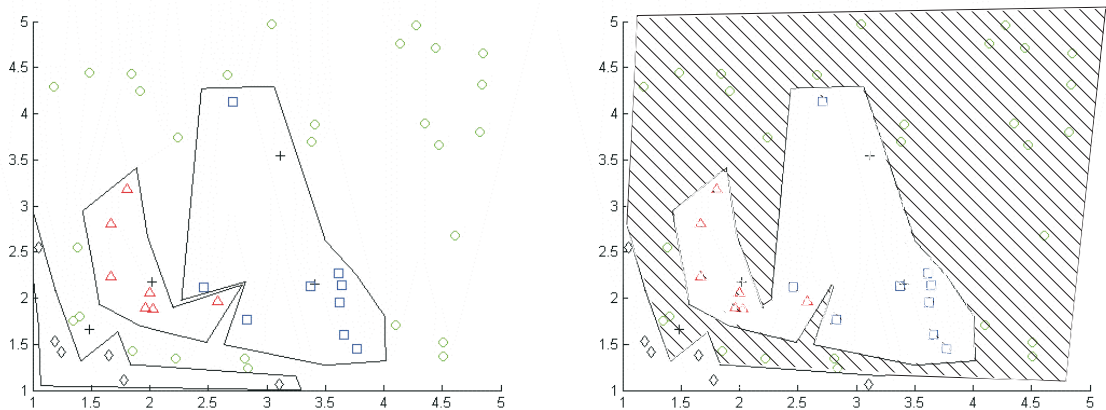


Figura 7.2: Clases identificadas.

Las funciones de pertenencia identificadas no son convexas y no corresponden a los modelos locales que daría un experto al inspeccionar la figura 7.1.



En definitiva, se puede concluir que los valores numéricos de los IF de los modelos obtenidos son comparables a los existentes en la literatura, siendo los resultados satisfactorios en estos modelos para predicción. La estructura algorítmica que se ha planteado demuestra así su validez para el modelado e identificación de sistemas por medio de modelos borrosos.

Sin embargo, la interpretabilidad de las funciones de pertenencia identificadas y la correspondencia de los modelos locales identificados con linealizaciones en el entorno de los prototipos no se puede asegurar y, de hecho, se ha mostrado en el ejemplo que no se corresponde con lo deseable, habiéndose comprobado en experimentos adicionales como el ruido empeora más este proceso de identificación. Estos problemas son muy comunes en la aplicación de los algoritmos de agrupamiento en identificación y no solo se dan con el algoritmo GK, tal y como se comenta en el apartado 7.2 y se muestra en el análisis del apartado 7.5.

7.2. Criterios de modelado

La discusión que se presenta en este apartado, aunque enfocada a los modelos que se ha decidido emplear en esta Tesis (modelos borrosos TS de sistemas discretos), es aplicable de forma general al modelado de no-linealidades mediante aproximadores universales.

En un sistema genérico dinámico no-lineal discreto, en el que se asume que las no-linealidades del sistema y la dinámica están separadas, la estructura del sistema será de la forma:

$$x_{k+1} = F(x_k, u_k) + e(k) \quad (7.5)$$

donde las variables de las que depende F son medibles y $e(k)$ representa el ruido y la dinámica no modelada, ecuación que se corresponde con (3.4) a la que se ha añadido el término $e(k)$ de error.

El objetivo de este apartado será discutir sobre la idoneidad, en lo que respecta a la aplicación al diseño de controladores e interpretabilidad para el usuario, de los diferentes criterios de modelado que se emplean para el modelado de la no-linealidad F mediante técnicas borrosas (incluyendo las de agrupamiento). Las ideas de modelado para su posterior uso en control han aparecido recientemente en el campo de los sistemas lineales (Albertos and Sala, 2002) y deben ser extendidas a las técnicas de identificación de modelos borrosos que se empleen para control (Díez *et al.*, 2001), pues el criterio de modelado puede ser distinto al de los modelos borrosos empleados para predicción.

Considérese la función continua de modelado $Y = F(z) + e(t)$, donde z agrega las variables de estado y de entrada, $e(t)$ es el término de ruido e Y es el término estimado (normalmente x_{k+1}).

Definición de Aproximador Universal. *Si para toda función $F(z)$ continua en z y para cualquier $\epsilon > 0$ la función $f()$ cumple siempre que*

$$\sup_z |f(z) - F(z)| < \epsilon \quad (7.6)$$

entonces $f()$ es un aproximador universal.

Teóricamente se ha demostrado que, bajo ciertas condiciones, un sistema borroso se comporta como una aproximador universal (en (Wang, 1997) se revisa este aspecto). Las técnicas de identificación tratan de encontrar una aproximación (borrosa) $f()$ al sistema real $F()$ desde un conjunto n -dimensional de datos entrada-salida con N elementos:

$$Z = \{(z_k, y_k) \quad k = 1, \dots, N\} \quad (7.7)$$

Se pueden considerar dos enfoques fundamentales en lo que a la identificación borrosa se refiere: la minimización de un error "global" de predicción (diferencia entre $y(t)$ y $f(t)$) (Ljung, 1999) y la minimización de un error "local" de un determinado número de modelos locales (definidos por el usuario o generados en el proceso de identificación) que representen el sistema en una región de su espacio de operación (Babuska, 1996). El enfoque *global* puede llegar a alcanzar una mayor precisión entre la salida real y estimada del proceso, mientras que los *modelos locales* mejoran la interpretabilidad de los modelos identificados (Lee and Yen, 2002; Johansen *et al.*, 2000). En la identificación con técnicas de inteligencia artificial, este compromiso entre interpretabilidad y precisión se debe tener en cuenta y el uso de los conceptos (conjuntos borrosos) y modelos originados dependerá del enfoque inicial por el que se haya optado para la identificación, como ya reflejan algunos autores en trabajos recientes (Abonyi *et al.*, 2000; Dzielinski and Graniszewski, 2002).

7.2.1. Modelos globales de sistemas no lineales

El enfoque global del modelado consiste en la minimización de una función objetivo como la definida por:

$$J_g = \sum_{k=1}^N (y_k - f(z_k))^2 \quad (7.8)$$

donde $f(\cdot)$ es cualquier aproximador universal de funciones, como los modelos borrosos TS, las redes de funciones de base radial, etc. Muchos de ellos pueden ser expresados como

$$f(z_k) = \sum_{i=1}^c \mu_i(z_k) f_i(z_k, \beta_i) \quad (7.9)$$

donde c ($2 \leq c \leq N$) representa el número de regiones en las que debe ser dividido el espacio de operación, β_i son los vectores de parámetros de los modelos "locales" f_i , y μ_i es una función de validez (pertenencia). Una opción muy común es emplear modelos afines de TS (5.3). Las entradas a las funciones de validez pueden incluir otras *variables de planificación* diferentes de las entradas-salidas que deben ser modeladas. Los datos $\mu_i(z_k)$ se pueden reorganizar en forma de matriz μ_{ik} de dimensiones $c \times N$ que cumple las condiciones de un espacio de partición borrosa (6.15).

Dar el nombre de "modelo local" a f_i en (7.9), algo muy habitual en la literatura (Hunt and Johansen, 1997), puede ser confuso ya que los f_i identificados pueden no corresponder a ningún comportamiento local y el modelo identificado explica los datos reales por la agregación de todo el conjunto de las f_i . De este modo es imposible comprender fácilmente el sistema complejo, a no ser que el solapamiento de los conjuntos borrosos sea muy pequeño.

Otra cuestión relacionada con modelos como (7.8) que puede empeorar aún más su ya baja interpretabilidad aparece al analizar su condicionamiento: si las funciones de pertenencia μ_{ik} se aproximan mediante una estructura suficientemente flexible podrían llegar a emular un aproximador universal de funciones para cualquier f_i fijado arbitrariamente.

En resumen, el empleo de una función objetivo global en la identificación de sistemas tiene la ventaja del fácil entrenamiento del modelo (si f_i son funciones de parámetros lineales, se pueden usar mínimos cuadrados) y su elevada precisión. En cuanto al diseño de controladores, pueden ser empleadas diferentes técnicas como LMI (Tanaka and Sugeno, 1992) donde se pueden calcular funciones de Lyapunov para asegurar la estabilidad (Carbonell *et al.*, 2000), control por planificación de ganancia (Hunt and Johansen, 1997), redes de controladores de mínima varianza generalizados (Díez and Previdi, 2001), etc. Sin embargo, emplear conceptos como “controladores locales”, “punto de operación” o “linealización” cuando se hable de las f_i puede no tener sentido. Adicionalmente, pueden aparecer problemas de mal condicionamiento en la función objetivo a optimizar.

7.2.2. Modelos locales de sistemas no lineales

La opción alternativa para el modelado de sistemas no lineales es dividir el espacio de operación en un determinado número de conjuntos, estimando modelos locales que aproximen realmente el comportamiento de la planta en un conjunto determinado (capítulo 3). La función objetivo a minimizar para este enfoque es:

$$J_l = \sum_{i=1}^c \sum_{k=1}^N \mu_{ik} (y_k - f_i(z_k))^2 \quad (7.10)$$

donde μ_{ik} verificará, de nuevo, las condiciones (6.15).

En este caso $f_i(z_k)$ debe ser ajustado a los datos ponderando el ajuste proporcionalmente a los valores de pertenencia. En el caso límite de funciones de pertenencia cercanas a los singletons, las f_i lineales que minimizan el índice (7.10) serán los modelos linealizados en el entorno de los prototipos (z_i^*, y_i^*) en los que $\mu_{ik}^* = 1$ (como los descritos en el capítulo 3).

La interpretabilidad y comprensión del usuario de este tipo de modelos es mayor que la de aquellos que optimizan (7.8). Esto lleva a una buena interacción entre el modelo identificado y el usuario (el ingeniero de control) dado que la intuición puede ser usada para los trabajos de diseño local y validación. Si los modelos locales elegidos son afines, las técnicas de diseño de controladores clásicas son aplicables al sistema dinámico no lineal en estudio.

La principal desventaja de este tipo de sistemas es que los diseños locales no pueden asegurar la estabilidad en bucle cerrado empleando funciones de interpolación como (5.4). Como se vió en el capítulo 5, esta combinación de los modelos obtenidos con (7.10) hace que su interpolación (posiblemente no convexa) sea errónea.

Para integrar correctamente estos modelos locales en un marco global se pueden emplear dos alternativas:

- determinar ciertas regiones de incertidumbre (y algoritmos para obtenerlas desde los datos) que se expandan según los modelos van perdiendo validez y diseñar posteriormente controladores robustos, o
- reemplazar interpolación original de los modelos TS (conservando el resto de la estructura) por otro tipo de interpolación que produzca funciones “convexas”, como la que se define a continuación, y emplear técnicas de planificación de ganancia (Rugh and Shamma, 2000) para el diseño de controladores.

Interpolación convexa en modelos TS. Para una aplicación al modelado, identificación y control basados en modelos borrosos TS como integración de los modelos locales lineales comentados en el capítulo 3, un mecanismo de inferencia conveniente sería el que resultase en una superficie de interpolación cóncava o convexa tal que:

- los gradientes de la función estuviesen limitados por los gradientes de los consecuentes de las reglas,
- la superficie interpolada fuese suave y con derivada continua, y
- el modelo se aproximase por una función más precisa que la generada por la interpolación TS original (5.4).

De este modo se evitarían los problemas que se ponen de manifiesto en el Ejemplo 5.1 del capítulo 5.

En el caso de dos reglas, una opción sencilla es optar por una transición brusca mediante la función máximo haciendo $\hat{y} = \max(y_1, y_2)$ para funciones convexas o $\hat{y} = -\max(-y_1, -y_2) = \min(y_1, y_2)$ para funciones cóncavas. Los problemas que se generan en el punto de intersección de los modelos (gradiente no definido) se pueden resolver reemplazando la función máximo con el *máximo suave* (Seber and Wild, 1989) dada por:

$$\max_{\gamma}(x) = \begin{cases} 0 & x \leq -\gamma \\ \frac{(x+\gamma)^2}{4\gamma} & x \in [-\gamma, \gamma] \\ x & x \geq \gamma \end{cases} \quad (7.11)$$

con lo que la nueva inferencia para modelos TS de dos reglas que sustituiría a $\hat{y}(x) = \mu_1(x)y_1(x) + \mu_2(x)y_2(x)$ (que es la que daría (5.4)) queda:

$$\hat{y}(x) = y_1(x) + \max_{\gamma}(y_2(x) - y_1(x)) \quad (7.12)$$

En el caso general de solapamiento de R reglas con antecedentes del tipo conjuntivo (unidos por “y”, como en (5.1)), la aplicación recursiva de (7.13) lleva a la alternativa a (5.4):

$$\hat{y}(x) = y_1(x) + \max_{\gamma_1}(y_2(x) + \max_{\gamma_2}(\dots \max_{\gamma_{R-2}}(y_{R-1} + \max_{\gamma_{R-1}}(y_R - y_{R-1}))\dots)) \quad (7.13)$$

La determinación de γ se puede realizar mediante alguna técnica de optimización o, si es conocido y fiable el conjunto borroso A_i del antecedente de la regla i , tomando los límites del núcleo (zona de validez uno) del conjunto borroso dados por $[c_i^{min}, c_i^{max}]$ (calculados como $c_i^{min} = \min(\text{nucleo}(A_i))$ y $c_i^{max} = \max(\text{nucleo}(A_i))$), para dos reglas (i, j) que se solapan y haciendo:

$$\gamma_i = \frac{1}{2} (\|(a_i^T c_i^{max} + b_i) - (a_j^T c_i^{max} + b_j)\| + \|(a_i^T c_j^{min} + b_i) - (a_j^T c_j^{min} + b_j)\|) \quad (7.14)$$

El resultado de la inferencia alternativa que se propone para la interpolación convexa es imposible distinguirlo de la función original para el ejemplo 5.1 mostrado en la figura 5.1. En el caso de necesitar una interpolación cóncava sería válido el esquema de inferencia tomando $\min_\gamma = -\max_\gamma(-y_1, -y_2)$, con lo que en un esquema completo de inferencia alternativa habría que determinar, en primer lugar, el tipo de aproximación necesaria. Esto se podría hacer, por ejemplo, evaluando el signo de $(y_i - y_j)$ en el núcleo de la función de pertenencia μ_i : positivo indica convexa y negativo cóncava.

Nota 1. Los enfoques global y local para el modelado inteligente coinciden en el caso de funciones de pertenencia que verifiquen la condición de las particiones clásicas (6.11), y lo hacen de forma aproximada en aquellas disposiciones en las que las zonas de transición son pequeñas comparadas con el núcleo ($\mu_{ik} = 1$) de la función de pertenencia.

Nota 2. Los métodos alternativos de desborrosificación (Babuska, 1996), en combinación con una técnica apropiada de identificación de las funciones de pertenencia (poco solape y amplias zonas de alta validez) lleva a soluciones de (7.8) similares a las que proporciona (7.10).

7.3. Técnicas de agrupamiento borroso en identificación de modelos locales

El enfoque global puede alcanzar una mayor precisión entre la salida real del proceso y la estimada, mientras que el enfoque local mejora la interpretabilidad de los modelos identificados. En general, a menor error de modelado menor interpretabilidad de las reglas.

Como se ha comentado anteriormente, los grados de libertad en los μ y f_i empleados en la minimización de los criterios de modelado presentados puede llevar, en algunos casos, a soluciones sin significado “conceptual”: las f_i pueden no tener interpretación como modelos locales y μ puede ser un conjunto de extrañas funciones no convexas a las que sea complicado asignar un sentido. Adicionalmente, podían aparecer problemas de mal condicionamiento si los grados de libertad de las funciones “apuntan en la misma dirección”.

En la identificación con técnicas de inteligencia artificial, el compromiso entre interpretabilidad y error de modelado se debe tener en cuenta ya que el empleo de los

conceptos y modelos depende de qué enfoque se ha elegido. Si el objetivo es la aplicación al diseño de controladores con técnicas lineales o los modelos son diseñados para interpretación humana, la interpretabilidad del usuario es fundamental y el criterio de modelado local parece más adecuado (Díez *et al.*, 2001).

A continuación se discutirá la adecuación para la identificación de modelos borrosos para control de las técnicas de agrupamiento existentes y que se han revisado en el capítulo 6, así como su relación con los índices global y local presentados, haciendo hincapié en la interpretabilidad de las funciones de pertenencia y la correspondencia de los consecuentes de los modelos borrosos TS identificados con linealizaciones en el entorno de los prototipos detectados.

No todos los algoritmos presentados en el capítulo 6 tienen interés en el ámbito de aplicación de esta Tesis. Por ser el primer algoritmo de agrupamiento borroso desarrollado, el método más referenciado en la bibliografía ((Davé and Krishnapuram, 1997), (Hwang and Wooc, 1995)) es el FCM (apartado 6.3). Sin embargo, en el caso de modelado de sistemas no lineales para su posterior aplicación al control del sistema, los métodos de agrupamiento más apropiados si se quieren aplicar técnicas de control lineales a cada grupo de datos serán aquellos que, de una u otra forma, sean capaces de detectar comportamientos lineales: GK (dado que es muy empleado en la literatura para la identificación de modelos lineales), FCRM (por la similitud de su índice de coste con la función de coste local) y AFRCR (por aunar las ventajas de todos los anteriores).

7.3.1. Algoritmos con distancia adaptativa

De entre los algoritmos con distancia adaptativa (revisados en el anexo C.3), cabe destacar por sus buenos resultados el de Gustafson-Kessel (apartado 6.4). Este algoritmo, aunque no tanto como el FCM, se usa bastante en la bibliografía para la obtención de modelos borrosos, dado que las clases hiperelipsoidales que busca detectan de forma bastante correcta los comportamientos quasi-lineales de los diversos regímenes de funcionamiento que pueden existir en un conjunto de datos, a pesar de que su índice de coste (6.30) no se ajusta a lo previsto por un índice local (7.10) sino más bien a un índice global (7.8). Esto lo realiza por la degeneración de las clases hiperelipsoidales si existe un autovalor/autovector poco significativo.

El algoritmo GK es bastante adecuado para el propósito de la identificación, ya que tiene las siguientes propiedades (Babuska, 1996):

- La dimensión de las clases viene limitada por la medida de la distancia y por la definición del prototipo de las clases como un punto.
- En comparación con otros algoritmos, GK es relativamente insensible a la inicialización de la matriz de partición.
- Como el algoritmo está basado en una norma adaptativa, no es sensible al escalado de los datos, con lo que se hace innecesaria la normalización previa de los mismos.

Por todas estas características es bastante empleado en la identificación de los sistemas que nos ocupan. Sin embargo, también tiene sus desventajas:

- La carga computacional es bastante elevada, sobre todo en el caso de grandes cantidades de datos.
- El algoritmo GK puede detectar clases de diferentes formas, no solo subespacios lineales que son los que en principio nos interesan (en realidad busca clases hipereleipsoidales).
- Cuando el número de datos disponibles es pequeño, o cuando los datos son linealmente dependientes, pueden aparecer problemas numéricos ya que la matriz de covarianzas se hace casi singular. El algoritmo GK no podrá ser aplicado a problemas puramente lineales en el caso ideal de no existir ruido.
- Si no hay información al respecto, los volúmenes de las clases se inicializan a valores todos iguales. De esta forma, no se podrán detectar clases con grandes diferencias en tamaño.

7.3.2. Algoritmos con prototipos lineales

De entre los algoritmos con prototipos lineales (revisados en el anexo C.4), el que en un principio parece más interesante es el FCRM, ya que su función de coste (6.36) está basada en la sustitución de D_{ik} en (6.30) (índice de coste de GK) por el error de modelado (6.35), con lo que finalmente coincide con un índice local como el definido en (7.10).

A pesar de lo interesante que en un principio parece el algoritmo, tan solo Kim (Kim *et al.*, 1997) sugiere emplearlo para el modelado borroso. Además, los resultados que obtiene no son muy prometedores, ya que él mismo considera que solo debe ser empleado como una primera aproximación y el ajuste fino del modelo final se hará posteriormente, por ejemplo, con el método del gradiente. En cuanto al antecedente de las reglas lo obtiene en un paso también posterior al agrupamiento, mediante un ajuste de las funciones de pertenencia a funciones exponenciales.

7.3.3. Algoritmos con prototipos mixtos

En el apartado 6.6 se comenta una alternativa para mejorar FCM por combinación de la opción adaptativa y la de prototipos lineales. Allí se revisan los algoritmos fuzzy c-elliptotypes (Bezdek, 1987) (FCE) y adaptive fuzzy c-regression models (Ryoke and Y.Nakamori, 1996) (AFCR), el último de los cuales conjuga en un solo algoritmo ventajas (apartado 6.6.2) de FCRM, FCM y GK. Este algoritmo parece por tanto el idóneo para la identificación de modelos locales.

7.4. Índices para la validación de los modelos locales

La medida del rendimiento global del sistema identificado es habitual en la bibliografía (ver ejemplo 7.1), no siendo tan común la validación de la interpretabilidad de los modelos identificados (Sugeno and Yasukawa, 1993), (Babuska, 1996). En esta sección se presentan unos nuevos índices más apropiados para la validación de los modelos

locales (Ahvenlampi *et al.*, 2003), con la intención de poder realizar posteriormente una análisis no solo cualitativo sino también cuantitativo (apartado 7.5) de los algoritmos que se desee.

7.4.1. Índice de compacidad

Este índice sirve para estudiar la forma de las funciones de pertenencia en lo que respecta a su compacidad. Se entiende que una función de pertenencia tiene una forma compacta si las altas pertenencias se dan en el entorno del prototipo y las bajas pertenencias lejos del prototipo, sin extenderse demasiado las zonas de pertenencia no nula en el universo de discurso.

El índice calcula el producto entre la pertenencia asociada a cada dato y su distancia al centro (prototipo) de la clase, realizando la suma de todos ellos, mediante:

$$IC_i = \sum_{k=1}^N \mu_{ik}(z_k - c_i)^2 \quad (7.15)$$

donde IC_i es el índice de compacidad para la clase i , N es el número de datos, z_k es un dato y c_i el prototipo. Para tener una idea intuitiva, se puede pensar que este cálculo equivale a calcular el momento de inercia del contorno de la función de pertenencia en torno al eje perpendicular al hiperplano $(n-1)$ -dimensional del universo de discurso y que pasa por el prototipo.

Nota 3. Con el fin de hacer comparables los valores numéricos de los índices de compacidad de diferentes clases, o de la misma clase pero identificada con diferentes algoritmos, puede ser conveniente dividir el valor de IC_i por el número de puntos asignados a la clase con pertenencia no nula (o mayor de cierto valor).

Finalmente, se puede calcular un índice de compacidad del sistema completo realizando la suma de IC_i para los c clases:

$$IC = \sum_{i=1}^c \sum_{k=1}^N \mu_{ik}(z_k - c_i)^2 \quad (7.16)$$

Estos índices pueden ser calculados para el α -corte de las funciones de pertenencia que se desee. En cualquier caso, a menor valor del IC mejores serán los modelos locales en el sentido de la compacidad.

7.4.2. Índice de solapamiento

En este caso se propone una medida para conocer si las funciones de pertenencia están poco solapadas o no. Para ello, en primer lugar se reordena la matriz de partición μ_{ik} de forma que cada una de las columnas tenga las pertenencias ordenadas de forma descendente en una matriz μ_{jk}^{IS} , siendo la primera fila de la columna j la mayor pertenencia a una clase de dicho dato, la segunda fila la segunda mayor pertenencia del dato k a una clase y así sucesivamente hasta llegar a la menor pertenencia del dato k

a una clase que se encontrará en la fila j . Evidentemente, los índices i y j no tienen relación alguna, ya que j indica el orden de importancia de una pertenencia e i es una clase concreta.

Una vez obtenida la matriz μ_{jk}^{IS} se procede a calcular el j -ésimo índice de solapamiento como la media de los valores de la fila correspondiente j con:

$$IS_j = \frac{\sum_{k=1}^N \mu_{jk}^{IS}}{N} \quad (7.17)$$

quedando toda la información almacenada en la matriz columna IS con c filas:

$$IS = [IS_1, IS_2, \dots, IS_c]^T \quad (7.18)$$

El análisis de este índice matricial se realiza por comparación de las diversas entradas de IS . Un elevado valor de IS_1 y bajos valores en el resto de términos indica poco solapamiento. En cambio, valores similares en todas las entradas de IS indican un elevado solapamiento.

7.4.3. Índice de funcionamiento local

La mayoría de estudios de validación en identificación emplean un índice de funcionamiento basado en el error global (Sugeno and Yasukawa, 1993) dado por el índice de funcionamiento:

$$IF = \frac{\sum_{k=1}^N (y_k - \hat{y}_k)^2}{N} \quad (7.19)$$

en el que y_k es la salida real e \hat{y}_k es la estimación del modelo. Este índice está directamente relacionado con el índice global J_g establecido en la ecuación 7.8. Sin embargo, si se pretende evaluar la idoneidad de los modelos desde el punto de vista del índice local J_i^l definido en 7.10, conviene definir el índice de funcionamiento local:

$$IF_i = \sum_{k=1}^N \mu_{ik} (y_k - \hat{y}_{ik})^2 \quad (7.20)$$

que indica la adecuación de cada modelo local a los puntos de la clase a la que corresponde.

7.5. Análisis de algoritmos de agrupamiento

En este apartado se realizará un estudio comparativo de los algoritmos de agrupamiento que, de entre los seleccionados por sus características como más apropiados en la tabla 6.1, se han mostrado interesantes (apartado 7.3) en el ámbito de esta Tesis. Para que sea posible realizar un análisis cualitativo, además del cuantitativo basado en los índices definidos en el apartado anterior, se definirán unos conjuntos de datos que

permiten la visualización de los resultados (modelos locales y funciones de pertenencia identificadas).

El análisis servirá para determinar, desde puntos de vista distintos al error de predicción visto en el apartado 7.1, el mejor algoritmo de agrupamiento en el ámbito de aplicación deseado (modelado TS para control por modelos locales) y/o para proponer las características deseables de un algoritmo específico.

Ejemplo 7.2. Caso 2: Parábola. Se propone emplear como ejemplo la identificación de modelos locales que se ajusten a la función no lineal que corresponde a una parábola, por aunar sencillez en su no linealidad (posee una variación lineal de su derivada) y facilidad de representación. Se ha añadido ruido a los datos con el fin de poder evaluar el comportamiento de los algoritmos frente en esta circunstancia. La función a identificar aparece en la figura 7.3 y se identificarán 3 clases. Una identificación ideal

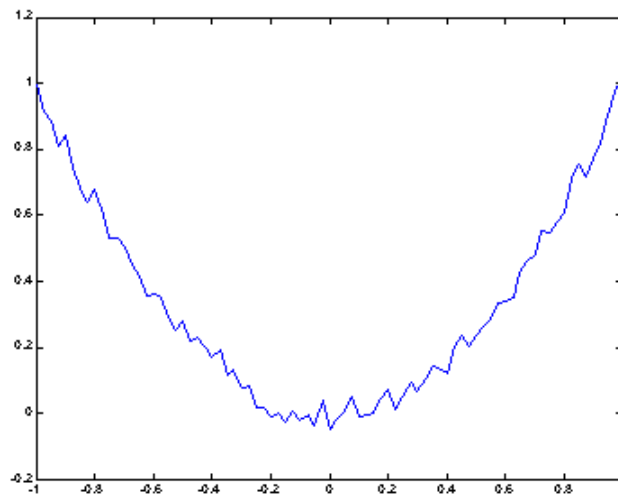


Figura 7.3: Datos de la parábola con ruido para el caso 2.

llevaría a unos modelos locales identificados que se ajustasen a la función a identificar y con unas funciones de pertenencia suaves y convexas (y por tanto interpretables). Las gráficas siguientes corresponden a los resultados que se obtienen con los métodos de agrupamiento en estudio: FCM (figura 7.4), GK (figura 7.5), FCRM (figura 7.6) y AFMR (figura 7.7).

Las figuras indican que los resultados obtenidos por GK y AFMR son casi idénticos y la compacidad y solapamientos de las funciones de pertenencia son adecuadas. Igualmente, los modelos locales se ajustan de forma correcta a la función a identificar. FCM, en cambio, tiene unos modelos locales ligeramente peores y unas funciones de pertenencia con mayor solapamiento. Los peores resultados se dan con FCRM, donde ni las funciones de pertenencia ni los modelos locales son apropiados.

Los resultados de los índices definidos en el apartado anterior deberán corroborar las impresiones mencionadas. En la tabla 7.2 se muestran los resultados para el índice de compacidad dado por la ecuación (7.16), la tabla 7.3 los del índice de solapamiento

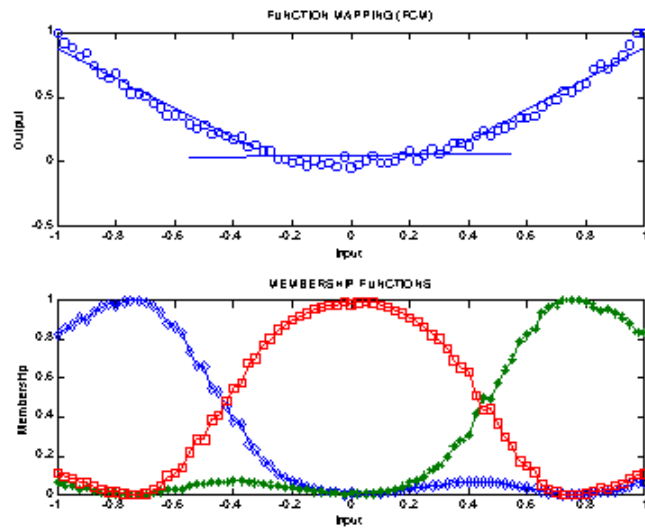


Figura 7.4: Identificación con FCM de modelos locales en la parábola del caso 2.

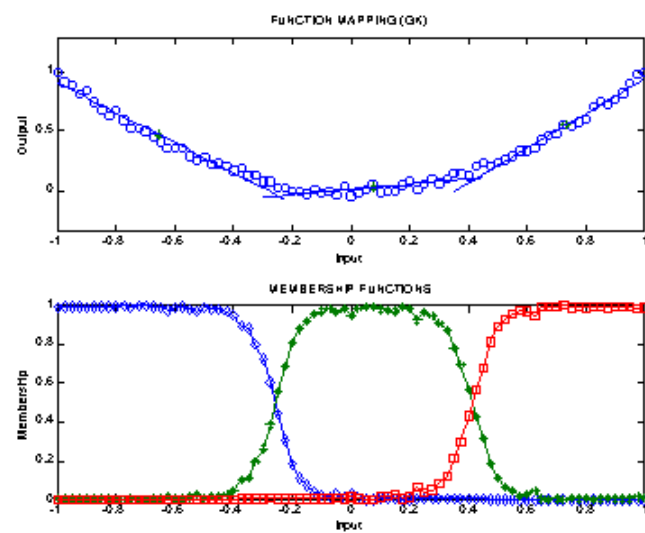


Figura 7.5: Identificación con GK de modelos locales en la parábola del caso 2.

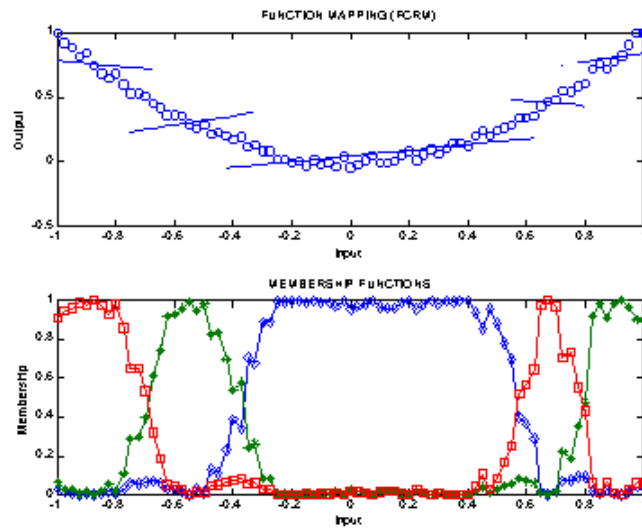


Figura 7.6: Identificación con FCRM de modelos locales en la parábola del caso 2.

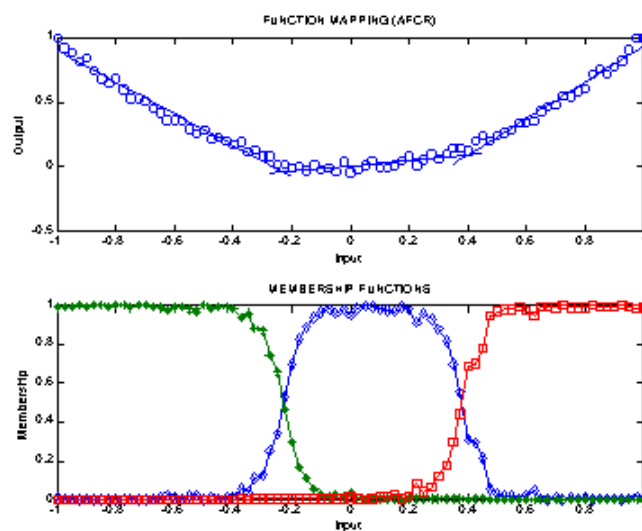


Figura 7.7: Identificación con AFMR de modelos locales en la parábola del caso 2.

dado por la ecuación (7.18) y la tabla 7.4 los del índice de funcionamiento local dado por la ecuación (7.20) junto al índice de funcionamiento global que habitualmente emplea la literatura (7.19).

Método de agrupamiento	Indice de compacidad (IC)
FCM	9.5308
GK	3.9629
FCRM	11.8673
AFCR	3.8664

Tabla 7.2: Resultados del índice de compacidad para el caso 2.

Método de agrupamiento	Indice de solapamiento ($IS_3; IS_2; IS_1$)
FCM	(0.0325; 0.1215; 0.8460)
GK	(0.0027; 0.0660; 0.9313)
FCRM	(0.0166; 0.1131; 0.8703)
AFCR	(0.0023; 0.0592; 0.9385)

Tabla 7.3: Resultados del índice de solapamiento para el caso 2.

Método de agrupamiento	Indices locales	IF
FCM	0.257 0.058 0.064	0.0127
GK	0.066 0.047 0.042	0.0022
FCRM	0.024 0.084 0.132	0.0032
AFCR	0.035 0.044 0.073	0.0019

Tabla 7.4: Resultados de los índices de funcionamiento local y global para el caso 2.

Los resultados de IC e IS muestran que los mejores resultados se obtienen para los algoritmos AFCR y GK, mientras que FCRM y FCM obtienen los peores. En cuanto a los índices locales, nos dan una idea de la adecuación a los datos experimentales de cada uno de los submodelos identificados en el proceso de agrupamiento por todos los métodos, lo que nos permite discernir (si todos los errores locales son pequeños y del mismo orden) que solo son modelos correctos los de AFCR y GK. Sin embargo, los resultados de los índices globales indican que, por ejemplo, el resultado de FCRM no es demasiado malo lo que justifica la necesidad de todos los índices definidos para conocer la adecuación de un algoritmo.

A pesar de que el análisis preliminar de las funciones de coste podría presuponer un buen comportamiento de FCRM y aunque el error entre los datos de salida y los

del modelo se minimiza, los modelos lineales obtenidos no son nada parecidos a lo esperado. Diferentes ejecuciones del algoritmo con inicialización de la matriz de partición U más cercana al resultado deseado llevan a mejores resultados, pero esta información no está generalmente disponible.

Aunque FCRM no obtiene los resultados deseados su filosofía es buena: obtiene los modelos locales en el proceso de agrupamiento e incluye el error de modelado en el proceso iterativo mediante el índice de coste. El problema es que la función objetivo definida tiene demasiados grados de libertad y algunas soluciones no son de la forma esperada, con lo que algunas restricciones a la solución se podrían incluir de forma que el error local (de la forma en que nos interese) sea más importante que el error global.

GK y AFMR obtienen buenos y similares resultados. Sin embargo, GK tiene una desventaja común a muchos algoritmos de agrupamiento empleados en la identificación de modelos borrosos: olvidar su cometido (el modelo borroso) en la etapa de agrupación de los datos. Siempre es un paso posterior derivar reglas en alguno de los tipos de modelo borroso existentes (Takagi-Sugeno en este caso).

En cambio, FCRM y AFMR si que tienen esta consideración en cuenta y dado que los resultados de AFMR se corresponden con la mejora indicada en (Ryoke and Y.Nakamori, 1996) y (Hathaway and Bezdek, 1993) respecto a FCRM y GK:

- iguala los resultados de GK,
- obtiene, al tiempo de realización del agrupamiento, una serie de modelos locales lineales que se ajustan perfectamente a la función en estudio, e
- incluye el error de modelado en la función de coste.

se puede concluir que, en este ejemplo, AFMR es el que tiene un mejor comportamiento. Además, la capacidad de detección de estructuras lineales de FCRM no se ha perdido en AFMR, obteniendo resultados idénticos a FCRM para diferentes conjuntos de datos ensayados.



Ejemplo 7.3. Caso 3: Ratio de cinética de crecimiento de un biorreactor.

Para los datos (Díez, 1999) del ratio de cinética de crecimiento de un biorreactor (en horas^{-1}) en función de la concentración de sustrato (en g/l) que aparece en la figura 7.8 se identificarán 4 clases, ya que por inspección de la figura parece adecuado. Este caso es interesante por corresponder a datos experimentales reales de una función no lineal y con tramos lineales que los algoritmos de agrupamiento deberían detectar. Las gráficas correspondientes que se obtienen con los métodos de agrupamiento son: FCM (figura 7.9), GK (figura 7.10), FCRM (figura 7.11) y AFMR (figura 7.12).

Las figuras indican, de nuevo, que los resultados obtenidos por GK y AFMR son casi idénticos y la compacidad y solapamientos de las funciones de pertenencia son adecuadas. Igualmente, los modelos locales se ajustan de forma correcta a la función a identificar. FCM, en cambio, tiene unos modelos locales ligeramente peores y unas funciones de pertenencia con mayor solapamiento. Los peores resultados se dan con FCRM, donde ni las funciones de pertenencia ni los modelos locales son apropiados.

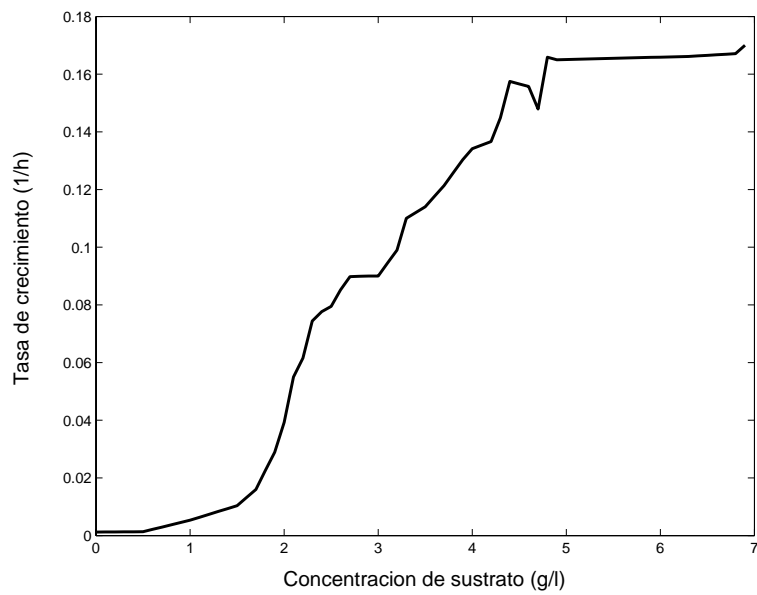


Figura 7.8: Datos de la función para el caso 3.

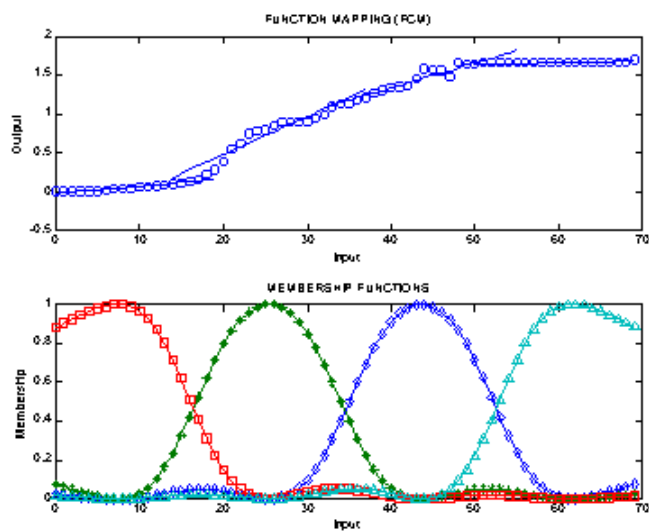


Figura 7.9: Identificación con FCM de modelos locales de la función del caso 3.

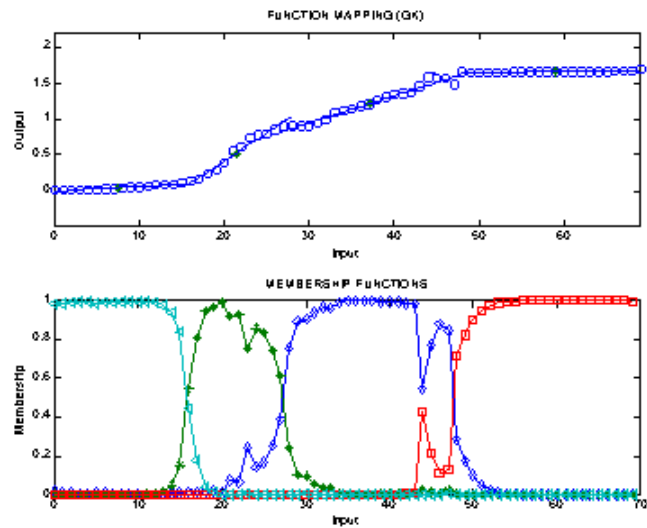


Figura 7.10: Identificación con GK de modelos locales de la función del caso 3.

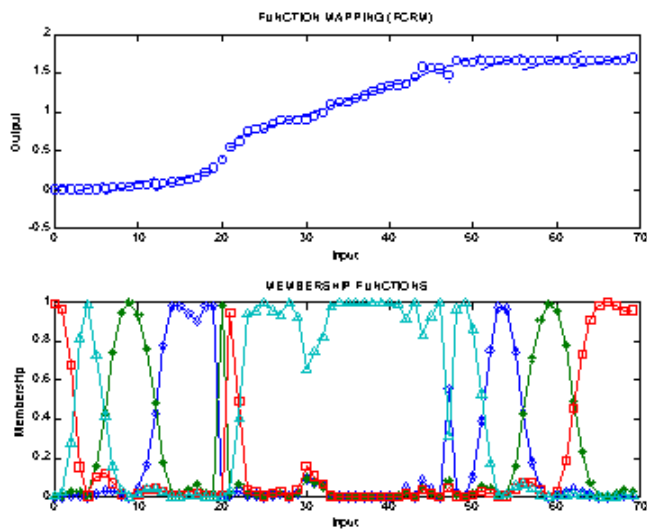


Figura 7.11: Identificación con FCRM de modelos locales de la función del caso 3.

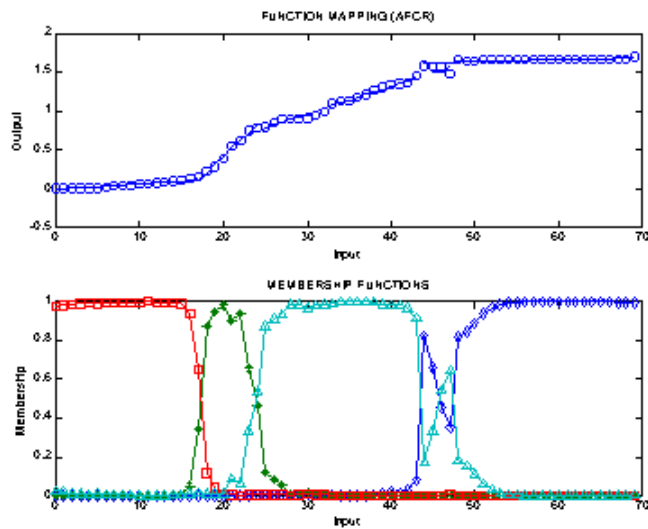


Figura 7.12: Identificación con AFCR de modelos locales de la función del caso 3.

En cuanto a los resultados de los índices de validez, la tabla 7.5 muestra los resultados para el índice de compacidad, la tabla 7.6 muestra los del índice de solapamiento y la tabla 7.7 los resultados para el índice de funcionamiento local dado por la ecuación (7.20) junto al índice de funcionamiento global que habitualmente emplea la literatura (7.19).

Método de agrupamiento	Índice de compacidad (IC)
FCM	5.1429e+003
GK	2.6865e+003
FCRM	2.8175e+004
AFCR	3.0669e+003

Tabla 7.5: Resultados del índice de compacidad para el caso 3.

Método de agrupamiento	Índice de solapamiento ($IS_4; IS_3; IS_2; IS_1$)
FCM	(0.0112; 0.0237; 0.1182; 0.8469)
GK	(0.0004; 0.0037; 0.0637; 0.9322)
FCRM	(0.0122; 0.0250; 0.1278; 0.8350)
AFCR	(0.0010; 0.0023; 0.0570; 0.9398)

Tabla 7.6: Resultados del índice de solapamiento para el caso 3.

Método de agrupamiento	Índices locales				IF
FCM	0.079	0.134	0.043	0.015	0.0027
GK	0.008	0.025	0.067	0.007	0.0013
FCRM	0.009	0.021	0.016	0.026	9.4e-004
AFCR	0.004	0.004	0.027	0.015	5.2e-004

Tabla 7.7: Resultados de los índices de funcionamiento local y global para el caso 3.

Con este conjunto de datos los mejores resultados de IC e IS se obtienen para los algoritmos AFCR y GK, mientras que FCRM y FCM obtienen los peores. En este caso los índices locales nos permite discernir (al ser todos los errores locales bajos y del mismo orden) que son modelos correctos los de AFCR, GK y FCRM. Este último consigue esa buena valoración gracias a la extraña forma de sus funciones de pertenencia, haciendo que en pequeños tramos inconexos sus “linealizaciones” sean cercanas a los datos experimentales. Además, los resultados de los índices globales indican que, por ejemplo, el resultado de AFCR y FCRM son los mejores, con lo que se concluye la necesidad de todos los índices para un estudio correcto.

Los resultados de GK y AFCR superan a FCRM, pero cabe destacar que en ambos casos de resultado positivo (AFCR y GK) las funciones de pertenencia borrosas obtenidas pierden parte de su interpretabilidad. Si se centra la atención en la forma de las funciones de pertenencia no se pueden considerar los resultados como satisfactorios, ya que GK y AFCR pueden producir formas arbitrarias no-convexas, perdiendo la interpretabilidad a cambio de un bajo error de modelado.

En cualquier caso se concluye de nuevo, por las mismas consideraciones que en el ejemplo anterior, que AFCR mejora a GK.

■

7.6. Conclusiones

El estudio que se ha llevado a cabo para comprobar la adecuación de los algoritmos de agrupamiento en el marco de la identificación automatizada de modelos borrosos para control, tanto en un análisis cuantitativo como cualitativo, ha mostrado que los algoritmos de la literatura que han sido revisados no resuelven el problema completamente.

El problema fundamental estriba en que en la identificación con técnicas de inteligencia artificial el compromiso entre interpretabilidad y error de modelado se debería tener en cuenta, ya que el empleo de los conceptos y modelos depende de qué enfoque (global o local) se ha elegido. Además, con el objetivo de la aplicación al diseño de controladores el criterio de modelado local parece en principio más adecuado y no siempre se aplica.

Los problemas encontrados en la mayoría de algoritmos de agrupamiento existentes sugieren plantear un nuevo algoritmo (una alternativa se puede ver en (Johansen and Babuska, 2002)) en el que:

- los modelos locales lineales se obtuviesen en el proceso de agrupamiento,
- información sobre el error de modelado local estuviese incluida en las iteraciones del algoritmo,
- se mejorase la interpretabilidad de las reglas, por ejemplo, favoreciendo la convexidad de las funciones de pertenencia, y
- se pudiese establecer un balance entre interpretabilidad y error de modelado.

La filosofía de la familia de algoritmos que provienen de FCRM y, más concretamente, su versión adaptiva AFCR permite:

- igualar o mejorar los resultados de otros algoritmos adaptativos,
- obtener, al tiempo de realización del agrupamiento, una serie de modelos locales lineales que se ajusten perfectamente a la función en estudio, e
- incluir el error de modelado en la función de coste.

Sin embargo, aunque tiene características de las consideradas deseables, como en el resto de algoritmos estudiados puede producir formas arbitrarias no-convexas en las funciones de pertenencia identificadas, con lo que la interpretabilidad se pierde a cambio de un bajo error de modelado.

Una inspección de la matriz de partición borrosa U definitiva a la que lleva AFCR, nos indica que se minimiza el error no por cercanía del modelo obtenido, sino por la ponderación de varios modelos. Sin embargo, lo que se pretende buscando modelos cuyo fin es el control, son modelos locales lineales que se aproximen lo más posible al modelo del sistema: se pretende “parchar” la función desconocida que representa al sistema con hiperplanos tangentes a la hipersuperficie a identificar.

La línea que se propone seguir para conseguir un buen algoritmo será incluir en el proceso de agrupamiento de AFCR condiciones para:

- mejorar la interpretabilidad de las reglas favoreciendo la convexidad de las funciones de pertenencia, y
- establecer un balance entre interpretabilidad y error de modelado.

De esta forma se mejorarán las técnicas de agrupamiento existentes en el sentido de la interpretabilidad y orientación a control, mientras que las características positivas del agrupamiento (adición de clases, fusión de clases, etc) serán aún posibles.

Con todo lo anterior, las aportaciones de este capítulo se pueden resumir en:

- Propuesta de un método sistematizado de identificación basado en técnicas agrupamiento y determinación de su aplicabilidad,
- Análisis general de la idoneidad de las funciones de coste de los algoritmos de identificación para control,
- Determinación de la función de coste adecuada en los algoritmos de agrupamiento en su aplicación a la determinación de modelos locales,

- Definición de unos índices para la evaluación objetiva del comportamiento de los algoritmos de agrupamiento,
- Aplicación de los índices anteriores en el análisis cuantitativo de algoritmos de agrupamiento,
- Análisis cualitativo de los algoritmos de agrupamiento existentes en la identificación de modelos locales para control,
- Propuesta de algoritmo de agrupamiento a emplear y modificaciones a realizar sobre el mismo para obtener un algoritmo adecuado en el marco de la Tesis.

En el capítulo 8 se propondrá una forma para incluir en el proceso de agrupamiento alguna condición que favorezca la interpretabilidad (convexidad): entradas de U (pertenencias) cercanas a 1 en la cercanía del prototipo de la clase y cercanas a 0 para puntos lejos del prototipo. Igualmente se procurará la posibilidad establecer un balance entre interpretabilidad y error de modelado.

Capítulo 8

Propuesta de una técnica mejorada de agrupamiento

*It is better to fail in originality
than to succeed in imitation.*
Herman Melville.

Tal y como se estableció en las conclusiones del capítulo 7, el objetivo primordial de un algoritmo de agrupamiento con la finalidad de identificar modelos borrosos para control sería desarrollar clases lineales o hiperplanos correspondientes a linealizaciones en el entorno de los prototipos (consecuentes afines), manteniendo la interpretabilidad de las reglas (antecedentes con formas adecuadas).

En este capítulo se desarrollará una función objetivo modificada y un nuevo algoritmo de agrupamiento que será más adecuado para el modelado de sistemas no lineales con modelos borrosos TS para control. La validación de su funcionamiento se realizará en base a su aplicación a diferentes ejemplos.

Se pretende por tanto en este capítulo conseguir la convexidad de las funciones de pertenencia borrosas suministradas en el proceso de identificación así como la correspondencia de los modelos locales con linealizaciones en el entorno de los prototipos. En capítulos posteriores se mostrará la aplicación al diseño de controladores de estos modelos identificados.

8.1. Un índice que favorece la convexidad

La modificación de la función objetivo empleada en el algoritmo AFGR (6.39) incluyendo criterios apropiados para aumentar la convexidad de las funciones de pertenencia μ , lleva a mejoras del comportamiento respecto al original en cuanto a la interpretabilidad de las funciones de pertenencia. Por supuesto, las mejoras proporcionadas por estos términos adicionales serán a costa de un posible aumento en el error de modelado si el número de clases se conserva (Díez *et al.*, 2002a).

Con el fin de aumentar la convexidad, se añaden dos términos adicionales al índice de coste definido en la ecuación (6.39): un primer término para penalizar las altas pertenencias de puntos lejanos al prototipo ($J_{lejosik}$) y un segundo término que penaliza

la baja pertenencia de puntos cercanos al prototipo ($J_{cerca\ ik}$):

$$J_{lejos\ ik} = \sum_{i=1}^c \sum_{k=1}^N \mu_{ik}^m [1 - \exp(-\frac{L_{ik}^2}{2\sigma_1^2})] \quad (8.1)$$

$$J_{cerca\ ik} = \sum_{i=1}^c \sum_{k=1}^N (1 - \mu_{ik})^m [\exp(-\frac{L_{ik}^2}{2\sigma_2^2})] \quad (8.2)$$

$$L_{ik}^2 = (z_k - p_i)^T B (z_k - p_i) \quad (8.3)$$

donde las distancias L_{ik} y los prototipos p_i son medidos en el espacio de entrada, conservando la matriz B la definición del algoritmo AFMR. $J_{lejos\ ik}$ resultará en un valor elevado en el índice de coste cuando se de una alta pertenencia de un punto lejano al prototipo gracias al producto de su pertenencia μ_{ik} por un valor elevado si la distancia al prototipo es grande $[1 - \exp(-\frac{L_{ik}^2}{2\sigma_1^2})]$. De forma opuesta, $J_{cerca\ ik}$ elevará su importancia en el índice de coste si hay una baja pertenencia de puntos cercanos al prototipo debido al producto del complementario de la pertenencia $(1 - \mu_{ik})$ por un valor elevado si la distancia al prototipo es pequeña $[\exp(-\frac{L_{ik}^2}{2\sigma_2^2})]$. En ambos casos se han empleado funciones exponenciales, aunque la elección de cualquier otra función de base radial llevará a resultados similares.

Los valores de σ_1 y σ_2 están relacionados con la distancia máxima del entorno del prototipo para la que deben estar la mayoría de los puntos de alta pertenencia (σ_2) y la distancia máxima a partir de la cual no deben haber altas pertenencia a la clase (σ_1). En el apartado 8.5 se clarifica este significado y se propone como determinar sus valores.

Dos nuevos parámetros γ_1 y γ_2 se han incluido, respectivamente, para premultiplicar $J_{lejos\ ik}$ y $J_{cerca\ ik}$ con el propósito de establecer un balance entre los diferentes términos del índice de coste, al igual que η proporciona en el algoritmo AFMR original (apartado 6.6.2). El parámetro α definido en (6.41) también se ha incluido siguiendo para su actualización el método de (6.40) (todos los términos α deben sumar la unidad) y dando a todos los términos el mismo peso.

La minimización del nuevo índice de funcionamiento:

$$\begin{aligned} J_C\ ik(Z; U, C) = & \sum_{i=1}^c \sum_{k=1}^N (\mu_{ik})^m \frac{\alpha_i}{3} E_{ik}(\beta_i) + \\ & \sum_{i=1}^c \sum_{k=1}^N (\mu_{ik})^m (1 - \alpha_i) \eta D_{ik}^2 + \\ & \sum_{i=1}^c \sum_{k=1}^N \mu_{ik}^m \frac{\alpha_i}{3} \gamma_1 [1 - \exp(-\frac{L_{ik}^2}{2\sigma_1^2})] + \\ & \sum_{i=1}^c \sum_{k=1}^N (1 - \mu_{ik})^m \frac{\alpha_i}{3} \gamma_2 [\exp(-\frac{L_{ik}^2}{2\sigma_2^2})] \end{aligned} \quad (8.4)$$

llevará a una solución con funciones de pertenencia más suaves y con formas más convexas. Como suele ocurrir en este tipo de funciones, la determinación de parámetros es un problema adicional a resolver previamente a la minimización del índice de coste. El usuario final del algoritmo debe emplear en esa etapa de ajuste los significados correctos de cada parametro:

- σ_1 está relacionado con el tamaño máximo permitido para cada clase,
- σ_2 está relacionado con el mínimo tamaño deseable para una clase,
- η proporciona un balance entre la importancia de la forma de clase y el error de modelado en las primeras etapas de la minimización,
- γ_1 y γ_2 ponderan los nuevos términos del índice respecto al resto de términos ya existentes en el índice del algoritmo AFCR.

8.2. Minimización del nuevo índice de coste

Aunque se podrían emplear otras alternativas de minimización de $J_{C\ ik}$ definido en (8.4), la opción que se va a emplear seguirá los pasos de la que típicamente se emplea en la minimización de índices derivados del algoritmo original FCM descrito en el apartado 6.3.

Para realizar la minimización es más conveniente reescribir (8.4) con una notación más compacta:

$$J_{C\ ik}(Z; U, C) = \sum_{i=1}^c \sum_{k=1}^N (\mu_{ik})^m D_I(i, k) + \sum_{i=1}^c \sum_{k=1}^N (1 - \mu_{ik})^m D_{II}(i, k) \quad (8.5)$$

donde:

$$\begin{aligned} D_I(i, k) &= \frac{\alpha_i}{3} E_{ik}(\beta_i) + (1 - \alpha_i) \eta D_{ik}^2 + \frac{\alpha_i}{3} \gamma_1 \left[1 - \exp\left(-\frac{L_{ik}^2}{2\sigma_1^2}\right) \right] \\ D_{II}(i, k) &= \frac{\alpha_i}{3} \gamma_2 \left[\exp\left(-\frac{L_{ik}^2}{2\sigma_2^2}\right) \right] \end{aligned} \quad (8.6)$$

La minimización de (8.5) debe hacerse en dos pasos independientes como es sugerido en (Bezdek, 1987) para FCM (apartado 6.3). En primer lugar, el índice debe ser minimizado para U con C fijo para cada uno de los datos $w_k = (z_k, y_k)$ (7.7). Esto da como resultado un problema de minimización con restricciones (μ_{ik} deben sumar la unidad) que puede ser resuelto con la aplicación de los multiplicadores de Lagrange. El empleo de esta técnica lleva a un nuevo índice de coste (8.7):

$$\begin{aligned} H_k(\lambda, \mu_k) &= \sum_{i=1}^c (\mu_{ik})^m D_I(i, k) + \\ &\quad \sum_{i=1}^c (1 - \mu_{ik})^m D_{II}(i, k) - \lambda \left(\sum_{i=1}^c \mu_k - 1 \right) \end{aligned} \quad (8.7)$$

y haciendo su gradiente igual a cero lleva a una expresión de actualización para las funciones de pertenencia dada por:

$$\mu_{ik} = \frac{1 - \sum_{j=1}^c \frac{-D_{II}(j,k) + D_{II}(i,k)}{D_I(j,k) + D_{II}(j,k)}}{\sum_{j=1}^c \frac{D_I(i,k) + D_{II}(i,k)}{D_I(j,k) + D_{II}(j,k)}} \quad (8.8)$$

El segundo paso para la minimización de (8.5) es minimizar para C con U fija para cada clase. Esto lleva a un problema de minimización, esta vez sin restricciones, del índice:

$$H_i = \sum_{k=1}^N (\mu_{ik})^m D_I(i, k) + \sum_{k=1}^N (1 - \mu_{ik})^m D_{II}(i, k) \quad (8.9)$$

Esta minimización se hace anulando las derivadas adecuadas de H_i y lleva a la expresión de actualización de los prototipos:

$$c_i = \frac{\sum_{k=1}^N \mu_{ik}^m K_I w_k + (1 - \mu_{ik})^m K_{II} w_k}{\sum_{k=1}^N \mu_{ik}^m K_I + (1 - \mu_{ik})^m K_{II}} \quad (8.10)$$

donde:

$$\begin{aligned} K_I(i, k) &= \frac{\alpha_i}{3} + (1 - \alpha_i)\eta + \frac{\alpha_i}{3}\gamma_1 \frac{\exp(-\frac{L_{ik}^2}{2\sigma_1^2})}{2\sigma_1^2} \\ K_{II}(i, k) &= -\frac{\alpha_i}{3}\gamma_2 \frac{\exp(-\frac{L_{ik}^2}{2\sigma_2^2})}{2\sigma_2^2} \end{aligned} \quad (8.11)$$

De esta forma se han obtenido las ecuaciones de actualización de parámetros necesarias para implementar un algoritmo de agrupamiento apropiado a la minimización de (8.4), cuyos pasos se describen en el apartado siguiente.

8.3. Algoritmo de agrupamiento para la mejora de la convexidad

Con las expresiones desarrolladas en el punto anterior, se presenta un nuevo algoritmo de agrupamiento que se denominará **AFCRC** (Adaptive Fuzzy C-Regression models with Convexity enhancement). Dicho algoritmo se implementará realizando iteraciones con las expresiones de actualización (8.8) y (8.10).

Dado un conjunto de datos $Z = \{(z_1, y_1), \dots, (z_N, y_N)\}$, seleccionados un número $c \in [1, N]$ de clases, un exponente de ponderación $m > 1$ y una tolerancia $\varepsilon > 0$ para la terminación:

1. Fijar los parámetros definidos por el usuario: η , σ_1 , σ_2 , γ_1 y γ_2 .
2. Inicializar la matriz de partición borrosa U y la de distancia L con valores aleatorios $U^{(0)}$ y $L^{(0)}$, respectivamente, haciendo $l=1$ (índice de iteración).
3. Calcular los centroides de las clases con (8.10).
4. Calcular las distancias:

$$D_{ik}^2 B_1 = (z_k - c_i^{(l)})^T B_1 (z_k - c_i^{(l)}), \quad 1 \leq i \leq c \quad 1 \leq k \leq N$$

$$L_{ik}^2 B_2 = (z_k - c_i^{(l)})^T B_2 (z_k - c_i^{(l)}), \quad 1 \leq i \leq c \quad 1 \leq k \leq N$$

con B_1 y B_2 matrices identidad de dimensiones N y $N-1$ respectivamente

5. Calcular la matriz de covarianzas de cada clase:

$$F_i^{(l)} = \frac{\sum_{k=1}^N (\mu_{ik}^{(l-1)})^m (z_k - c_i^{(l)})(z_k - c_i^{(l)})^T}{\sum_{k=1}^N (\mu_{ik}^{(l-1)})^m}, \quad 1 \leq i \leq c$$

6. Calcular autovalores λ_{il} de cada matriz de covarianzas cada $F_i^{(l)}$, así como los diferentes α_i :

$$\alpha_i = 1 - \frac{\min_l \{\lambda_{il}\}}{\max_l \{\lambda_{il}\}}, \quad i = 1, 2, \dots, c$$

7. Calcular valores para los parámetros β_i^l por mínimos cuadrados ponderados con $U^{(l)}$.
8. Recalcular la matriz de partición para todo $1 \leq i \leq c$, $1 \leq k \leq N$, tomando (8.8). Si $E_{ik} = 0$ para algún $i = s$, hacer $\mu_{sk} = 1$ y $\mu_{ik} = 0$ para $\forall i \neq s$.
9. Si $\|U^{(l)} - U^{(l-1)}\| < \varepsilon$, parar. Si no, volver al tercer paso del algoritmo.

Las funciones de pertenencia que se obtienen con el algoritmo anterior son, gracias a los nuevos términos en el índice de coste, más interpretables y las clases son más similares a la estructura deseada por un ingeniero de control. Todo esto se comprobará en apartados posteriores para diferentes ejemplos.

En lo que respecta al primer paso del algoritmo dado que γ_1 y γ_2 representan, en las últimas iteraciones, la importancia relativa de la interpretabilidad frente al error de modelado, su determinación debe ser supervisada por el usuario. En el apartado 8.5 se propone una estructura de trabajo mediante la cual el usuario del algoritmo puede obtener una buena estimación de los parámetros.

Extensiones basadas en AFCRC. El nuevo índice de coste propuesto genera toda una nueva familia de algoritmos de agrupamiento que sería conveniente estudiar ya que, como a todos los algoritmos de agrupamiento, se le pueden aplicar las filosofías de unión de clases compatibles para evitar redundancias (apartado C.8.1), particiones posibilísticas para robustez ante ruido (apartado C.7), etc.

Aunque los algoritmos posibilísticos no obtienen directamente una partición borrosa (se podrían normalizar posteriormente las funciones de pertenencia obtenidas) y hasta puede ser un contrasentido ligar una definición posibilista con el error de modelado (una baja μ anula dicho error), se presenta a modo de ejemplo esta modificación al algoritmo original.

Empleando la partición posibilística en AFCRC (de la misma forma que se propone en el apartado C.7.3 con FCM llegando a PCM), el índice a minimizar en este caso sigue siendo (8.6). La comparación de (8.6) con (C.55) muestra una gran similitud. La minimización de (8.6) se realizará de la misma forma (en dos pasos) que en el apartado

8.2, siendo en esta caso una minimización sin restricciones y no siendo necesario el empleo de los multiplicadores de Lagrange.

Haciendo el gradiente respecto a U igual a cero con C fija lleva a una expresión de actualización para las funciones de pertenencia:

$$\mu_{ik} = \frac{1}{1 + \left(\frac{D_I(i,k)}{D_{II}(i,k)}\right)^{2/(m-1)}} \quad (8.12)$$

El segundo paso para la minimización para C con U fija para cada clase lleva a la expresión de actualización de los prototipos:

$$c_i = \frac{\sum_{k=1}^N \mu_{ik}^m w_k}{\sum_{k=1}^N \mu_{ik}^m} \quad (8.13)$$

El algoritmo de agrupamiento será idéntico al descrito en 8.3 sustituyendo las expresiones de actualización de c_i y μ_{ik} por las que se acaban de obtener.

8.4. Efecto de los nuevos términos en el algoritmo AFCRC

En este apartado se presentan una serie de simulaciones en las que se aprecia el efecto de las variaciones de los parámetros σ_1 y σ_2 que ha de definir el usuario en el algoritmo AFCRC.

Para poder hacer comparables los resultados del algoritmo ante diferentes valores de σ_1 y σ_2 , considerando que los valores numéricos finales que resultan de aplicar el índice (8.4) son:

$$J_{AFCRC} = J_E + J_D + J_{lejos} + J_{cerca} \quad (8.14)$$

se intentará en todas las ejecuciones (variando los valores de γ_1 y γ_2) que los términos de interpretabilidad $J_{lejos} + J_{cerca}$ sean el 10% del total del error J_{AFCRC} , dejando el 90% para los componentes del índice de AFCR $J_{AFCR} = J_E + J_D$ correspondientes al error de modelado J_E y a la forma apropiada de la clase J_D .

Con el fin de apreciar mejor el efecto de los parámetros del nuevo algoritmo se ha tomado un ejemplo de los tratados en el capítulo 7: el conjunto de datos del caso 2 (parábola con ruido). AFCRC se comporta para diferentes valores de σ_1 de la forma en que se observa en las figuras 8.1 y 8.2. Para las primeras pruebas, con el fin de eliminar el efecto del término σ_2 se hará $\gamma_2 = 0$.

Como se puede observar el efecto no es muy apreciable a simple vista, aunque un análisis en detalle lleva a decir que la zona de pertenencia uno en el entorno de la zona central de las clases es más pequeña en el segundo caso, lo que encaja con lo que se pretende. Los valores de inicialización del algoritmo no parecen ser muy críticos en los resultados.

Los resultados son similares, aunque con funciones de pertenencia más suaves (como se pretendía), a los de AFCR (ver figura 7.7 en 85). Aunque en el punto 8.6 se verá un estudio comparativo de AFCRC con los algoritmos revisados en el capítulo 7,

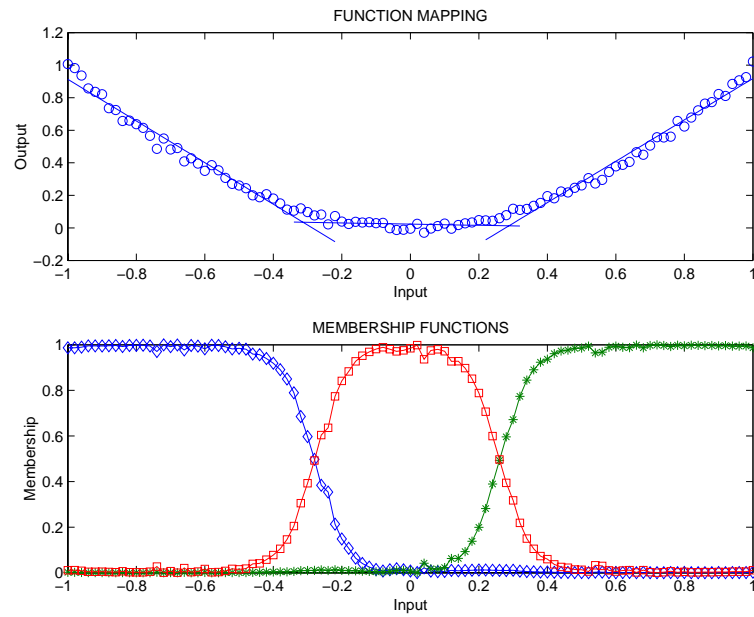


Figura 8.1: Clases con AFCRC ($\sigma_1^2 = 1$, $\gamma_1 = 2 \times 10^{-1}$).

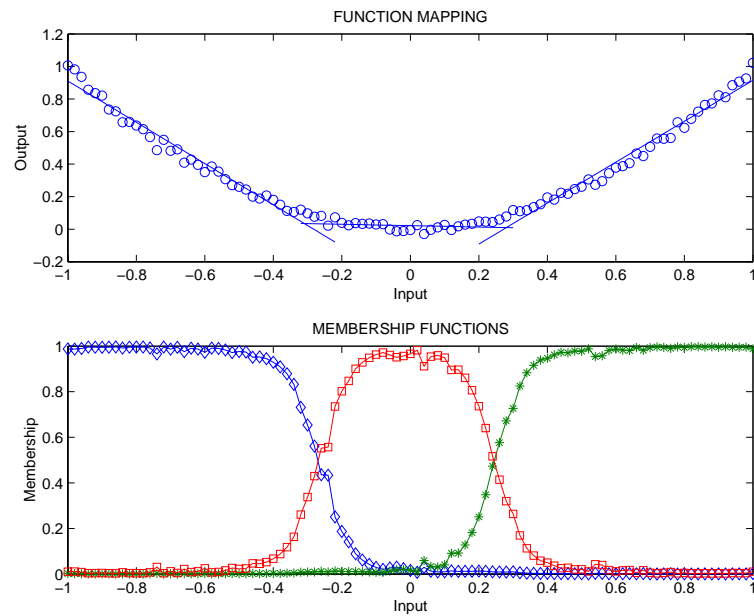


Figura 8.2: Clases con AFCRC ($\sigma_1^2 = 10^{-5}$, $\gamma_1 = 1,5 \times 10^{-2}$).

para observar con mayor claridad el efecto de J_{lejos} se puede aumentar el valor de γ_1 , permitiendo así que el nuevo término sea responsable de alrededor del 25 % del error total. Esto se muestra, por ejemplo, en la figura 8.3 para un valor de $\sigma_1^2 = 10^{-5}$. El

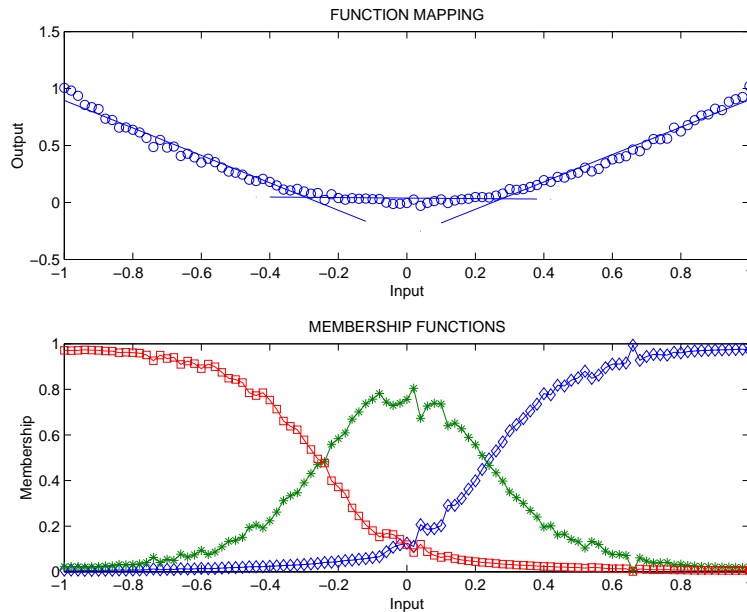


Figura 8.3: Clases con AFCRC ($\sigma_1^2 = 10^{-5}$, $\gamma_1 = 3 \times 10^{-1}$).

efecto del término definido en (8.1) en el proceso de agrupamiento es ahora muy claro, haciendo muy importante la baja pertenencia de los puntos lejanos al prototipo y degenerando así la forma de las clases.

El efecto de J_{cerca} se ha mostrado suficiente con un peso de 1 % del total del error J_{AFCRC} , haciendo en ocasiones que el algoritmo lleve a situaciones indeseadas para valores muy superiores (por encima del 10 %). En las figuras 8.4 y 8.5 se puede observar el efecto de este segundo término:

- no es apreciable si σ_2 es muy elevado,
- uniformiza el valor de la zona de pertenencia más alta de una clase.

Los términos σ_2 "elevado" (o "pequeño") y "apropiado" se harán comprensibles en el apartado 8.5 donde se comenta una propuesta sobre determinación de parámetros. La combinación de ambos términos determinando los parámetros de la forma en que se define en 8.5 lleva a resultados como el de la figura 8.7, donde se aprecia que el algoritmo AFCRC ejecutado con valores apropiados de los parámetros lleva a resultados que mejoran a los algoritmos existentes en el sentido de la interpretabilidad de las funciones de pertenencia, cumpliendo ambos términos su cometido y ayudando a conseguir unas clases más ajustadas a lo deseado. Como ya se ha comentado anteriormente, en el punto 8.6 se hace un estudio comparativo (cualitativo y cuantitativo) del nuevo algoritmo con los estudiados en el capítulo 7), tanto para este ejemplo como para otros.

Con la idea de minimizar en lo posible la tarea del ajuste de los parámetros, en el apartado siguiente se propone una estructura sencilla para su determinación.

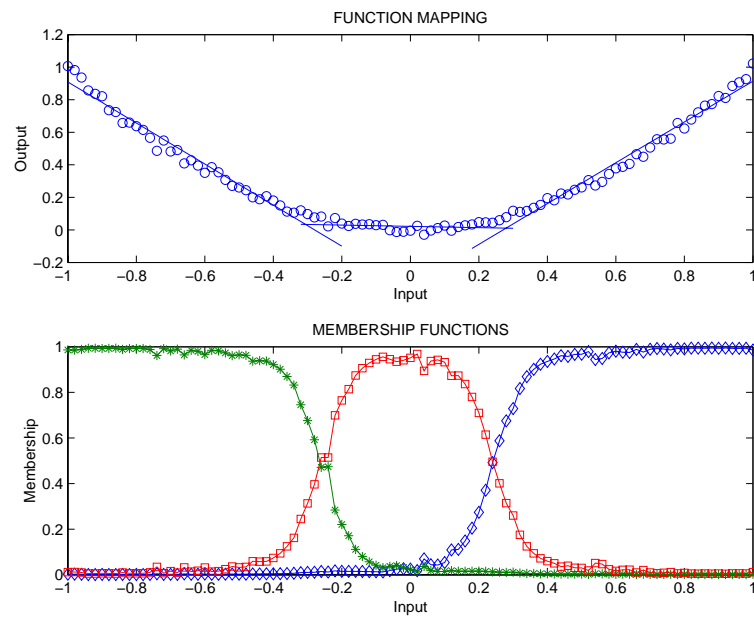


Figura 8.4: Clases con AFCRC ($\sigma_2^2 = 1$, $\gamma_2 = 2 \times 10^{-2}$).

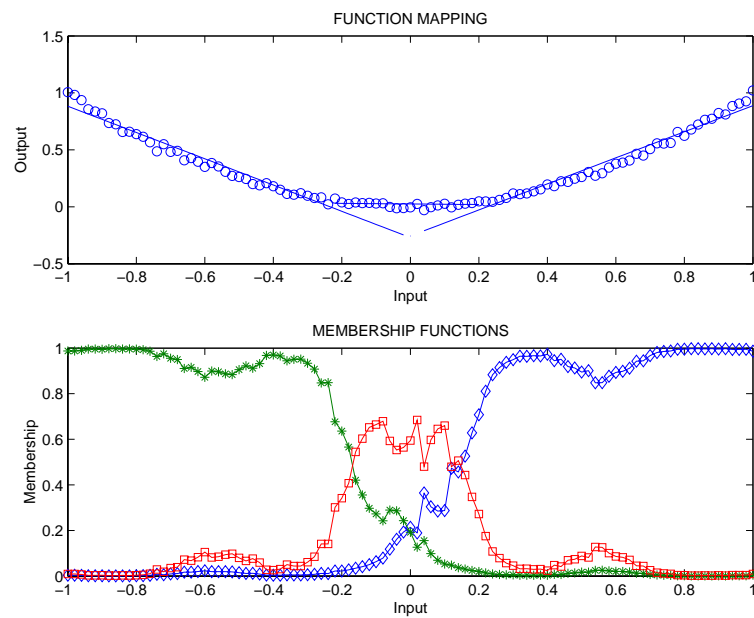


Figura 8.5: Clases con AFCRC ($\sigma_2^2 = 2 \times 10^{-2}$, $\gamma_2 = 2 \times 10^{-1}$).

8.5. Parámetros de ajuste del algoritmo

En este apartado se va a clarificar el significado de los diferentes parámetros del algoritmo AFCRC y se van a proponer reglas intuitivas que ayudarán al usuario en la determinación y ajuste de los nuevos parámetros definidos en la función de coste.

Como ya se ha venido comentando, los valores de σ_1 y σ_2 están relacionados con la distancia máxima (r_2) del entorno del prototipo para la que deben estar la mayoría de los puntos de alta pertenencia (σ_2) y la distancia máxima (r_1) a partir de la cual no deben haber altas pertenencia a la clase (σ_1).

Una posible interpretación de (8.1) y (8.2) se puede realizar relacionando estos términos con la distribución estadística exponencial de la forma $\exp(-\frac{L_{ik}^2}{2\sigma^2})$ en la que la mayoría de los puntos de una distribución (el 99.73 % de ellos) están en un radio 3σ de la media de la distribución, como se muestra en la figura 8.6. Otra forma de explicar el significado de los términos σ es como el radio a partir del cual se producirá un importante incremento de coste en la función objetivo a minimizar cuando los datos incluidos en una clase no se ajuste a lo deseado (formas suaves).

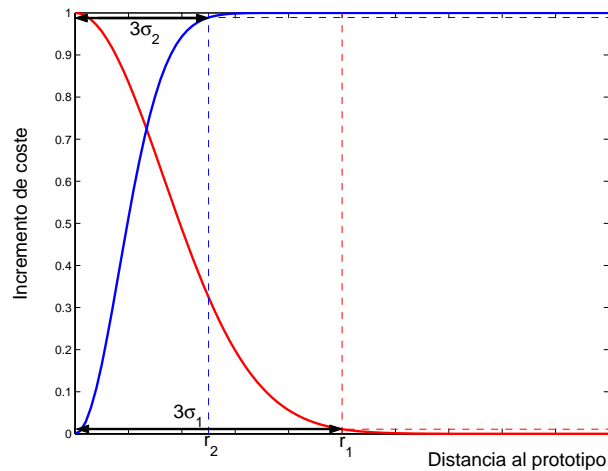


Figura 8.6: Aumento de la función de coste en función de la distancia de los puntos a los prototipos.

Una estimación de los radios r_1 de las clases se puede hacer empleando datos tan sencillos como las dimensiones del espacio de trabajo (donde se realiza la medida de las distancias L_{ik}) y el número de clases. Para ello se puede calcular la máxima distancia L_{ik} de dos puntos en dicho espacio de trabajo y dividir por el número de clases, lo que dará una estimación del máximo diámetro d_1 de una clase cuya mitad será un valor para $r_1 = 3\sigma_1$.

En cuanto a la elección de σ_2 , una elección que da buenos resultados consiste en una relación aproximada de 2 entre los radios mayor y menor de la clase ($d_2 = d_1/2$).

En lo que respecta a los valores de γ_1 y γ_2 , dado que como se dijo en 8.3 representan (en las últimas iteraciones) la importancia relativa de la interpretabilidad frente al error de modelado, su determinación también debe ser supervisada por el usuario. Un procedimiento posible es ejecutar el algoritmo AFCRC para determinar el mínimo error de

modelado posible y aumentar los valores de γ_1 y γ_2 hasta que se produzca determinado aumento admisible del error de modelado (por ejemplo, un 10%) a cambio de una mejora en la interpretabilidad. Para facilitar el ajuste, esto se puede hacer en dos pasos: buscando primero un valor para γ_1 y haciendo $\gamma_2 = 0$, para posteriormente determinar el valor de γ_2 (una buena inicialización de este último puede ser el valor obtenido previamente para γ_1). Cuanto mayor sea el ruido existente en los datos, mayores deberán ser estos parámetros para obtener una forma aceptable en las funciones pertenencia.

Una recomendación final podría ser emplear datos normalizados (ver apartado 6.3.3 para una normalización de datos con media nula y varianza unitaria), pues esto permite interpretar de la misma forma los valores de los parámetros a definir por el usuario en conjuntos de datos distintos.

Con todo lo comentado en este punto, previamente al empleo del algoritmo AFCRC para la identificación de modelos, se podrían seguir una serie de pasos para obtener unos parámetros adecuados en la ejecución del algoritmo que serían (tras normalizar los datos y decidir el número de clases a emplear):

- Obtener la máxima distancia d_{max} posible con los datos disponibles. En el caso de trabajo en R^2 , los datos normalizados variarán en un rango $[x_1, x_2]$ para la primera variable y $[x_3, x_4]$ para la segunda, con lo que $d_{max} = \sqrt{(x_2 - x_1)^2 + (x_4 - x_3)^2}$.
- Determinar el valor de σ_1 . Dado el número de clases c , la distancia de dos puntos de una misma clase será $d_1 = d_{max}/c = 6\sigma_1$, de donde σ_1 puede ser despejada.
- Calcular un valor para σ_2 . Procediendo como en el caso anterior, hacemos que la distancia a la que se encuentran la mayoría de los puntos de la clase es la mitad de d_1 y obtenemos σ_2 .
- Ajustar, mediante diferentes ejecuciones, los valores para γ_1 y γ_2 de forma que las componentes en el índice de coste de AFCRC debidas a J_{lejos} y J_{cerca} no superen el 10% del total.

En el caso de la parábola estudiada en el apartado anterior, la aplicación de estas ideas llevaría a que los parámetros adecuados para la ejecución del algoritmo AFCRC en ese caso son:

- Los datos normalizados varían, en el espacio de entrada, en el rango $[-1.7, 1.7]$, con lo que $d_{max} = \sqrt{(1.7 + 1.7)^2} = 3.4$.
- La distancia máxima de dos puntos de una misma clase (se consideran 3 clases) será $d_1 = 3.4/3 = 1.13 = 6\sigma_1$, con lo que $\sigma_1^2 = 0.036$.
- Suponiendo que hacemos que la distancia a la que se encuentran la mayoría de los puntos de la clase es la mitad de d_1 , obtenemos $\sigma_2^2 = 0.009$.
- El ajuste, mediante diferentes ejecuciones, de los valores de γ_1 y γ_2 de forma que las componentes en el índice de coste de AFCRC debidas a J_{lejos} y J_{cerca} no superen el 10% del total (J_{lejos} el 9% y J_{cerca} 1%), lleva a $\gamma_1 = 10^{-2}$ y $\gamma_2 = 10^{-2}$.

con los resultados mostrados en la gráfica 8.7.

En la sección 8.6, así como en el ejemplo final de la Tesis que se verá en el capítulo 10, se ha empleado esta estructura de determinación de parámetros con buenos resultados. Hay que resaltar que, como se vió en el apartado 8.4, el algoritmo no es excesivamente sensible a estos parámetros y permite variaciones (siempre que estén en un entorno apropiado de valores) llegando a resultados coherentes con variaciones en las formas de las funciones de pertenencia.

8.6. Ejemplos de aplicación

En primer lugar se estudian los mismos ejemplos empleados en la evaluación de las técnicas de agrupamiento existentes que se realizó en el punto 7.5 para, en caso de mejorar aquellos resultados, pasar a la identificación de funciones más complejas (análisis de datos dinámicos).

8.6.1. Identificación de funciones no-lineales unidimensionales

Ejemplo 8.1. Caso 2: Parábola. Como se puede observar por inspección de la figura 8.7, AFCRC mejora los resultados del resto de algoritmos (ejemplo 7.2) en cuanto a la suavidad de las funciones de pertenencia identificadas. Además, los modelos lineales locales identificados son correctos y los parámetros han sido calculados en el proceso de agrupamiento. En cuanto a los resultados de los índices definidos en la sección 7.4 y que

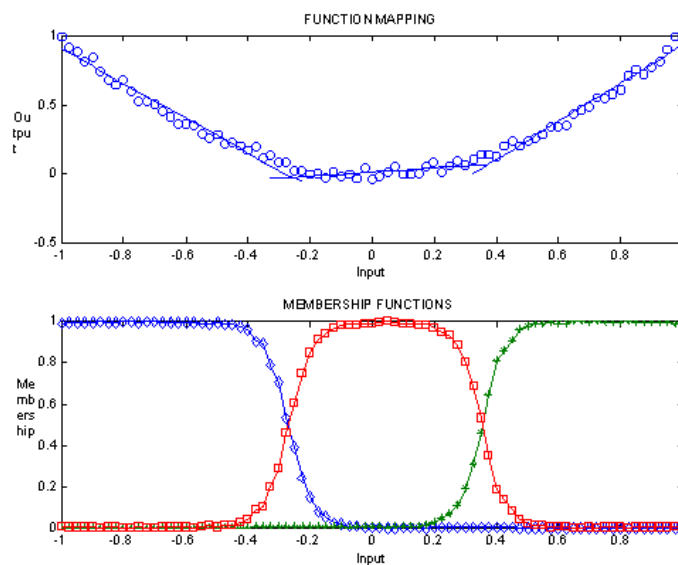


Figura 8.7: Detección de modelos locales con AFCRC (círculos: datos, líneas: modelos locales identificados).

se emplearon en el apartado 7.5, se han ampliado las tablas ya conocidas 7.2, 7.3 y 7.4 con los resultados del nuevo algoritmo. La primera (tabla 8.1) muestra los resultados

para índice de compacidad dado por la ecuación (7.16), la segunda (tabla 8.2) los del índice de solapamiento y la tercera (tabla 8.3) los índices de funcionamiento local y global. Como se puede observar, también el análisis cuantitativo da como resultado que el algoritmo AFCRC mejora al resto de algoritmos ensayados.

Método de agrupamiento	Índice de compacidad (IC)
FCM	9.5308
GK	3.9629
FCRM	11.8673
AFCR	3.8664
AFCRC	3.7046

Tabla 8.1: Resultados del índice de compacidad para el caso 2.

Método de agrupamiento	Índice de solapamiento (IS ₃ ; IS ₂ ; IS ₁)
FCM	(0.0325; 0.1215; 0.8460)
GK	(0.0027; 0.0660; 0.9313)
FCRM	(0.0166; 0.1131; 0.8703)
AFCR	(0.0023; 0.0592; 0.9385)
AFCRC	(0.0020; 0.0561; 0.9420)

Tabla 8.2: Resultados del índice de solapamiento para el caso 2.

Método de agrupamiento	Índices locales	IF
FCM	0.257 0.058 0.064	0.0127
GK	0.066 0.047 0.042	0.0022
FCRM	0.024 0.084 0.132	0.0032
AFCR	0.035 0.044 0.073	0.0019
AFCRC	0.063 0.040 0.051	0.0022

Tabla 8.3: Índices locales e IF para el caso 2.

Ejemplo 8.2. Caso 3: Ratio de cinética de crecimiento de un biorreactor.

Los resultados del nuevo algoritmo (figura 8.8) vuelven a mejorar en este caso los de los algoritmos existentes (ejemplo 7.2) en el sentido de la interpretabilidad de las funciones de pertenencia de las reglas identificadas. Los parámetros empleados se basan en el procedimiento del apartado 8.5 con lo que:

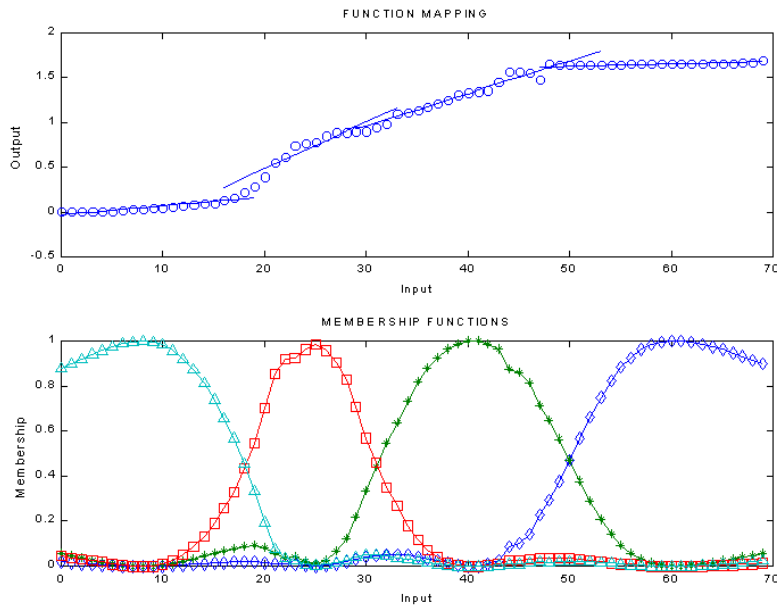


Figura 8.8: Detección de modelos locales con AFCRC (círculos: datos experimentales, líneas: modelos locales identificados).

- Los datos normalizados varían en el rango $[-1,7, 1,7]$, con lo que $d_{max} = \sqrt{(1,7 + 1,7)^2} = 3,4$.
- La distancia máxima de dos puntos de una misma clase será $d_1 = 3,4/4 = 6\sigma_1$, con lo que $\sigma_1^2 = 0,02$.
- Suponiendo que hacemos que la distancia a la que se encuentran la mayoría de los puntos de la clase es la mitad de d_1 , obtenemos $\sigma_2^2 = 0,005$.
- El ajuste, mediante diferentes ejecuciones, de los valores de γ_1 y γ_2 de forma que las componentes en el índice de coste de AFCRC debidas a J_{lejos} y J_{cerca} no superen el 10% del total (J_{lejos} el 9% y J_{cerca} 1%), lleva a $\gamma_1 = 10^{-3}$ y $\gamma_2 = 10^{-3}$.

Se han realizado diferentes identificaciones aumentando el número de clases a detectar, pudiéndose observar que el algoritmo mantiene la interpretabilidad de las funciones de pertenencia y disminuye los rangos de validez de los modelos locales lineales.

■

8.6.2. Identificación de funciones de varias variables

El análisis de la identificación de funciones con dos entradas y una salida es más complicado de visualizar y, más allá de este número de variables, es imposible. En cualquier caso, se van a analizar dos casos de funciones con dos entradas y una salida que se podrían corresponder (de hecho el segundo caso así es) con la identificación de funciones dinámicas no lineales (ver el tratamiento de funciones dinámicas no lineales en modelos borrosos del apartado 5.3).

Ejemplo 8.3. Caso 1: Función bidimensional estática no lineal. Los parámetros para AFCRC serán, según los datos presentados en el ejemplo 7.1:

- Los datos normalizados varían, para ambos ejes, en el rango $[-1,7, 1,7]$, con lo que $d_{max} = \sqrt{(1,7 + 1,7)^2 + (1,7 + 1,7)^2} = 4,80$.
- La distancia máxima de dos puntos de una misma clase será $d_1 = 4,80/4 = 6\sigma_1$, con lo que $\sigma_1^2 = 0,04$.
- Suponiendo que hacemos que la distancia a la que se encuentran la mayoría de los puntos de la clase es la mitad de d_1 , obtenemos $\sigma_2^2 = 0,01$.
- El ajuste, mediante diferentes ejecuciones, de los valores de γ_1 y γ_2 de forma que las componentes en el índice de coste de AFCRC debidas a J_{lejos} y J_{cerca} no superen el 10 % del total (J_{lejos} el 9 % y J_{cerca} 1 %), lleva a $\gamma_1 = 10^{-2}$ y $\gamma_2 = 10^{-3}$.

Las figuras 8.9 y 8.10 muestran, respectivamente, como los pares de datos de entrada son asignados a las clases (se ha considerado la mayor pertenencia de un punto a una clase para realizar la figura) por la aplicación de AFCR (los resultados de GK son similares y se vieron en el ejemplo 7.1) y AFCRC. Para favorecer la interpretación se ha dibujado el prototipo de la clase (+) y los límites de las clases, excepto en el caso de la clase no interpretable de la figura 8.9. Dado que las funciones de pertenencia identificadas

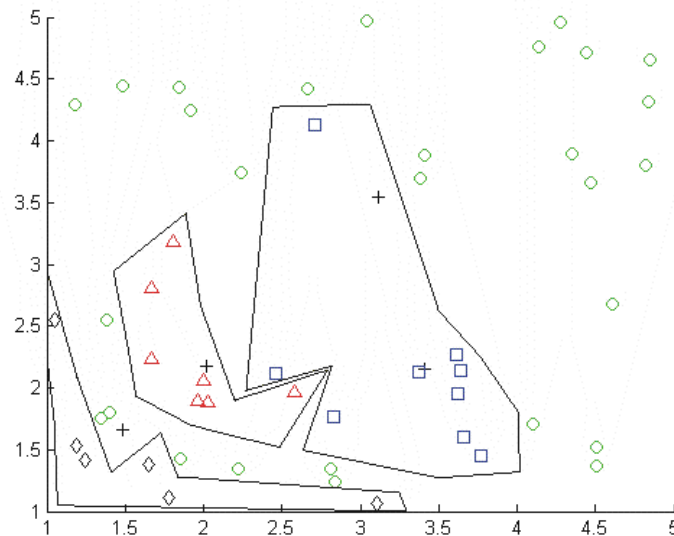


Figura 8.9: Asignación de datos a clases realizada por AFCR.

son más compactas y corresponden de forma más precisa a los modelos locales que daría un experto al inspeccionar la figura 7.1, AFCRC mejora de nuevo los resultados de AFCR en el sentido de la interpretabilidad de las funciones de pertenencia. Como se ha comentado aunque la función identificada es estática, dado que los sistemas dinámicos son tratados para la identificación borrosa como funciones multidimensionales estáticas (Sugeno and Yasukawa, 1993), este ejemplo muestra la adecuación de AFCRC en la

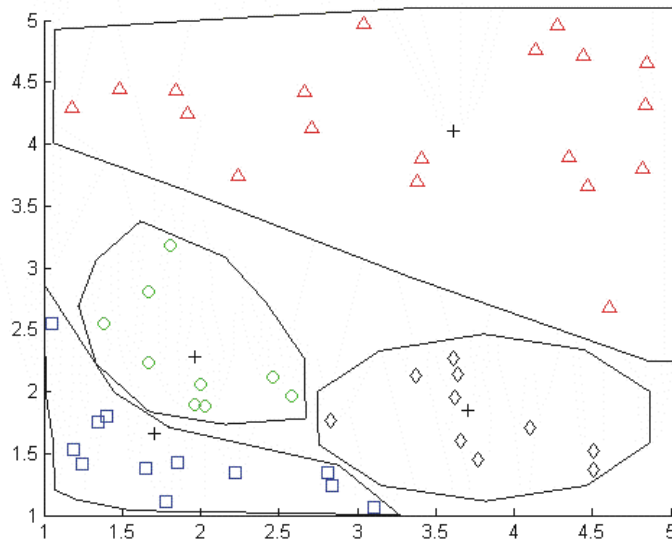


Figura 8.10: Asignación de datos a clases realizada por AFCRC.

identificación de sistemas dinámicos. El algoritmo AFCRC se puede usar con conjuntos de datos con más dimensiones pero, en esos casos, los resultados ya no pueden ser visualizados.

En este caso bidimensional los índices de validez definidos en 7.4 son también buenos indicadores de la adecuación de los modelos locales identificados. La tabla 8.4 muestra los resultados del índice de compacidad, la tabla 8.5 los del índice de solapamiento y la tabla 8.6 los índices de funcionamiento local y global.

Método de agrupamiento	Índice de compacidad (IC)
FCM	81.0828
GK	94.9793
FCRM	140.6643
AFCR	104.8621
AFCRC	57.7180

Tabla 8.4: Resultados para el índice de compacidad del caso 2.

Método de agrupamiento	Índice de solapamiento (IS) (IS ₄ ; IS ₃ ; IS ₂ ; IS ₁)
FCM	(0.0388; 0.0761; 0.1394; 0.7457)
GK	(0.0056; 0.0317; 0.0907; 0.8720)
FCRM	(0.0053; 0.0257; 0.0724; 0.8965)
AFCR	(0.0202; 0.0553; 0.1211; 0.8034)
AFCRC	(0.0116; 0.0310; 0.0940; 0.8634)

Tabla 8.5: Resultados para el índice de solapamiento del caso 2.

Método de agrupamiento	Índices locales				IF
FCM	0.4969	5.3139	1.8984	2.7130	0.1613
GK	0.9901	1.0077	0.9220	1.0222	0.0654
FCRM	0.0247	0.3300	0.1797	0.0546	0.0088
AFCR	0.1609	0.0815	0.1126	0.0930	0.0072
AFCRC	0.2385	0.9487	1.1030	0.7984	0.0365

Tabla 8.6: Índices locales e IF del caso 2.

Los índices de compacidad indican que los métodos AFCRC y GK dan clases más compactas que FCM, FCRM y AFCR, mientras que los índices de solapamiento son bastante adecuados en todos los algoritmos. Se concluye, por tanto, que AFCRC es capaz de mejorar los resultados de los algoritmos que se habían estudiado previamente. ■

Ejemplo 8.4. Caso 4: Función dinámica no-lineal. Finalmente, se van a comprobar las capacidades del algoritmo AFCRC desde el punto de vista de la identificación de los parámetros de unos modelos locales ocultos tras una estructura no lineal. Para ello se ha realizado un simulador de un sistema dinámico no lineal basado en la dinámica del movimiento longitudinal del vehículo presentado en (Hunt and Johansen, 1997). Este modelo simplificado de un vehículo consiste en una única señal de control u que combina aceleración y frenado (variable única normalizada en el rango 0-1, ya que se considera que si se frena no se acelera y viceversa), siendo la única salida la velocidad longitudinal y .

El modelo consta de tres regímenes de operación, cada uno de ellos correspondiente a cierto rango en el entorno del punto de operación (u^* , y^*). Los modelos locales definidos se muestran en la tabla 8.7. La relativamente pequeña variación de parámetros entre

Régimen	Pto. operación	Func. transferencia	Offset
1	(0.1,3.5)	$\frac{0,3z^{-1}}{1-0,98z^{-1}}$	0.04
2	(0.5,15)	$\frac{1,8z^{-1}}{1-0,94z^{-1}}$	0.0
3	(0.9,26.5)	$\frac{0,6z^{-1}}{1-0,96z^{-1}}$	0.52

Tabla 8.7: Modelos locales.

los tres modelos a identificar puede hacer que el proceso de identificación del modo de operación sea algo más complicado que en el caso de modelos muy distintos. Por otra parte, este tipo de sistemas pueden tener la ventaja de hacer más sencillo un hipotético cambio de controladores si se desea realizar un posterior control del sistema. La superficie a identificar corresponde con el sistema dinámico discreto no-lineal (con funciones de validez trapezoidales f_1 , f_2 y f_3 de núcleo 0.2 y normalizadas) de la ecuación (8.15):

$$\begin{aligned}
 y_{k+1} = & f_1(0,98y_k + 0,3u_k + 0,04) + \\
 & + f_2(0,94y_k + 1,8u_k + 0,0) + \\
 & + f_3(0,96y_k + 0,6u_k + 0,52)
 \end{aligned} \tag{8.15}$$

El algoritmo AFCRC lleva, siguiendo la evolución de la función objetivo mostrada en

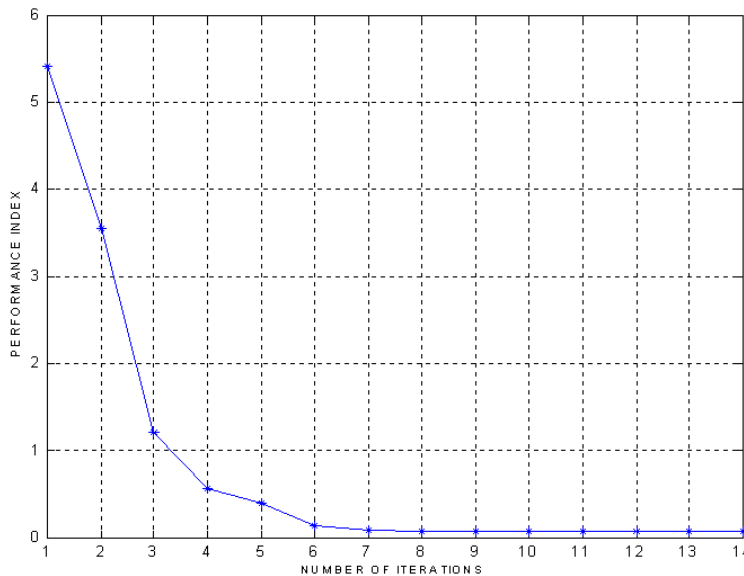


Figura 8.11: Evolución de la función objetivo del algoritmo AFCRC.

8.11, a los modelos locales identificados que se presentan en la tabla 8.8, empleando los parámetros:

- Los datos normalizados varían en el rango $[-1,7, 1,7]$, con lo que $d_{max} = \sqrt{(1,7 + 1,7)^2} = 3,4$.
- La distancia máxima de dos puntos de una misma clase será $d_1 = 3,4/4 = 6\sigma_1$, con lo que $\sigma_1^2 = 0,02$.
- Suponiendo que hacemos que la distancia a la que se encuentran la mayoría de los puntos de la clase es la mitad de d_1 , obtenemos $\sigma_2^2 = 0,005$.
- El ajuste, mediante diferentes ejecuciones, de los valores de γ_1 y γ_2 de forma que las componentes en el índice de coste de AFCRC debidas a J_{lejos} y J_{cerca} no superen el 10% del total (J_{lejos} el 9% y J_{cerca} 1%), lleva a $\gamma_1 = 10^{-3}$ y $\gamma_2 = 10^{-3}$.

Régimen	Pto. operación	Func. transferencia	Offset
1	(0.1004,3.1084)	$\frac{0,2583z^{-1}}{1-0,9852z^{-1}}$	0.0291
2	(0.5035,14.7769)	$\frac{1,7313z^{-1}}{1-0,9420z^{-1}}$	0.0044
3	(0.9030,26.2319)	$\frac{0,5473z^{-1}}{1-0,9647z^{-1}}$	0.4436

Tabla 8.8: Modelos identificados por AFCRC.

Los resultados de AFCRC son de nuevo los mejores obtenidos y se presentan para la comparación en la tabla 8.9, de la que se concluye la elección de este algoritmo en este ejemplo. Con el algoritmo GK, además, ha requerido para su convergencia numerosas ejecuciones con diferentes inicializaciones de parámetros.

Régimen	Pto. operación	Func. transferencia	Offset
1	(0.0998,3.0824)	$\frac{0,4402z^{-1}}{1-0,9739z^{-1}}$	0.0433
2	(0.4993,14.6181)	$\frac{2,4549z^{-1}}{1-0,9174z^{-1}}$	0.0038
3	(0.8993,26.0735)	$\frac{0,4781z^{-1}}{1-0,9623z^{-1}}$	0.5723

Tabla 8.9: Modelos identificados por GK.

8.7. Conclusiones

Tomando las características deseables en un algoritmo de agrupamiento específico para la identificación de modelos borrosos para control, se ha definido una nueva función objetivo que las reúne. La minimización de la función de coste ha dado lugar a un nuevo algoritmo de agrupamiento, adecuado para el modelado de sistemas no lineales con modelos borrosos TS para control.

El algoritmo desarrollado se ha aplicado a numerosos ejemplos con resultados muy interesantes, poniendo de manifiesto su aplicabilidad. El único problema del algoritmo está en la determinación y ajuste de los nuevos parámetros que se han definido en su función de coste, para lo cual se han propuesto reglas intuitivas que ayudarán al usuario en su determinación.

Por lo tanto, las aportaciones de este capítulo se pueden resumir en:

- Propuesta de un índice de coste que reúne las características deseables en la identificación para control determinadas en el capítulo 7,
- Minimización de la función de coste en la forma de un algoritmo de agrupamiento (AFCRC), manteniendo así las ventajas de esta forma de trabajo,
- Generación de una nueva familia de algoritmos de agrupamiento desde la función de coste, mostrando como ejemplo la versión posibilista del mismo (PAFCRC),
- Determinación de una estructura de trabajo para la determinación de los valores de los parámetros que aparecen en AFCRC a definir por el usuario,
- Aplicación del algoritmo en diferentes ejemplos, en comparación con algoritmos existentes, dando resultados apropiados.

Definido así un algoritmo de agrupamiento que da como resultados modelos locales (como los definidos en el capítulo 3) de sistemas no lineales desde sus datos experimentales, se estudiará su aplicabilidad a control (según los métodos revisados en el capítulo 4).

Capítulo 9

Integración de técnicas de agrupamiento y control

*Son vanas y están plagadas de errores
las ciencias que no han nacido del experimento.*

Leonardo Da Vinci.

En este capítulo se evaluará el comportamiento de las técnicas de control aplicables a los modelos identificados con el algoritmo descrito en el capítulo 8. Este algoritmo proporcionaba modelos locales que procuran ser linealizaciones del proceso en el entorno de puntos de operación (prototipos de las clases). Por tanto, las técnicas de control aplicables serán las revisadas en el capítulo 4.

9.1. Introducción

Una vez se ha desarrollado un modelo borroso lo suficientemente preciso de un proceso no lineal, éste se puede usar para diseñar un controlador para el proceso. En (Babuska, 1996) se comenta la posibilidad de diseñar un controlador no lineal por inversión del modelo borroso del sistema (siempre que la dinámica del proceso invertido sea estable), o usar las capacidades de predicción del modelo borroso para diseñar un control predictivo basado en el modelo.

Si los modelos lineales identificados desde los datos experimentales son realmente linealizaciones del proceso (como busca el algoritmo de agrupamiento AFCRC), se pueden emplear las mismas técnicas de control que se basan en representar sistemas no lineales por un modelo lineal que lo aproxime y diseñar un controlador para él, como las técnicas de control por planificación de ganancia que parecen directamente aplicables a los modelos identificados con el algoritmo de agrupamiento comentado en el capítulo anterior.

Otra opción que se evaluará es la que emplea la resolución de LMIs (Linear Matrix Inequalities) para asegurar la estabilidad del sistema. Con este enfoque, el problema se convierte en una resolución matemática de ecuaciones y no es necesaria la interpretabilidad de los modelos locales como linealizaciones del proceso.

9.2. Control por planificación de ganancia

Como ya se comentó en el capítulo 4, el diseño de un controlador por planificación de ganancia para una planta no lineal se describe como un procedimiento de cuatro pasos:

- calcular un modelo lineal de parámetros variables de la planta,
- diseñar controladores mediante métodos lineales,
- realizar la planificación de ganancia propiamente dicha, y
- evaluar el rendimiento del control diseñado.

La propuesta de esta Tesis es emplear el algoritmo AFCRC para obtener el modelo de la planta, siguiendo el método descrito en el capítulo 4 para el resto de pasos.

9.2.1. Modelos LPV identificados con AFCRC

Este apartado describe la posible correspondencia de los distintos modelos que emplea la planificación de ganancia con aquellos modelos que, tras un proceso de identificación por agrupamiento con el algoritmo AFCRC, podrían estar disponibles.

El agrupamiento, en principio, puede dar lugar a modelos locales que correspondan a cualquier tipo de linealización en el entorno de puntos de operación. Es evidente que, si se desean seguir los pasos de diseño de un controlador por planificación de ganancia desde el modelo identificado en el agrupamiento, se hará necesario un paso previo de estudio del modelo que se ha obtenido para clarificar qué opción de las existentes (desde el punto de vista del conjunto de puntos de operación empleados) se adecúa, con la idea de emplear las técnicas de control adecuadas y que lleven a un controlador global apropiado.

Dos son las posibilidades que nos podemos encontrar para los modelos identificados, ya que pueden corresponder a modelos basados en puntos de equilibrio o de operación. Las linealizaciones en torno a puntos de equilibrio se darán siempre que se tenga la fortuna de que todos los prototipos identificados se correspondan con puntos de equilibrio del sistema. Sin embargo, dado que el algoritmo empleado para la identificación no tiene ninguna restricción respecto a las características del prototipo, lo habitual será obtener linealizaciones en puntos fuera del equilibrio.

Respecto a la posibilidad de identificar linealizaciones basadas en la velocidad, si bien el agrupamiento no proporciona directamente modelos de este tipo (como los descritos por (3.12) ó (3.14)), estas representaciones pueden ser obtenidas fácilmente desde las linealizaciones en puntos de operación. De este modo, las aproximaciones al control que parten de modelos basados en velocidad del sistema no lineal serán totalmente aplicables.

En definitiva, por ser los dos casos más generales y no necesitar que los prototipos sean puntos de equilibrio, los métodos que se puede decir que serán directamente aplicables a todos los modelos identificados con las técnicas propuestas a lo largo de esta Tesis serán los de linealización en torno a puntos de operación y linealización basada en la velocidad.

9.2.2. Aplicabilidad de controladores basados en modelos LPV del sistema en puntos de operación

En este apartado, mediante un ejemplo, se ilustrará el método de diseño propuesto para diseño de controladores basados en modelos LPV del sistema en puntos de operación, resaltando los posibles problemas que aparecen y proponiendo soluciones para mejorar el comportamiento.

Ejemplo 9.1. Sea el sistema no lineal descrito por la ecuación (9.1)

$$y_{k+1} = 0,0875(y_k + 2,0714)^2 + 0,3u_k + 2,6714 \quad (9.1)$$

en el que y es la salida del sistema y u la entrada. De (9.1) se desprende que la no linealidad del sistema está en la salida del mismo, siendo constante el término correspondiente a la entrada. La superficie asociada al sistema se puede observar en la figura 9.1.

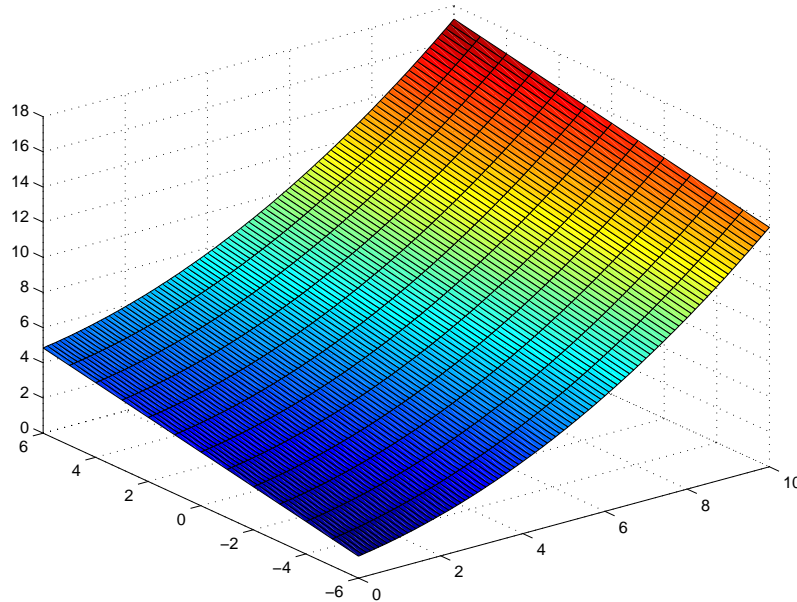


Figura 9.1: Superficie correspondiente al sistema (9.1).

Identificación de modelos locales. Supongamos que, de forma automática tras la aplicación del algoritmo AFCRC o simplemente tras la elección de dos puntos de operación (u_{k-1}, y_{k-1}, y_k) en el espacio entrada-salida como en los métodos tradicionales de planificación de ganancia descritos, se ha llegado a los modelos locales (linealizaciones) de la tabla 9.1.

Estos modelos se corresponden con los planos de la figura 9.2, los cuales serán tangentes a la superficie de la figura 9.1 en los puntos de operación que aparecen en la tabla 9.1.

Dado que el sistema no presenta no linealidad alguna en la entrada, como se puede observar en (9.1), se pueden analizar los modelos locales del sistema, de forma simplificada, con una proyección en dos dimensiones que puede ayudar a su interpretación y a

Punto de operación	Función de transferencia	Offset
$P_1=(0,2.5,4.5)$	$\frac{0,3z^{-1}}{1-0,8z^{-1}}$	2.5
$P_2=(0,5.3572,7.5)$	$\frac{0,3z^{-1}}{1-1,3z^{-1}}$	0.5356

Tabla 9.1: Modelos locales del sistema (9.1).

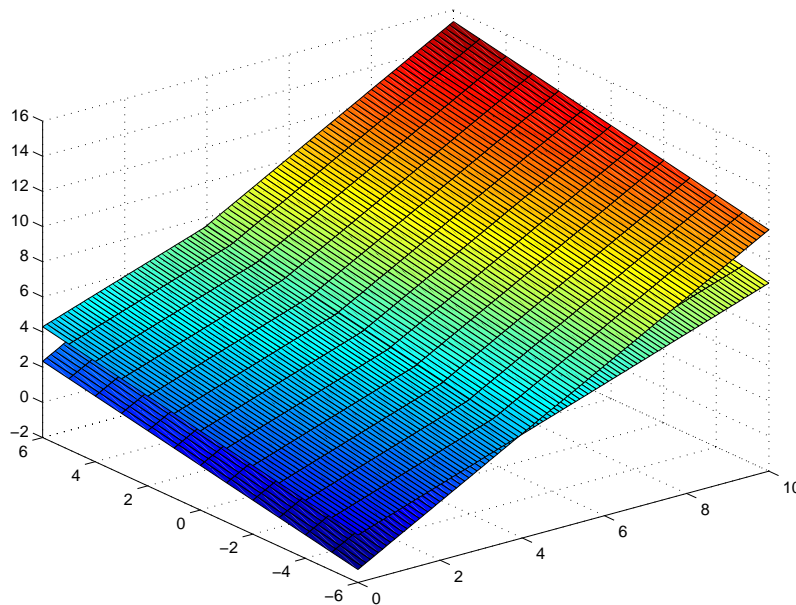


Figura 9.2: Planos correspondientes a los modelos locales de la tabla 9.1.

la búsqueda de la función de interpolación correcta. La figura 9.4 representa el sistema en el caso $u = 0$. En esta figura es fácil ver que la no linealidad del sistema es de tipo parabólico y que en los puntos de operación elegidos los modelos locales son tangentes al sistema y lo modelan perfectamente.

Integración de modelos locales. Con los modelos locales disponibles, se puede pensar que empleando las funciones de pertenencia identificadas de la figura 9.3 (interpolación lineal), se obtendrá una aproximación buena al modelo (9.1) entre los puntos de operación y que esta forma de interpolación puede ser válida incluso para el desarrollo de un controlador por planificación de ganancia. Sin embargo, ya se vió en 5.2 que esto no es válido para el modelado y puede corroborar en este caso en la figura 9.5.

A continuación se demostrará que, aunque una de las propuestas de la planificación de ganancia es utilizar dichas funciones de pertenencia para la ponderación de las acciones de control necesarias entre puntos de funcionamiento para los que se han desarrollado controladores, esta interpolación tampoco es así para control. Con la interpolación dada en (5.4) (la alternativa (7.13) es complicado aplicarla en un caso bidimensional como el que nos ocupa y habría que optimizar el parámetro γ), las ganancias del controlador global pueden ser localmente negativas incluso cuando las ganancias de los controladores locales sean positivas.

Diseño de controladores locales. Para el diseño de los controladores se puede

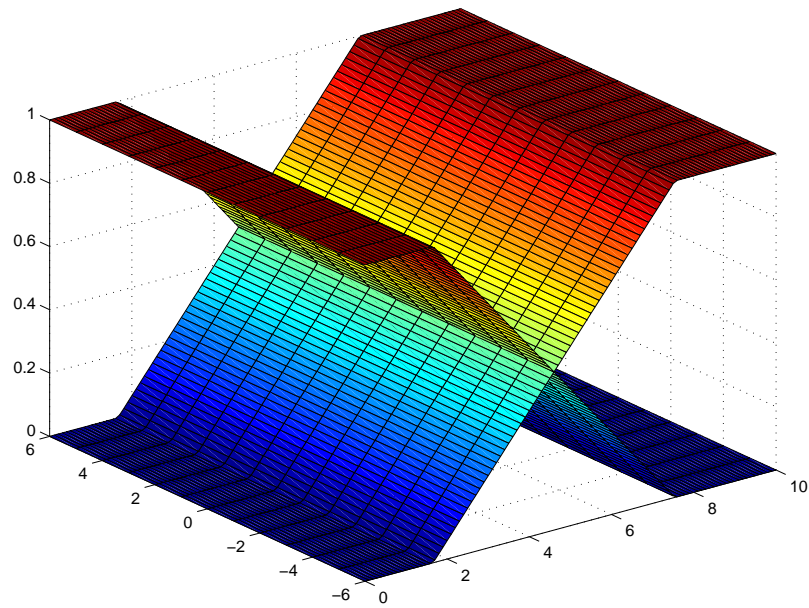


Figura 9.3: Funciones de pertenencia para la interpolación de los modelos locales.

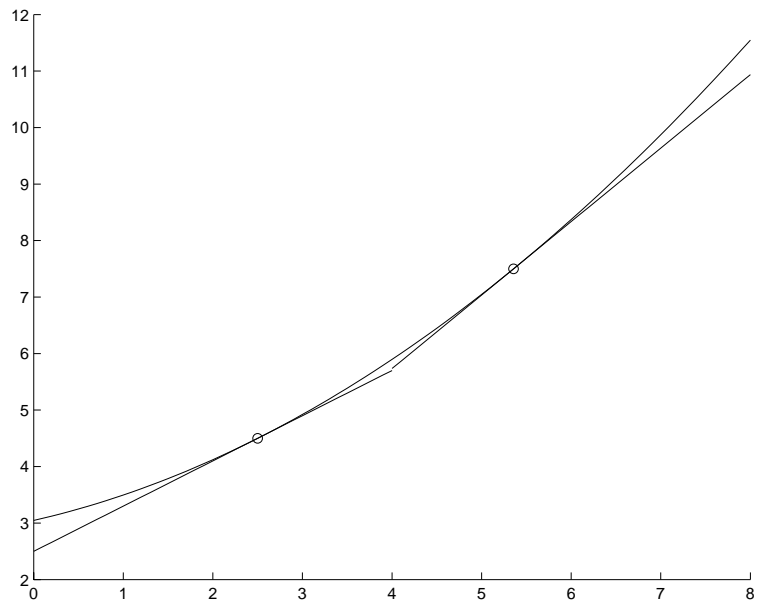


Figura 9.4: Proyección sobre el plano (y_k, y_{k-1}) del sistema y sus linealizaciones.

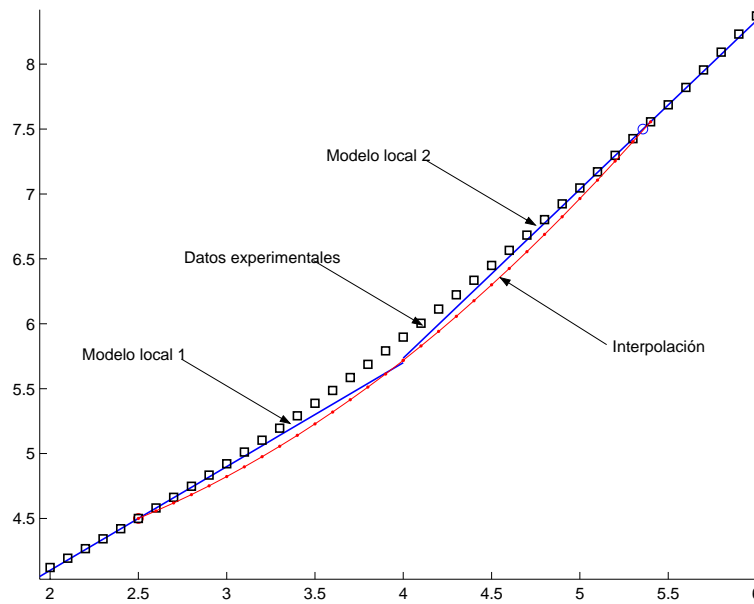


Figura 9.5: Diferencia entre los datos experimentales (cuadrados) y la interpolación de los modelos lineales identificados (línea con puntos) con las funciones de pertenencia de la figura 9.3.

usar un criterio de diseño tradicional como, por ejemplo, el de asignación de polos. Para asignar un polo en 0.4, la acción de control necesaria en cada punto de operación se puede obtener resolviendo el sistema de ecuaciones formado por el correspondiente modelo local de la tabla 9.1 y (9.2), donde r es la referencia a seguir por el sistema.

$$\hat{y}_{k+1} - r_k = 0,4(\hat{y}_k - r_k) \quad (9.2)$$

Para el modelo local en el punto de operación (0,2.5,4.5) la ecuación a resolver es:

$$0,8y_k + 0,3u_k + 2,5 - r_k = 0,4(\hat{y}_k - r_k) \quad (9.3)$$

lo que resulta, despejando u_k en (9.3), en el controlador dado por la acción de control (9.4):

$$u_k^{(1)} = \frac{(0,4 - 0,8)}{0,3}y_k + \frac{(1 - 0,4)}{0,3}r_k - \frac{2,5}{0,3} = -1,33y_k + 2r_k - 8,33 \quad (9.4)$$

En el caso del modelo local en el punto de operación (0,5.3572,7.5) la ecuación a resolver es:

$$1,3y_k + 0,3u_k + 0,5356 - r_k = 0,4(\hat{y}_k - r_k) \quad (9.5)$$

lo que resulta, despejando u_k en (9.5), en el controlador dado por la acción de control (9.6):

$$u_k^{(2)} = \frac{(0,4 - 1,3)}{0,3}y_k + \frac{(1 - 0,4)}{0,3}r_k - \frac{0,5356}{0,3} = -3y_k + 2r_k - 1,79 \quad (9.6)$$

Integración de controladores locales. La teoría de planificación de ganancia lleva a un controlador global del sistema como combinación de los anteriores:

$$u_k^{(LPO)} = \mu_1 u_k^{(1)} + \mu_2 u_k^{(2)} \quad (9.7)$$

donde μ_1 y μ_2 no se definen y podrían ser, por ejemplo, las funciones de pertenencia identificadas (figura 9.3).

Aunque en un caso real es imposible, a modo de comprobación, pueden calcularse en este ejemplo las funciones de pertenencia que idealmente deberían haberse identificado. Estas funciones de pertenencia deberían obtener la acción de control adecuada al sistema por combinación de las acciones de control obtenidas para los modelos locales. Su valor se puede calcular analíticamente resolviendo el sistema de ecuaciones formado por (9.7) (con la restricción $\mu_1 + \mu_2 = 1$) y el controlador que, con las mismas especificaciones que los modelos locales, se habría calculado analíticamente para el sistema no lineal. Dicho controlador se obtiene aplicando las especificaciones (9.2) a (9.1) llegando a:

$$0,0875(y_k + 2,0714)^2 + 0,3u_k + 2,6714 - r_k = 0,4(y_k - r_k) \quad (9.8)$$

de donde al despejar se obtiene una acción de control:

$$u_k^{(OPT)} = -0,29y_k^2 + 0,13y_k + 2r_k - 10,1713 \quad (9.9)$$

Como se ha comentado, desde el sistema de ecuaciones formado por (9.7) y (9.9) (teniendo en cuenta condición de partición borrosa $\mu_1 + \mu_2 = 1$) podremos despejar las pertenencias adecuadas para una correcta interpolación:

$$\mu_1^{(OPT)}(y_k) = \frac{-0,29y_k^2 + 0,13y_k - 8,3813}{1,67y_k - 6,54}; \quad \mu_2^{(OPT)}(y_k) = 1 - \mu_1^{(OPT)}(y_k) \quad (9.10)$$

Evidentemente esto no corresponde a una interpolación lineal de controladores, como se puede observar en la representación de $\mu_1^{(OPT)}$ en la figura 9.6, ni a ninguna función de pertenencia borrosa identificada experimentalmente en las que la restricción impuesta por partición borrosa hace que las funciones de pertenencia de modelos borrosos tengan sus valores en el rango $[0,1]$.

Para evitar en lo posible los diferentes efectos de las malas interpolaciones de la planificación de ganancia, en (Johansen and Murray-Smith, 1997) se propone incluir una acción integral. Esto se puede hacer, por ejemplo, diseñando un controlador clásico que incluya acción integral (lo que puede dar problemas por la suma ponderada de las acciones integrales de todos los controladores) o poniendo la acción integral en paralelo con el controlador global (con constante de integración K_i a definir) que lleve a una acción conjunta $u_k^{(INT)}$. El integrador eliminará todo tipo de "perturbaciones", considerando una de ellas el término de offset de los modelos (y por tanto el mal diseño del control por planificación de ganancia al obviar parte del modelo).

Evaluación del rendimiento. En la tabla 9.2 se comentan de forma resumida los resultados de diferentes simulaciones de los controladores por planificación de ganancia propuestos (en comparación con los locales en los puntos de operación y el óptimo) y que se pueden observar en las figuras 9.7 a 9.11.

Como conclusión general, el estudio de los controladores que la teoría de control por planificación de ganancia propone (con integrador $u_k^{(INT)}$ o sin él $u_k^{(LPO)}$), se puede

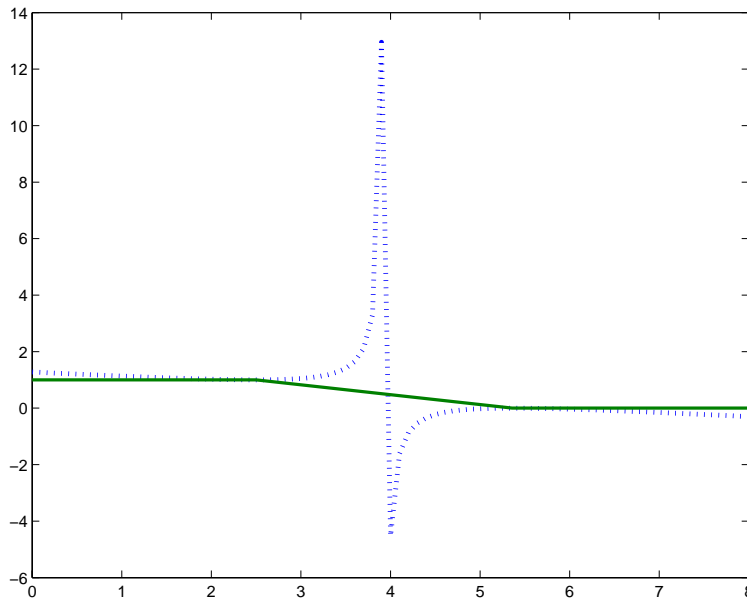


Figura 9.6: Funciones de pertenencia para la interpolación lineal $\mu_1^{(LIN)}$ (línea continua) e interpolación óptima $\mu_1^{(OPT)}$ (línea de puntos).

EXPERIMENTO	ACCIÓN DE CONTROL				
	$u_k^{(1)}$ (azul oscuro)	$u_k^{(2)}$ (azul claro)	$u_k^{(LPO)}$ (magenta)	$u_k^{(INT)}$ (mag. disc.)	$u_k^{(OPT)}$ (negro)
A. Entorno de P_1 (figura 9.7).	Pequeño error de posición	Inestable	Gran error de posición	Elevada sobrees oscilación	Correcto
B. Entorno de P_2 (figura 9.8).	Inestable	Pequeño error de posición	Gran error de posición	Elevado tiempo de establecimiento	Correcto
C. Entorno zona intermedia (figura 9.9).	Gran error de posición	Gran error de posición	Gran error de posición	Gran error de posición	Correcto
D. Salto de P_1 a P_2 (figura 9.10).	Malo	Malo	Correcto	-	Correcto
E. Salto de P_2 a P_1 (figura 9.11).	Malo	Malo	Correcto	-	Correcto

Tabla 9.2: Comportamiento, en experimentos a lo largo del rango de operación, de los diferentes controladores diseñados.

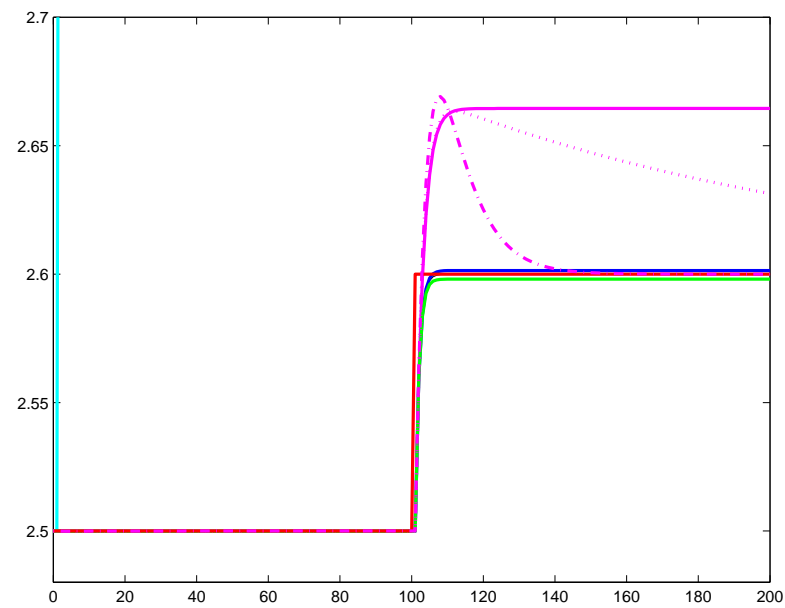


Figura 9.7: Experimento A. Control por planificación de ganancia en el entorno de P_1 : punto inicial 2.5, punto final 2.6.

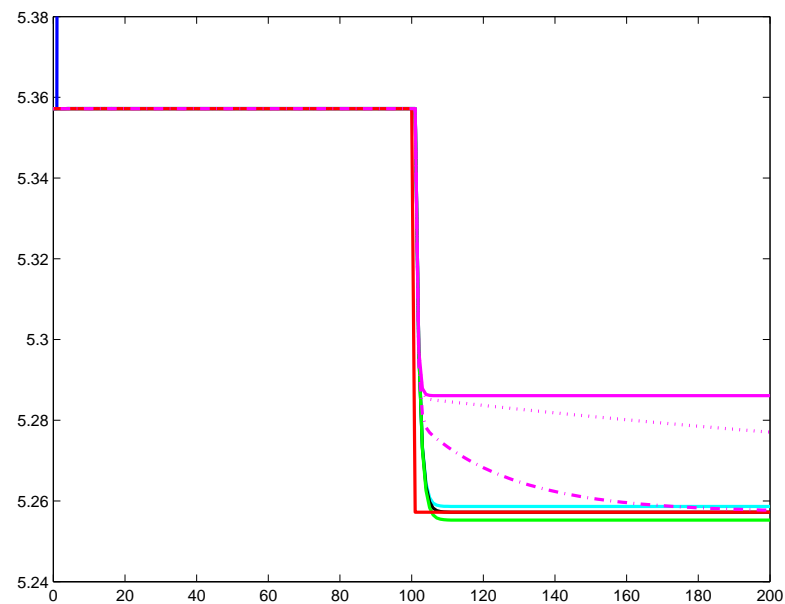


Figura 9.8: Experimento B. Control por planificación de ganancia en el entorno de P_2 : punto inicial 5.3572, punto final 5.2572.

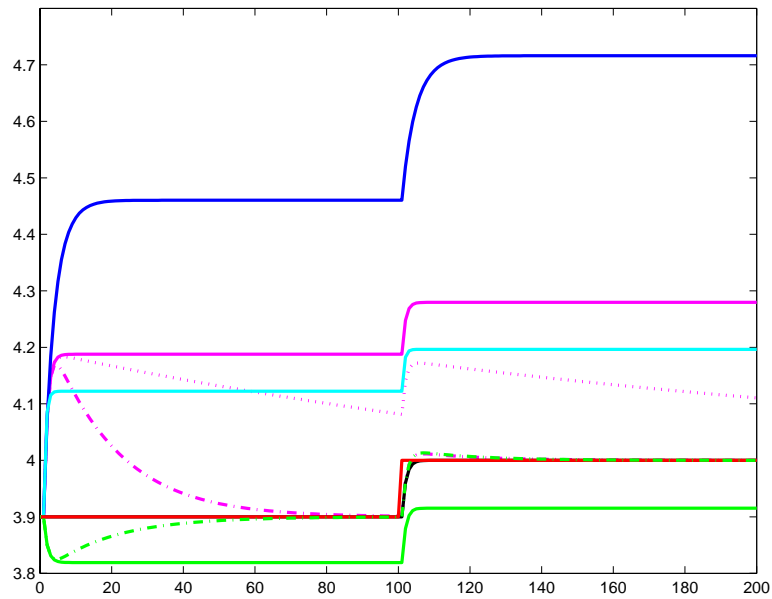


Figura 9.9: Experimento C. Control por planificación de ganancia en el entorno de un punto intermedio: punto inicial 3.9, punto final 4.

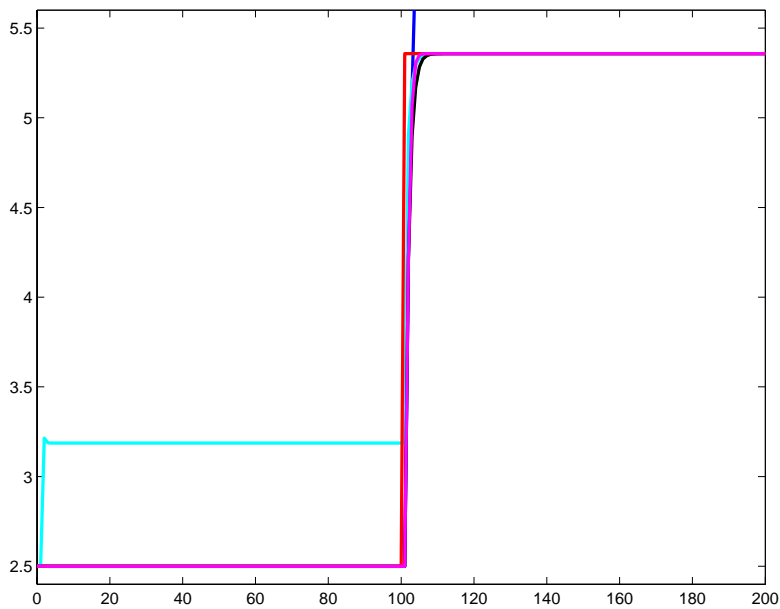


Figura 9.10: Experimento D. Control por planificación de ganancia de P_1 a P_2 : punto inicial 2.5, punto final 5.3572.

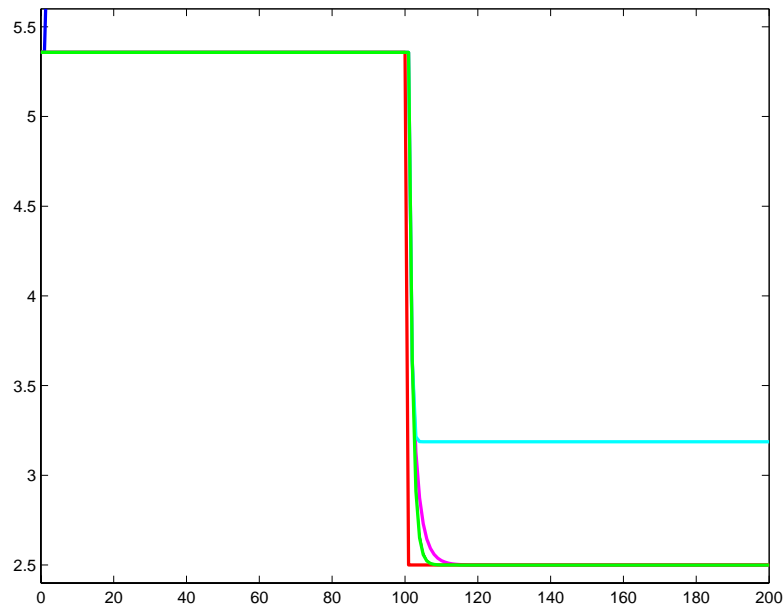


Figura 9.11: Experimento E. Control por planificación de ganancia de P_2 a P_1 : Punto inicial 5.3572, punto final 2.5.

decir que el mejor comportamiento lo muestra el controlador con acción integral, pero en cualquier caso sería deseable una mejora. ■

Queda por tanto clarificado que obviar la existencia de los términos no homogéneos para el diseño de los controladores, aunque se incluya acción integral, no es muy apropiado e incluso se pueden encontrar ejemplos en los que un controlador así diseñado lleve a sistemas inestables en bucle cerrado (Leith and Leithead, 2000).

Posible mejora: evaluación del error. Con la forma de diseño comentada anteriormente, una propuesta que puede ser interesante es evaluar en cada instante k el error $\epsilon(k)$ (9.11) debido a aplicar una combinación de controladores lineales para controlar un sistema de naturaleza no lineal (Hunt and Johansen, 1997). El valor instantáneo de este error $\epsilon(k)$ puede ser acotado de la forma (9.11):

$$\begin{aligned}
 |\epsilon(k)| &\leq \varepsilon_I(k) = (\alpha + \gamma)K(|y(k)| + \dots + |y(k-n)| + |u(k)| + \dots + |u(k-n)| + 1) \\
 \alpha &= \max_{i=1, \dots, n_\phi} \left(\max_{h \in N_i} \|c_i - c_h\|_2 \right) \\
 \gamma &= \max_{h=1, \dots, n_\phi} \left(\sup_{k \geq 0} |\mu_h(\phi(k-1)) - \mu_h(\phi(k))| \right)
 \end{aligned} \tag{9.11}$$

donde K es una constante, u es la entrada e y la salida del sistema. Siguiendo con la notación habitual en esta Tesis, c corresponde a los puntos de operación, μ a las funciones de pertenencia y ϕ la variable de planificación. El resto de parámetros son: n_ϕ el número de modelos locales de los que se dispone y N_i el conjunto de regimenes de operación que son los vecinos más cercanos al regimen i .

El valor de dicha función se puede mantener relativamente pequeño cuando hay un suficiente número de modelos y la variación del punto de operación es lo suficientemente lenta, con lo que $\epsilon(k)$ se puede emplear como una cota superior en el diseño de controladores con técnicas de control robusto.

Si se construyen los modelos del sistema con un rango suficientemente amplio en el entorno del punto de operación, es decir, la planta realmente responde a un sistema linealizable, el valor de $\epsilon(k)$ se reduce de forma considerable (Díez and Previdi, 2001). Además, si se dispone de cierta libertad en la selección de las formas de las funciones de pertenencia, su forma suave, plana en la cercanía del punto de operación y su variación temporal lenta ayudan a mantener $\epsilon(t)$ en valores pequeños, haciendo incluso que α en (9.11) se anule. La identificación con una técnica de agrupamiento como AFCRC, en la que la forma de las funciones de pertenencia se incluye en su índice de coste, ayuda a encontrar (si existe) el modelo del sistema más adecuado en este sentido (Díez *et al.*, 2001).

9.2.3. Aplicabilidad de controladores basados en modelos LPV del sistema linealizados por velocidad

De nuevo se ilustrará mediante un ejemplo el método de diseño, en este caso para controladores basados en modelos LPV del sistema linealizados por velocidad, describiendo problemas y proponiendo soluciones para mejorar el comportamiento del controlador final.

Ejemplo 9.2. Considérese de nuevo el sistema no lineal dado por la ecuación (9.1).

Identificación de modelos locales por velocidad. Desde los modelos locales (linealizaciones) identificados y dados en la tabla 9.1 se puede contruir un modelo linealizado por velocidad, linealización por incrementos $\Delta\hat{y}_k = \hat{y}_k - \hat{y}_{k-1}$ en el caso discreto, con los modelos locales de la tabla 9.3.

Punto de operación	Modelo incremental
(0,2.5,4.5)	$\Delta\hat{y}_{k+1} = 0,8\Delta\hat{y}_k + 0,3\Delta u_k$
(0,5.3572,7.5)	$\Delta\hat{y}_{k+1} = 1,3\Delta\hat{y}_k + 0,3\Delta u_k$

Tabla 9.3: Modelos locales del sistema (9.1).

Integración de modelos locales por velocidad. Para evaluar la aplicabilidad de las funciones de pertenencia identificadas (ver figura 9.3) a la integración de los nuevos modelos, se empleará el método descrito en el Ejemplo 9.1. De este modo se determinarán las funciones de pertenencia necesarias para realizar un buen control con estos modelos locales por incrementos en los que el offset ha desaparecido y se compararán con las identificadas.

Diseño de controladores locales por velocidad. Para asignar en este caso un polo en 0.4, la acción de control necesaria en cada punto de operación se puede obtener resolviendo el sistema de ecuaciones formado por el correspondiente modelo local de la tabla 9.3 y (9.12), donde r es la referencia a seguir por el sistema.

$$\Delta \hat{y}_{k+1} - \Delta r_k = 0,4(\Delta \hat{y}_k - \Delta r_k) \quad (9.12)$$

Para el modelo local por incrementos en el punto de operación (0, 2.5, 4.5) la ecuación a resolver es:

$$0,8\Delta y_k + 0,3\Delta u_k - \Delta r_k = 0,4(\Delta \hat{y}_k - \Delta r_k) \quad (9.13)$$

lo que resulta, despejando Δu_k en (9.13), en el controlador dado por la acción de control (9.14):

$$\Delta u_k^{(VEL1)} = \frac{(0,4 - 0,8)}{0,3}\Delta y_k + \frac{(1 - 0,4)}{0,3}\Delta r_k = -1,33\Delta y_k + 2\Delta r_k \quad (9.14)$$

En el caso del modelo local en el punto de operación (0,5.3572,7.5) la ecuación a resolver es:

$$1,3\Delta y_k + 0,3\Delta u_k - \Delta r_k = 0,4(\Delta \hat{y}_k - \Delta r_k) \quad (9.15)$$

lo que resulta, despejando Δu_k en (9.15), en el controlador dado por la acción de control (9.16):

$$\Delta u_k^{(VEL2)} = \frac{(0,4 - 1,3)}{0,3}\Delta y_k + \frac{(1 - 0,4)}{0,3}\Delta r_k = -3\Delta y_k + 2\Delta r_k \quad (9.16)$$

La teoría de planificación de ganancia por velocidad (4.9) lleva a un controlador global del sistema como combinación de los anteriores:

$$\Delta u_k^{(VEL)} = \mu_1 \Delta u_k^{(VEL1)} + \mu_2 \Delta u_k^{(VEL2)} \quad (9.17)$$

donde μ_1 y μ_2 están, de nuevo, por definir y podrían ser, por ejemplo, las funciones de pertenencia identificadas (figura (9.3)).

Integración de controladores locales por velocidad. Las funciones de pertenencia que idealmente deberían haberse identificado para, desde las acciones de control por velocidad obtenidas para los modelos locales por incrementos, obtener la acción de control adecuada al sistema se pueden calcular resolviendo en este caso el sistema de ecuaciones formado por (9.17) y el controlador que, con las mismas especificaciones que los modelos locales, habríamos calculado analíticamente para el sistema no lineal. Dicho controlador se obtiene aplicando las especificaciones (9.12) al modelo incremental del sistema (9.1) dado por (9.18):

$$\Delta y_{k+1} = 0,0875\Delta y_k^2 + 0,3625\Delta y_k + 0,3\Delta u_k \quad (9.18)$$

donde los incrementos se definen como en todas las linealizaciones por velocidad hasta el momento y $\Delta y_k^2 = y_k^2 - y_{k-1}^2$. Al aplicar las especificaciones (9.12) al modelo incremental (9.18) se obtiene:

$$0,0875\Delta y_k^2 + 0,3625\Delta y_k + 0,3\Delta u_k - \Delta r_k = 0,4(\Delta y_k - \Delta r_k) \quad (9.19)$$

de donde al despejar se obtiene una acción de control incremental:

$$\Delta u_k^{(INC)} = -0,2917\Delta y_k^2 + 0,1250\Delta y_k + 2\Delta r_k \quad (9.20)$$

Como ya se vio en el ejemplo 9.1, desde el sistema de ecuaciones formado por (9.17) y (9.20) (teniendo en cuenta condición de partición borrosa $\mu_1 + \mu_2 = 1$) podremos despejar las pertenencias adecuadas para una correcta interpolación:

$$\mu_1^{(INC)}(y_k, y_{k-1}) = \frac{-0,2917\Delta y_k^2 + 3,1250\Delta y_k}{1,6667\Delta y_k} \mu_2^{(INC)}(y_k, y_{k-1}) = 1 - \mu_1^{(INC)}(y_k, y_{k-1}) \quad (9.21)$$

La ecuación de $\mu_1^{(INC)}$, dado que $\Delta y_k^2 = y_k^2 - y_{k-1}^2$ no se puede simplificar, tampoco corresponde a una interpolación lineal de controladores. Sin embargo, al deshacer los incrementos y simplificar (haciendo $y_k^2 - y_{k-1}^2 = [y_k + y_{k-1}][y_k - y_{k-1}]$) se obtiene:

$$\begin{aligned} \mu_1^{(INC)}(y_k, y_{k-1}) &= \frac{-0,2917(y_k^2 - y_{k-1}^2) + 3,1250(y_k - y_{k-1})}{1,6667(y_k - y_{k-1})} = \\ &= \frac{-0,2917(y_k + y_{k-1})(y_k - y_{k-1}) + 3,1250(y_k - y_{k-1})}{1,6667(y_k - y_{k-1})} = \\ &= \frac{-0,2917(y_k + y_{k-1}) + 3,1250}{1,6667} = \frac{-0,2917(y_k + y_{k-1}) + 3,1250}{1,6667} \end{aligned} \quad (9.22)$$

que en el caso de una discretización apropiada en la que los incrementos sean lo suficientemente pequeños e $y_k \cong y_{k-1}$ corresponde a:

$$\begin{aligned} \mu_1^{(INC)}(y_k, y_{k-1}) &= \frac{-0,2917(y_k + y_{k-1}) + 3,1250}{1,6667} \cong \frac{-0,2917 \times 2y_k + 3,1250}{1,6667} = \\ &= -0,35y_k + 1,8750 = \mu_1^{(INC)}(y_k) \end{aligned} \quad (9.23)$$

que es una función lineal con valor 1 en $y(k) = 2,5$ y 0 en $y(k) = 5,3572$, idéntica a la función de pertenencia de la gráfica 9.3, lo que nos indica que la interpolación con las funciones de pertenencia lineales (identificadas) será una buena opción con estos controladores. Esto es debido a que la linealización por incrementos, al hacer desaparecer el término independiente de los modelos, convierte los modelos locales afines de TS en modelos locales homogéneos de TS que se pueden interpolar sin problemas con (5.4), pero que no aproximan funciones correctamente sin el marco adicional de las linealizaciones por velocidad.

Evaluación del rendimiento. La tabla 9.4 resume los resultados de las simulaciones realizadas con el controlador por velocidad propuesto (en comparación con los locales en los puntos de operación y el teórico no lineal) y, en aquellos casos en los que aparece un error en régimen permanente, un nuevo controlador $u_k^{(INT)}$ con un incremento de acción de control debido a acción integral $\Delta u_k^{INT} = K_i(r(k) - y(k))$ (con K_i a definir) sumado al incremento $u_k^{(INC)}$ de la ecuación 9.20. Todas estas simulaciones se pueden observar en las figuras 9.12 a 9.16.

EXPERIMENTO	ACCIÓN DE CONTROL				
	$u_k^{(VEL1)}$ (azul oscuro)	$u_k^{(VEL2)}$ (azul claro)	$u_k^{(VEL)}$ (verde)	$u_k^{(INT)}$ (verde disc.)	$u_k^{(INC)}$ (negro)
A. Entorno de P_1 (figura 9.12).	Pequeño error de posición	Gran error de posición	Correcto	-	Correcto
B. Entorno de P_2 (figura 9.13).	Gran error de posición	Pequeño error de posición	Correcto	-	Correcto
C. Entorno zona intermedia (figura 9.14).	Gran error de posición	Gran error de posición	Correcto	-	Correcto
D. Salto de P_1 a P_2 (figura 9.15).	Gran error de posición	Gran error de posición	Pequeño error de posición	Tiempo de establecimiento	Correcto
E. Salto de P_2 a P_1 (figura 9.16).	Gran error de posición	Gran error de posición	Pequeño error de posición	Sobreoscilación	Correcto

Tabla 9.4: Comportamiento, en experimentos a lo largo del rango de operación, de los diferentes controladores por velocidad diseñados.

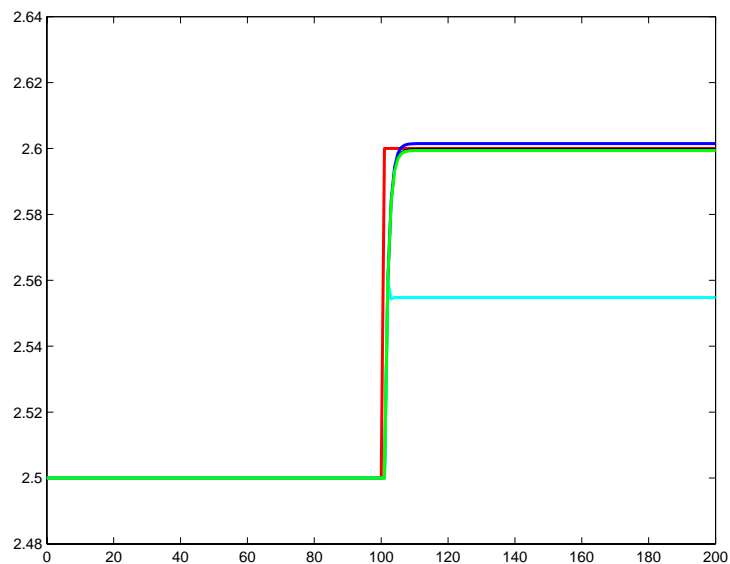


Figura 9.12: Experimento A. Control por velocidad en el entorno de P_1 : punto inicial 2.5, punto final 2.6.

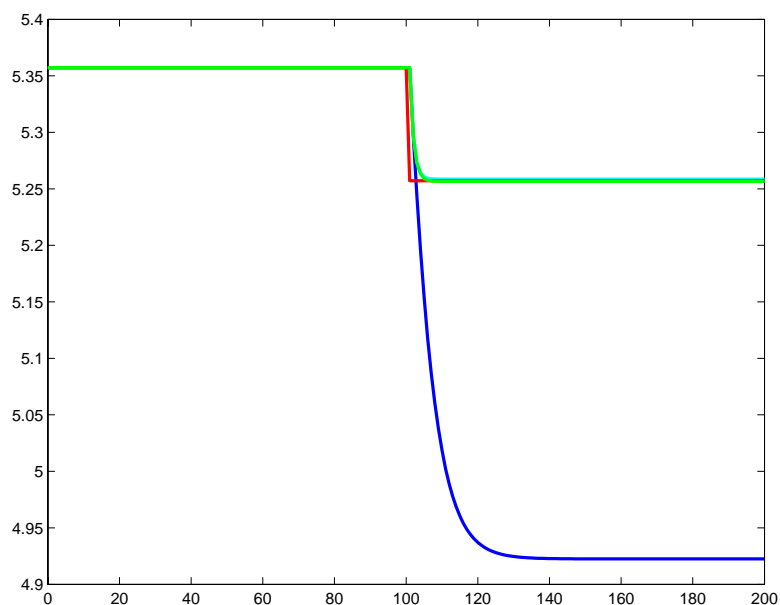


Figura 9.13: Experimento B. Control por velocidad en el entorno de P_2 : punto inicial 5.3572, punto final 5.2572.

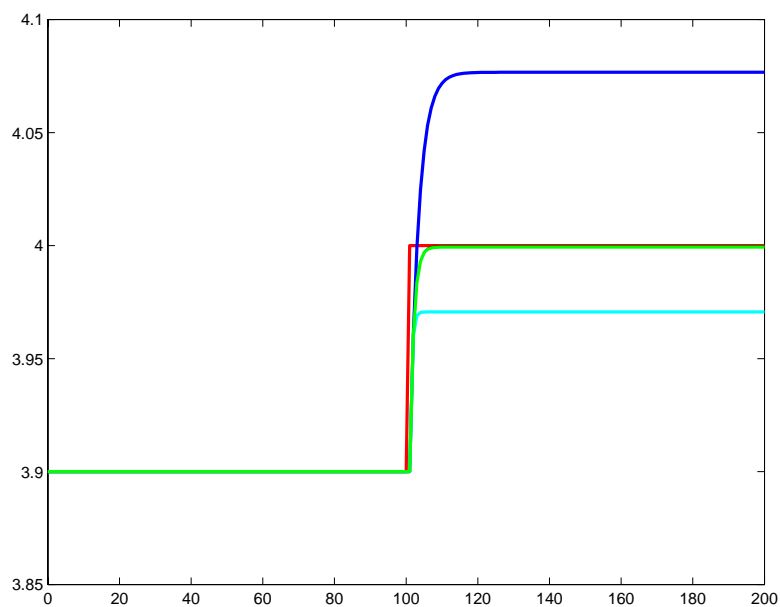


Figura 9.14: Experimento C. Control por velocidad en el entorno de un punto intermedio: punto inicial 3.9, punto final 4.

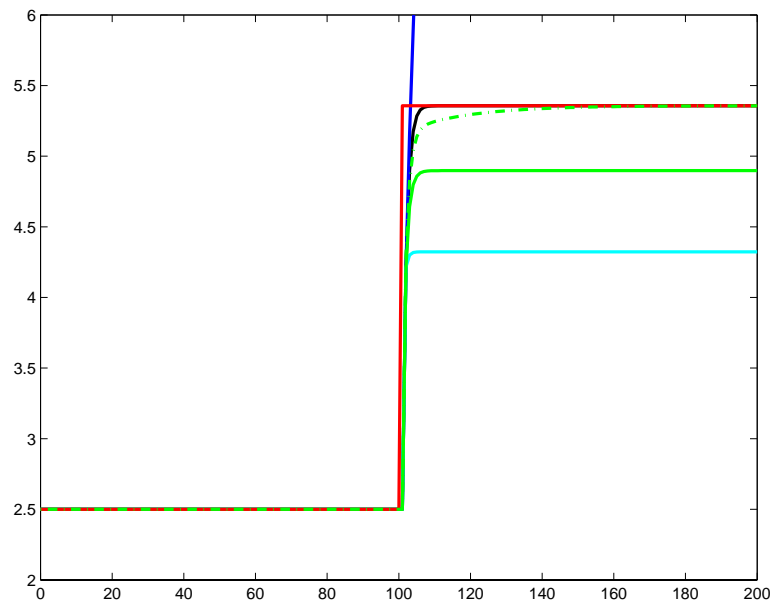


Figura 9.15: Experimento D. Control por velocidad de P_1 a P_2 : punto inicial 2.5, punto final 5.3572.

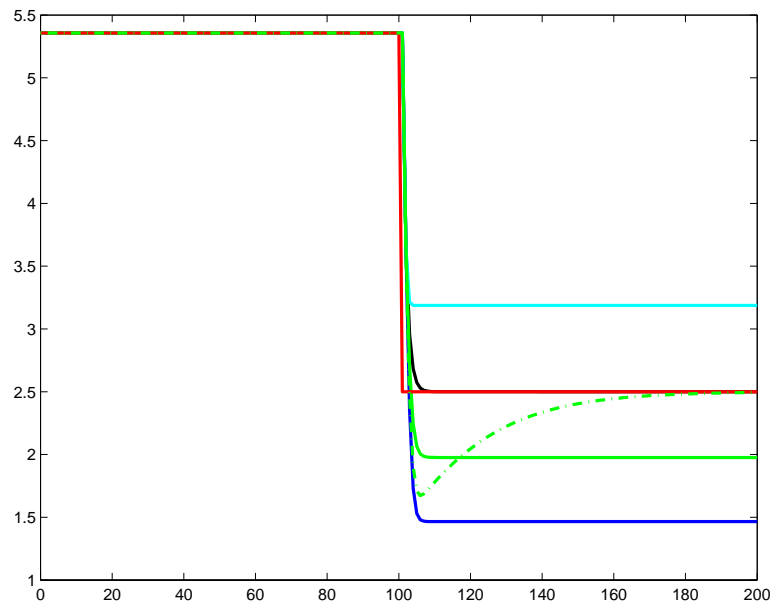


Figura 9.16: Experimento E. Control por velocidad de P_2 a P_1 : punto inicial 5.3572, punto final 2.5.

Las familias de controladores por incrementos interpoladas linealmente superan en todos los casos de pequeñas variaciones a las familias de controladores en los puntos de operación interpoladas linealmente (figuras 9.7 a 9.11) y por eso será el controlador más apropiado para el sistema, ya que esta será la forma habitual de operación del sistema. En aquellos casos en los que aparecen errores de posición, éstos se pueden subsanar con la adición de acción integral.

■

Posible mejora: linealización por incrementos parcial. El controlador elegido (9.20) está basado tan solo en las variaciones incrementales de la variables, lo que puede llevar a una situación en la que un error inicial en el seguimiento de una referencia constante (debido, por ejemplo, a unas condiciones iniciales no adecuadas) no sea nunca subsanado. Este error en régimen permanente se elimina fácilmente aplicando una acción integral en paralelo con el controlador diseñado de la forma comentada anteriormente, pero que habrá que ajustar experimentalmente.

Otra opción más elegante consiste en rehacer el criterio de diseño para que aparezca explícitamente la acción integral, además de los incrementos, mediante alguna medida del error en el seguimiento de la referencia. Esto se puede hacer partiendo de la especificación (9.2) y restando y_k a ambos lados (en lugar de tomar incrementos en todas las variables, con lo que se llegaría a (9.12)) se obtiene una expresión (9.24) equivalente a (9.12) en la que se explicita el error de seguimiento de la referencia, con lo que solo se hace una linealización por incrementos parcial.

$$\begin{aligned}
 \hat{y}_{k+1} - r_k &= 0,4(\hat{y}_k - r_k) \\
 &\Downarrow \\
 \hat{y}_{k+1} - \hat{y}_k - r_k &= 0,4(\hat{y}_k - r_k) - \hat{y}_k \\
 &\Downarrow \\
 \Delta\hat{y}_{k+1} &= 0,4(\hat{y}_k - r_k) - \hat{y}_k + r_k \\
 &\Downarrow \\
 \Delta\hat{y}_{k+1} &= (1 - 0,4)(r_k - \hat{y}_k)
 \end{aligned} \tag{9.24}$$

Por tanto, dado un modelo incremental genérico de la forma $\Delta\hat{y}_{k+1} = A_i\Delta u_k + B_i\Delta\hat{y}_k$, la acción de control se podrá calcular para cada controlador i despejando como en (9.25)

$$\begin{aligned}
 A_i\Delta u_{k\ i} + B_i\Delta\hat{y}_k &= (1 - 0,4)(r_k - \hat{y}_k) \\
 &\Downarrow \\
 \Delta u_{k\ i} &= \frac{1 - 0,4}{A_i}(r_k - \hat{y}_k) - \frac{B_i}{A_i}\Delta\hat{y}_k
 \end{aligned} \tag{9.25}$$

La figura 9.17 muestra como un controlador diseñado con la linealización parcial por incrementos propuesta supera, a lo largo de todo el rango de funcionamiento del

sistema descrito en el ejemplo 9.2, tanto al controlador por planificación de ganancia tradicional como al controlador por velocidad que mostraba el mejor comportamiento en el ejemplo anterior. De este modo, este nuevo método de diseño de controladores será el seleccionado para desarrollar un ejemplo más detallado en el capítulo siguiente.

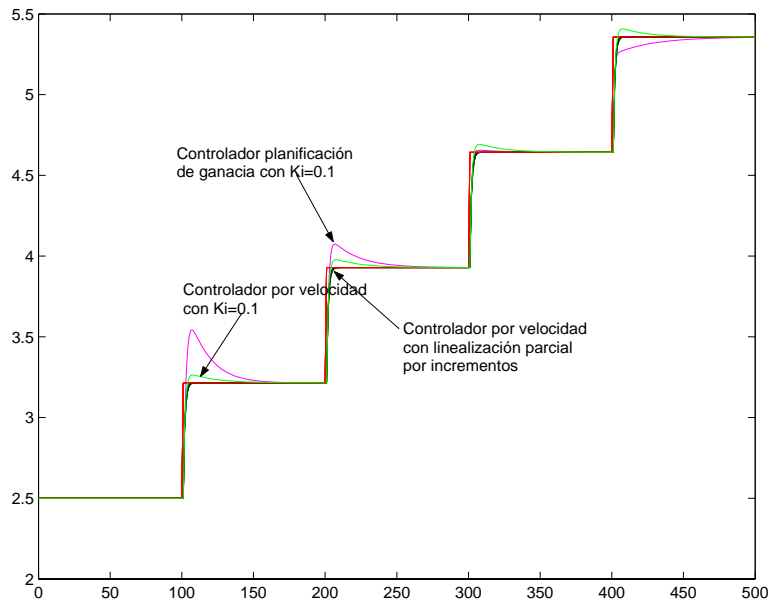


Figura 9.17: Funcionamiento de todos los controladores a lo largo de la zona de trabajo.

9.3. Control por LMIs

De forma general, los métodos basados en las representaciones de Takagi-Sugeno o redes de modelos locales no tienen por qué seguir la filosofía de división del problema en un conjunto de subproblemas lineales, sino que los modelos obtenidos son una combinación de múltiples modelos (solo válidos en el punto de operación) cada uno de los cuales no tiene por qué responder a un comportamiento lineal del sistema en el entorno de un punto de operación (linealización), sino que es la combinación de todos ellos la que aproxima el modelo al sistema real (Díez *et al.*, 2001). La interpolación se convierte en este caso en la parte más importante de este método, ya que no hay modelos que dominen en cada región de operación sino que varios de ellos contribuyen a la descripción del funcionamiento del sistema. Esta interpretación se aleja de la idea de los métodos de planificación de ganancia clásicos, los métodos de diseño se asemejan más a los que trabajan con representaciones quasi-LPV y se basan en la resolución de sistemas LMI (Linear Matrix Inequalities) (Tanaka and Sano, 1994; Wang *et al.*, 1996; Tanaka *et al.*, 1998; Kiriakidis, 1999).

En el caso de modelos homogéneos (sin término independiente u offset, como los modelos por velocidad) la aplicabilidad de las técnicas vistas en el apartado 4.3 es directa. Sin embargo, en el caso de modelos no homogéneos (como las redes de modelos locales o los modelos TS con consecuente afín) se hace necesario un tratamiento específico para

considerar el término adicional y comprobar su estabilidad mediante técnicas LDIs (ver sección 4.3.3).

Este tipo de control, sin pérdida de generalidad, puede ser ejercido por ejemplo empleando una base de reglas borrosas (Carbonell *et al.*, 2000) que actúe como elemento supervisor de ajuste de la ganancia de control por estimación del punto de trabajo a partir de leyes de control local:

$$R_j : \quad \text{Si } \mathbf{u}(k-1) \text{ es } \mathbf{U}_j \text{ y } \mathbf{x}(k) \text{ es } \mathbf{X}_j \text{ entonces} \quad (9.26)$$

$$\mathbf{u}(k) = \mathbf{u}_{0j} + \mathbf{K}_j (\mathbf{x}(k) - \mathbf{x}_{0j})$$

La utilidad del empleo de modelos locales lineales consiste en permitir el planteamiento a su vez de leyes de control local lineales estables, según un diseño adecuado de la realimentación del estado.

La expresión que se obtiene para el lazo cerrado viene dada por:

$$\mathbf{x}(k+1) = \sum_{i=1}^N \sum_{j=1}^N \mu_i(k) \mu_j(k) [(\nabla_{y_i} + \nabla_{u_i} K_j)] (\mathbf{x}(k) - \mathbf{x}_{0i}) \quad (9.27)$$

siendo $\mu_i(k)$ la función base borrosa correspondiente a la regla i en el instante k .

La resolución del problema LMI que se plantea es puramente matemática y olvida (no necesita) la interpretabilidad del sistema, pudiendo llegar a soluciones válidas a nivel matemático y de estabilidad.

Ejemplo 9.3. Sea de nuevo el sistema no lineal dado por la ecuación (9.1). Desde los modelos locales (linealizaciones) identificados y dados en la tabla 9.1 se puede construir la descripción del sistema con LDI como se vio en el apartado 4.3

$$G_1 = \begin{bmatrix} \frac{\partial f_1}{\partial x} & \frac{\partial f_1}{\partial u} \\ \frac{\partial g_1}{\partial x} & \frac{\partial g_1}{\partial u} \end{bmatrix} = \begin{bmatrix} 0,3 & 0,8 \\ 1 & 0 \end{bmatrix}, \quad (9.28)$$

$$G_2 = \begin{bmatrix} \frac{\partial f_2}{\partial x} & \frac{\partial f_2}{\partial u} \\ \frac{\partial g_2}{\partial x} & \frac{\partial g_2}{\partial u} \end{bmatrix} = \begin{bmatrix} 0,3 & 1,3 \\ 1 & 0 \end{bmatrix}.$$

A esta descripción se le pueden aplicar los métodos de diseño de controladores lineales que se desee según lo visto en 4.3.

Un análisis de estabilidad de G_1 y G_2 según Lyapunov lleva a determinar la inestabilidad del sistema que representa la LDI G_2 . La resolución del problema LMI, en el que se ha definido $X = P^{-1}$ e $Y = KX$, consiste en encontrar X e Y comunes que satisfagan

$$X > 0, \quad (9.29)$$

$$0,3X + 0,8Y + X0,3 + Y0,8 < 0,$$

$$0,3X + 1,3Y + X0,3 + Y1,3 < 0,$$

o determinar que X no existe. El problema es factible y su resolución, tras deshacer los cambios de variable, lleva a una ganancia de realimentación $K = -6$ que estabiliza el sistema, ya que el valor de P es 0.0165.



Ejemplo 9.4. Sea el sistema no lineal:

$$\dot{x}(t) = \begin{cases} \begin{bmatrix} 2 & 1 \\ 1 & 1 \end{bmatrix} x(t) + \begin{bmatrix} 1 \\ 2 \end{bmatrix} u(t) + f(x_1, y_1), & (x, u) \cong (x_1, u_1), \\ \begin{bmatrix} -2 & 1 \\ 1 & -1 \end{bmatrix} x(t) + \begin{bmatrix} 1 \\ 2 \end{bmatrix} u(t) + f(x_2, u_2), & (x, y) \cong (x_2, u_2), \end{cases} \quad (9.30)$$

Su descripción del sistema con LDIs, de forma que se puedan aplicar los métodos de diseño de controladores lineales que se desee, es

$$\dot{x}(t) = \begin{cases} \begin{bmatrix} 2 & 1 \\ 1 & 1 \end{bmatrix} x(t) + \begin{bmatrix} 1 \\ 2 \end{bmatrix} u(t), & (x, u) \cong (x_1, u_1), \\ \begin{bmatrix} -2 & 1 \\ 1 & -1 \end{bmatrix} x(t) + \begin{bmatrix} 1 \\ 2 \end{bmatrix} u(t), & (x, u) \cong (x_2, u_2). \end{cases} \quad (9.31)$$

Un análisis de estabilidad según Lyapunov de este sistema lleva a determinar la inestabilidad del primero de ellos. La resolución del problema LMI para determinar una ganancia que estabilice todo el sistema, se hace definiendo $X = P^{-1}$ e $Y = KX$, y encontrando X e Y comunes que satisfagan

$$\begin{aligned} X &> 0, \\ \begin{bmatrix} 2 & 1 \\ 1 & 1 \end{bmatrix} X + \begin{bmatrix} 1 \\ 2 \end{bmatrix} Y + X \begin{bmatrix} 2 & 1 \\ 1 & 1 \end{bmatrix}^T + Y \begin{bmatrix} 1 \\ 2 \end{bmatrix}^T &< 0, \\ \begin{bmatrix} -2 & 1 \\ 1 & -1 \end{bmatrix} X + \begin{bmatrix} 1 \\ 2 \end{bmatrix} Y + X \begin{bmatrix} -2 & 1 \\ 1 & -1 \end{bmatrix}^T + Y \begin{bmatrix} 1 \\ 2 \end{bmatrix}^T &< 0, \end{aligned} \quad (9.32)$$

o determinar que X no existe. El problema es factible y su resolución, tras deshacer los cambios de variable lleva a una ganancia de realimentación $K = [-6,78 \quad -0,76]$ que estabiliza el sistema dado que $P = \begin{bmatrix} 43,7 & 0,028 \\ 0,03 & 2,503 \end{bmatrix}$.



9.4. Conclusiones

En este capítulo se han presentado las diferentes alternativas de control que son aplicables a un conjunto de modelos lineales locales de un sistema no lineal, como los que se obtienen de la aplicación de los métodos de identificación descritos a lo largo de los capítulos anteriores y que consisten en las técnicas de planificación de ganancia para familias de modelos LPV. Por sus diferentes ventajas, han demostrado ser buenas opciones los controladores basados en modelos LPV en puntos de operación y los relacionados con la linealización por velocidad, especialmente los últimos por ser más sencilla de encontrar (basada en las identificadas) la función para la interpolación.

Por otra parte, los métodos que emplean las LMIs permiten desarrollar de forma sencilla controladores globales estables para el sistema, aunque más conservadores que los anteriores.

Las aportaciones de este capítulo han sido:

- Analizar la aplicabilidad de las técnicas de planificación de ganancia a los modelos identificados con el algoritmo AFCRC,
- Aplicar controladores basados en puntos de operación, determinando los problemas generados por el término independiente de los modelos y la interpolación de éstos, proponiendo:
 - estructuras que minimizan el efecto del offset (basadas en integradores), y
 - cotas al error debido a la interpolación (que se hace pequeño con variaciones lentas en la referencia y funciones de pertenencia suaves con núcleos amplios), ya que las funciones de interpolación apropiadas no se corresponden con las funciones de pertenencia identificadas,
- Aplicar igualmente los controladores por velocidad (en los que no hay problemas con los términos independientes), los cuales se ha manifestado que permiten (bajo condiciones de discretización apropiada) la interpolación con las funciones de pertenencia identificadas. El problema de estos controladores en cuanto al uso de variables incrementales y la imposibilidad de subsanación de errores constantes (excepto con integradores) se puede solventar con las estructuras de linealización parcial por incrementos que se han propuesto,
- Aplicar finalmente las técnicas de LMIs al diseño y análisis de estabilidad de controladores en diferentes puntos de operación.

Capítulo 10

Ejemplos de aplicación de los algoritmos

*Si sale, sale.
Si no sale, hay que volver a empezar.
Todo lo demás son fantasías.
Édouard Manet.*

En el presente capítulo se emplearán los algoritmos de identificación de modelos locales propuestos a lo largo de la Tesis a dos casos: el modelado de un sensor y la identificación de un sistema térmico sencillo. En este segundo ejemplo, adicionalmente, se mostrará como desarrollar un controlador por planificación de ganancia adecuado.

10.1. Modelado de un sensor de biomasa

Para la estimación en línea de la concentración de biomasa en fermentadores, en el Departamento de Ingeniería de Sistemas y Automática se ha desarrollado un sensor basado en la medida de la absorbancia de una muestra del medio. Este dispositivo permite variar su ganancia a través de la regulación de la intensidad de un LED de referencia y la tensión que ofrece es función de la concentración de biomasa en el medio.

Debido a que la relación entre la tensión de salida del sensor (H_r) y la concentración de biomasa (c_H) es no-lineal, se hace necesario un modelado (para la posterior calibración del sensor) mediante algún método de identificación de la relación entre ambas variables. Para este proceso se han registrado diferentes valores de la tensión de salida del sensor (H_r) para distintas concentraciones de biomasa (c_H), así como los valores de la tensión en el led de referencia (v_l). La superficie a identificar es la que representan los datos de la figura 10.1.

10.1.1. Identificación de modelos locales

Se realizará la identificación de cuatro modelos locales lineales que se ajusten a la superficie mediante el algoritmo AFCRC. Los parámetros adecuados para la ejecución

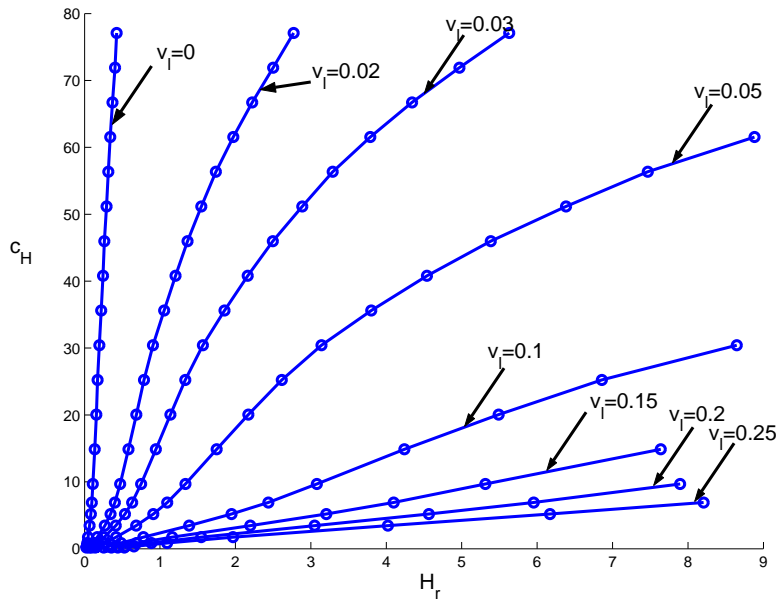


Figura 10.1: Relación no lineal entre las variables del sensor.

del algoritmo AFCRC son, tras normalizar los datos y seguir los pasos sugeridos en el apartado 8.5:

- Obtener la máxima distancia d_{max} posible con los datos disponibles. Los datos normalizados varían en el rango $[-0,91, 2,49]$ para v_l y en el rango $[-0,85, 3,12]$ para H_r , con lo que $d_{max} = \sqrt{(2,49 + 0,91)^2 + (3,12 + 0,85)^2} = 5,23$.
- Determinar el valor de σ_1 . Dado que tomaremos cuatro clases, la distancia máxima de dos puntos de una clase será a lo sumo $d_1 = 5,23/4 = 1,3 = 6\sigma_1$, con lo que $\sigma_1^2 = 0,048$.
- Calcular un valor para σ_2 . Procediendo como en el caso anterior, hacemos que la distancia a la que se encuentran la mayoría de los puntos de la clase es la mitad de d_1 , obtenemos $\sigma_2^2 = 0,012$.
- Ajustar, mediante diferentes ejecuciones, los valores para γ_1 y γ_2 de forma que las componentes en el índice de coste de AFCRC debidas a J_{lejos} y J_{cerca} no superen el 10% del total. Con ello se llega a $\gamma_1 = 10^{-2}$ y $\gamma_2 = 10^{-2}$.

La aplicación del algoritmo AFCRC con los parámetros anteriores lleva, tras 44 iteraciones, a la identificación del sistema con los modelos presentados en la tabla 10.1. En la figura 10.2 se muestra la asignación de datos a clases, según el criterio de mayor pertenencia.

10.1.2. Validez de los modelos locales

El proceso de identificación proporciona los puntos prototipos de las clases y los modelos locales (planos en este caso) en el entorno de cada uno de ellos. Como se

Pto. operación (u, y)	Modelo de c_H
$(0.027, 1.56)$	$-901.65 v_l + 122.29 H_r + 386.91$
$(0.18, 2.74)$	$-44.44 v_l + 1.37 H_r + 8.39$
$(0.02, 0.25)$	$-161.70 v_l + 166.29 H_r + 3.12$
$(0.04, 3.49)$	$-685.74 v_l + 33.23 H_r + 68.16$

Tabla 10.1: Modelos locales identificados por el algoritmo AFCRC.

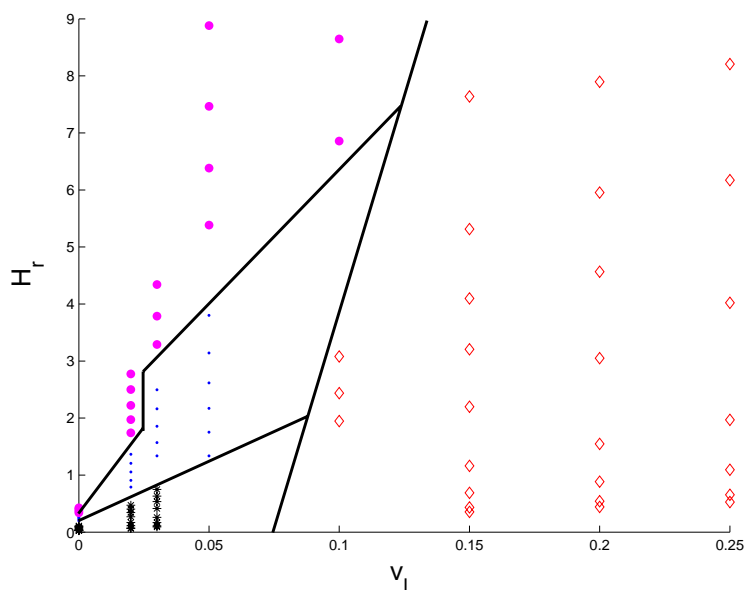


Figura 10.2: Clases detectadas por el algoritmo AFCRC.

mostró en el ejemplo 5.1 de la sección 5.2, para integrar todos estos modelos locales en un único modelo borroso no lineal se ha de emplear alguna técnica de interpolación apropiada. Esta integración no estará basada únicamente en una ponderación afectada por la pertenencia (como se hace en los modelos TS originales), sino en algún método de interpolación convexa con las características definidas en el apartado 7.2. Precisamente en el apartado 7.2 se propuso una solución para el caso monovariante, pero no se ha encontrado una extensión apropiada para el caso multivariante que nos ocupa.

Lo que sí es posible determinar con los resultados del proceso de identificación es el rango de validez de cada uno de los modelos, es decir, las zonas de las clases en las que los puntos tienen altas pertenencias. Para ello se puede emplear la información que proporciona la forma de las funciones de pertenencia (ver apartado 6.7.5), bien utilizando las mismas funciones de pertenencia identificadas (si hay suficiente información) o ajustando los parámetros de alguna superficie con forma similar a las funciones de pertenencia.

La aproximación empleada en este caso, de las comentadas en 6.7.5, es la que se basa en medir la distancia a los puntos de operación del punto del cual se desea obtener una pertenencia y transformar estas distancias en un grado de pertenencia haciendo la media ponderada de la inversa de todas ellas. Para ello se suelen usar expresiones

similares a las que aparecen en los algoritmos de agrupamiento y en este caso, para seguir la filosofía del algoritmo de identificación AFCRC, se afectará la distancia medida por un término hiperexponencial (Previdi, 1998) realizando posteriormente la inversión. Con todo ello, la evaluación de la pertenencia de un punto k al cluster i queda como se muestra en (10.1).

$$\mu_{ik} = \frac{1}{\sum_{j=1}^c \frac{e^{\frac{L_{ik}^2}{2\sigma^2}} - 1}{e^{\frac{L_{jk}^2}{2\sigma^2}} - 1}} \quad (10.1)$$

Nota 1.1. Hay que resaltar la importancia de la normalización de los datos previamente al cálculo de las distancias en 10.1 pues, de otro modo, grandes diferencias en los valores de las variables involucradas en un experimento pueden llevar a formas inadecuadas de las funciones de pertenencia.

Nota 1.2. Los parámetros N (orden de la función exponencial) y σ permiten ajustar la forma de la función a las pertenencias identificadas teniendo en cuenta, respectivamente, la amplitud de las zonas planas de alta pertenencia así como la anchura de las funciones.

La estimación de las funciones de pertenencia así calculadas para este ejemplo (con $N=1$ y $\sigma = 2$) se puede observar en la figura 10.3.

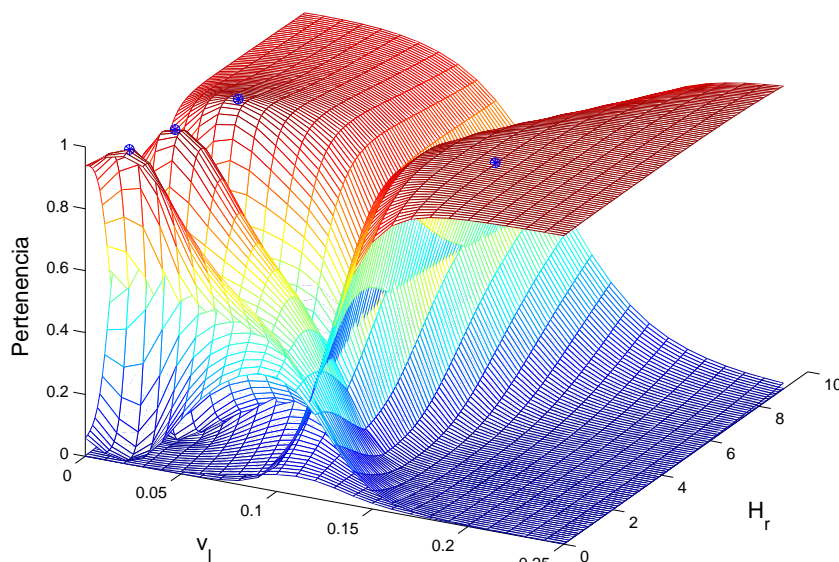


Figura 10.3: Funciones de pertenencia estimadas.

En la figura 10.4 se muestran los modelos locales en la zona en la que se puede confiar en el modelo como un modelo local, que se ha tomado en este caso para pertenencias mayores de 0.8. La figura 10.5 muestra los errores que cada modelo comete en la estimación de cada dato, coincidiendo la zona de bajo error con la cercana al prototipo en todos los casos.

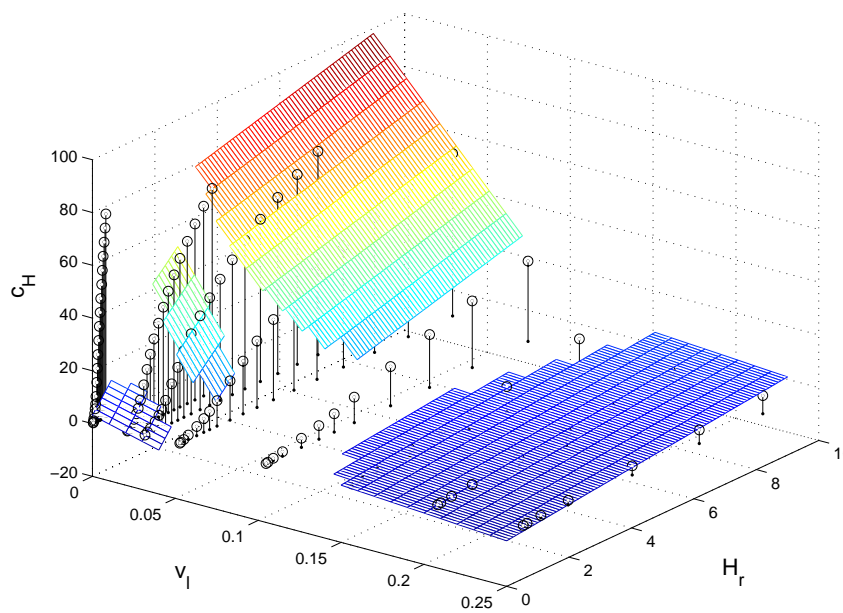


Figura 10.4: Modelos locales identificados.

10.2. Identificación y control de un sistema térmico

En este apartado se va a ejemplificar, mediante la aplicación a un sistema térmico sencillo, el uso de todas las técnicas propuestas a lo largo de esta Tesis para la identificación y control por modelos locales de sistemas no lineales, siguiendo para ello los pasos necesarios para desarrollar un controlador por planificación de ganancia como los que fueron descritos en el capítulo 4.

10.2.1. Descripción del proceso

El proceso consiste en el modelo a escala de un horno como el mostrado en la figura 10.6 en el que se pueden estudiar los comportamientos habituales de los procesos térmicos. El sistema consta esencialmente de una caja de plástico con una resistencia acoplada a un radiador en su interior, a través de la cual circula una intensidad proporcionada por una fuente de tensión controlada. La contribución de energía en el interior del proceso será debida a la potencia disipada por la resistencia. Adicionalmente, hay un ventilador que introduce aire de forma continua y constante desde el exterior, facilitando así la circulación de aire y la homogeneización de temperatura en el interior del proceso. El actuador consiste en una fuente de tensión, cuyo rango de tensión va de 0 a 7.5 Voltios y se corresponde con un rango de actuación desde el exterior del 0 al 100%. Para medir la temperatura en la resistencia hay un termopar tipo K instalado sobre ella el cual proporciona, tras pasar su señal a través de un acondicionador, una tensión de 0 a 10 Voltios.

El propósito de este apartado será identificar un modelo adecuado para el control del proceso térmico desde datos experimentales, así como realizar posteriormente un control de la temperatura de la resistencia generadora de calor.

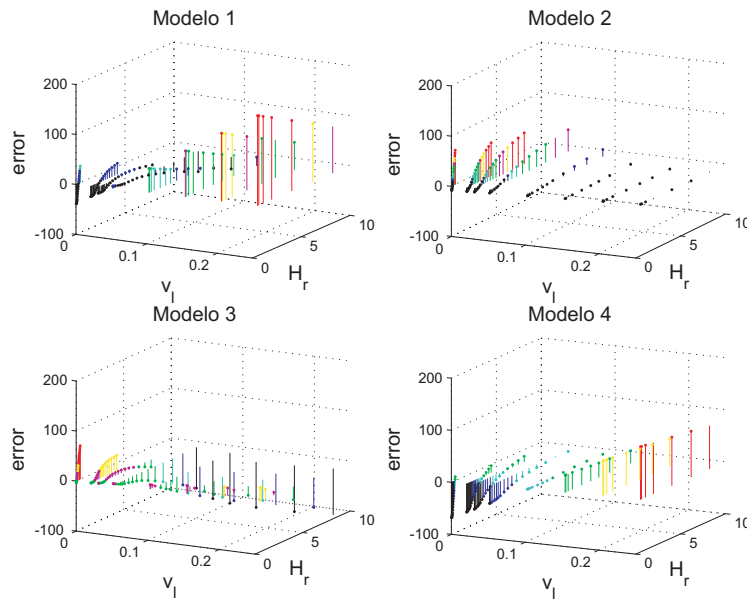


Figura 10.5: Errores de los modelos locales identificados.

El comportamiento del proceso es no lineal respecto a la entrada, dependiendo además de perturbaciones como las variaciones en la temperatura interna del horno y el flujo de renovación del aire.

De acuerdo con las leyes físicas de la termodinámica aplicadas al sistema descrito, un modelo teórico podría ser el definido por la ecuación diferencial (10.2), en la que los fenómenos de conducción y convección se han tenido en cuenta, obviando las pérdidas por radiación en la resistencia por no ser de relevancia a las temperaturas relativamente bajas con las que se trabaja:

$$\dot{y} = k_1 u^2 - k_2 (y - x_i), \quad (10.2)$$

donde:

- y es la temperatura de la resistencia en grados centígrados,
- u es la tensión de entrada del actuador en voltios,
- x_i es la temperatura en la periferia de la resistencia, y
- k_1 y k_2 son parámetros experimentales del modelo.

Los valores de los parámetros del modelo se pueden identificar experimentalmente (Blasco *et al.*, 2001), llegando a $k_1 = 0,0000792^\circ\text{C}/\text{V}^2$ y $k_2 = 0,00536$. Respecto a la temperatura en la periferia de la resistencia, se considerará que es igual a la temperatura ambiente ($y_{amb} = 25^\circ\text{C}$), ya que el flujo constante de aire desde el exterior debido al ventilador así lo permite. Las variaciones en esa temperatura se considerarán, como ya se ha comentado, una perturbación en el sistema.

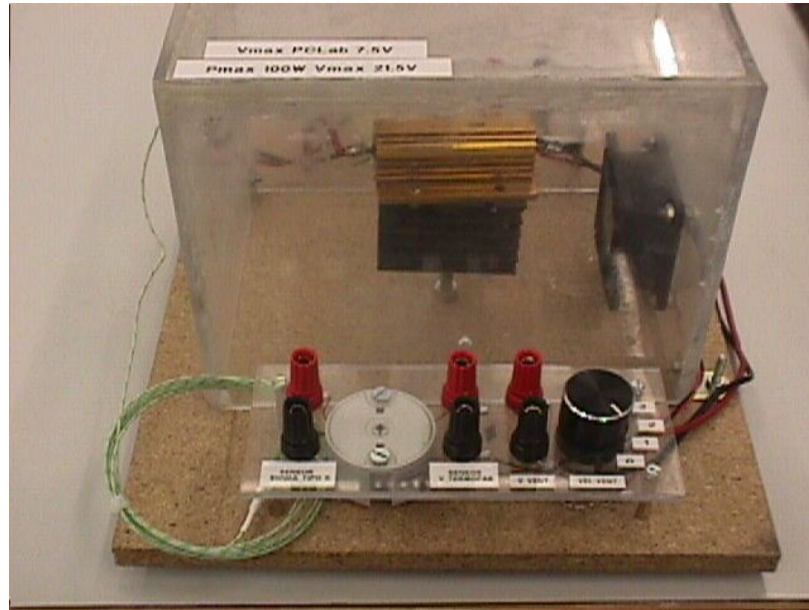


Figura 10.6: Aspecto del horno a modelar y controlar.

Con todo lo anterior, se puede construir el siguiente modelo discreto, inspirado en el modelo (10.2):

$$y_{k+1} = y_k + T(k_1 u_k^2 - k_2(y_k - y_{amb})) \quad (10.3)$$

donde T es el periodo de muestreo elegido que, dado lo lento del proceso a estudiar, 10 segundos se ha mostrado un valor suficiente.

10.2.2. Identificación de un modelo no lineal

La identificación de un proceso es, en términos generales, muy sensible a las series de datos empleadas (apartado 6.7.1). El caso más favorable es aquel en el que, al disponer de un modelo del proceso (como (10.3) en este caso) se pueden generar todas las combinaciones de datos que se deseen. En el caso de emplear el proceso real, disponer de un buen conjunto de datos para identificar el proceso es más complicado y requiere de un adecuado diseño de experimentos. Con el fin de aislar en lo posible la adecuación de las técnicas de identificación propuestas de los errores que puede conllevar no disponer de los datos necesarios (lo que evitaría poder evaluar lo correcto o no de las técnicas), se ha optado por realizar en primer lugar una identificación desde un conjunto de datos simulados y creados con el modelo (10.3). Posteriormente se repetirá el proceso de identificación con un conjunto de datos reales.

Identificación desde datos simulados

En primer lugar se ha optado por el típico ensayo basado en introducir a la entrada un conjunto de escalones de amplitud variable aleatoriamente en el rango de interés y con una duración tal que permita al sistema alcanzar su régimen permanente ante cada uno de los escalones. Este tipo de ensayos tienen la ventaja de su sencillez pero

dada su poca rica (baja) componente frecuencial, aunque permitirá identificar adecuadamente la ganancia del sistema, llevará a errores mayores en cuanto a identificar la dinámica del proceso se refiere. Los datos empleados se pueden observar en la figura 10.7. Si desde los datos generados con el modelo del proceso térmico fuesen identifica-

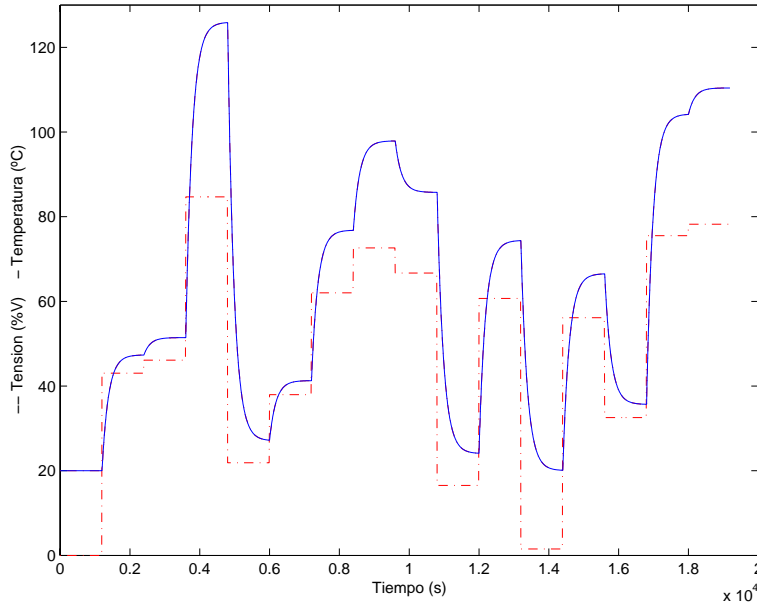


Figura 10.7: Entrada y salida del ensayo para identificación del modelo LPV.

dos perfectamente dos modelos lineales en dos puntos de funcionamiento del sistema, cada uno de ellos debería de corresponder con la linealización del proceso (10.3) en dichos puntos. Esto quiere decir que para un punto de operación formado por el par genérico tensión-temperatura (u_0, y_0) , el modelo identificado debería corresponder con la linealización:

$$\begin{aligned} \hat{y}_{k+1} &= 0,9464 y_k + 0,001584 u_0 u_k + f(u_0, y_0), \\ f(u_0, y_0) &= 0,9464 y_0 + 0,000792 u_0^2 + 1,0720 - (0,9464 y_0 + 0,001584 u_0^2). \end{aligned} \quad (10.4)$$

Previamente al empleo del algoritmo AFCRC para la identificación de modelos, tal y como se sugirió en 8.5, se han de seguir una serie de pasos para obtener unos parámetros adecuados en la ejecución del algoritmo AFCRC que son, tras normalizar los datos y decidir el número de clases a emplear (dos en este caso):

- Obtener la máxima distancia d_{max} posible con los datos disponibles. Los datos normalizados, tanto en tensión como en temperatura, varían en el rango $[-1,5, 2]$, con lo que $d_{max} = \sqrt{(1,5 + 2)^2 + (1,5 + 2)^2} = 4,95$.
- Determinar el valor de σ_1 . Dado que tenemos dos clases, la distancia máxima de dos puntos de una clase será a lo sumo $d_1 = 4,95/2 = 2,475 = 6\sigma_1$, con lo que $\sigma_1^2 = 0,17$.
- Calcular un valor para σ_2 . Procediendo como en el caso anterior, hacemos que la distancia a la que se encuentran la mayoría de los puntos de la clase es la mitad de d_1 , obtenemos $\sigma_2^2 = 0,043$.

- Ajustar, mediante diferentes ejecuciones, los valores para γ_1 y γ_2 de forma que las componentes en el índice de coste de AFCRC debidas a J_{lejos} y J_{cerca} no superen el 10 % del total. Con ello se llega a $\gamma_1 = 10^{-5}$ y $\gamma_2 = 10^{-5}$.

La aplicación del algoritmo AFCRC con dos clases y los parámetros anteriores lleva, tras 22 iteraciones, a la identificación del sistema con los modelos presentados en la tabla 10.2.

Pto. operación (u, y)	Función de transferencia	Offset	Ganancia	t_e
(18.8834,33.4136)	$\frac{0,0316z^{-1}}{1-0,9445z^{-1}}$	1.0016	0.5694 °C/V	685.12 s
(67.3762,84.0316)	$\frac{0,09859z^{-1}}{1-0,9490z^{-1}}$	-2.0909	1.933 °C/V	747.33 s

Tabla 10.2: Modelos locales identificados por el algoritmo AFCRC.

Como se puede observar, estos modelos son bastante similares a los que resultarían al linealizar el sistema (10.3) de la forma sugerida en (10.4), tal y como se observa en la tabla 10.3 tanto en lo referente a polos y offset como a tiempo de establecimiento t_e y, sobre todo, a ganancia.

Pto. operación (u, y)	Función de transferencia	Offset	Ganancia	t_e
(18.8834,33.4136)	$\frac{0,0299z^{-1}}{1-0,9464z^{-1}}$	1.0575	0.5578 °C/V	710.12 s
(67.3762,84.0316)	$\frac{0,1067z^{-1}}{1-0,9464z^{-1}}$	-2.2557	1.9907 °C/V	710.12 s

Tabla 10.3: Modelos locales linealizados en los puntos de operación dados por el algoritmo AFCRC.

Para la determinación de las zona de validez de cada modelo, un indicio pueden ser las forma de las clases, indicador de la dirección de la no linealidad identificada. Para ello se puede tomar la dirección del autovector asociado al autovalor menor de la matriz de dispersión borrosa de cada clase, siempre y cuando el conjunto de datos empleado en el agrupamiento sea lo suficientemente rico, es decir, cubra todo el rango de combinaciones de entradas y salidas del sistema.

Sin embargo, el conjunto de datos empleados hasta aquí se distribuye tan solo en una zona del rango de operación posible, tal y como se observa en la figura 10.8 correspondiente a la asignación de los datos a los modelos identificados y las direcciones principales de las clases. Esto se puede corroborar calculando la dirección de la no linealidad que sugieren las matrices de dispersión borrosa asociadas a la clase de cada modelo identificado, la cual se obtiene desde las matrices de autovalores λ_i y autovectores v_i correspondientes y que, para el ejemplo, se muestran en las matrices (10.5). La primera columna de la matriz de autovectores corresponde a la dirección del primer autovalor y la segunda al segundo autovalor. La primera fila de las matrices de autovalores será la correspondiente al primer autovector y la segunda al segundo autovector. Estas direcciones y la importancia relativa de cada una de ellas se representa por las

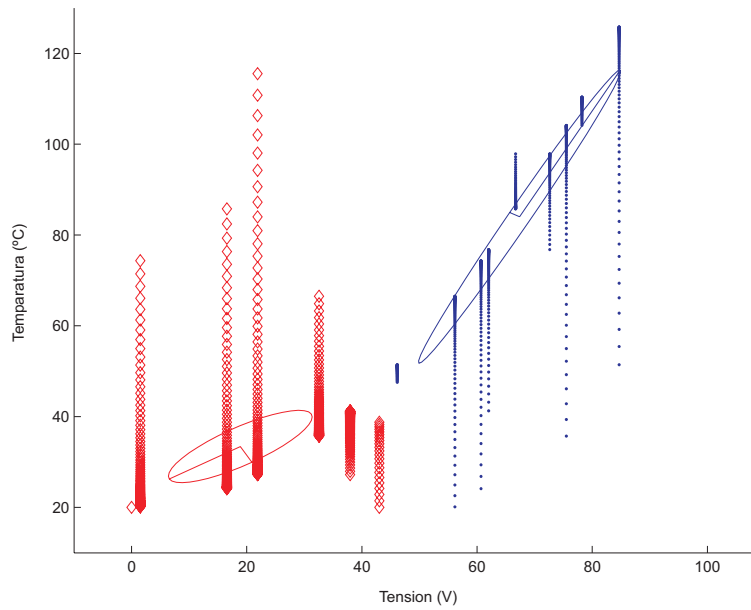


Figura 10.8: Datos asignados a la clase 1 (o) y a la clase 2 (Δ) y direcciones principales.

elipses de la figura 10.8.

$$\begin{aligned} \lambda_1 &= \begin{bmatrix} 80,41 \\ 285,28 \end{bmatrix}; & v_1 &= \begin{bmatrix} 0,5025 & -0,8646 \\ -0,8646 & -0,5025 \end{bmatrix} \\ \lambda_2 &= \begin{bmatrix} 38,93 \\ 734,31 \end{bmatrix}; & v_2 &= \begin{bmatrix} -0,8794 & 0,4761 \\ 0,4761 & 0,8794 \end{bmatrix} \end{aligned} \quad (10.5)$$

Según los datos identificados, la no linealidad del sistema se corresponde con una dirección entre $v_{1\lambda_{min}} = [0,5025 \ -0,8646]$ y $v_{2\lambda_{min}} = [-0,8794 \ 0,4761]$ (media en $v_{\lambda_{min}} = [0,6910 \ -0,6704]$). Sin embargo, dado que el modelo del sistema es conocido (10.3), se sabe que la no linealidad es debida a la tensión, con lo que el resultado del autovector asociado al autovalor menor debería ser en ambas clases $v_{min} = [1 \ 0]$.

Esto corrobora la idea de que la forma de las clases resultantes es muy dependiente del conjunto de datos empleados. Si bien este conjunto de datos sería típico en la identificación real de un proceso, por no ser el más idóneo no permite estudiar independientemente la adecuación de las técnicas de identificación propuestas de los errores conlleva no disponer de los datos necesarios. Para hacerlo posible se realizó, dada la disponibilidad del modelo del sistema, un experimento artificial consistente en una matriz de datos que cubra todo el rango de valores a estudiar para el mismo rango de tensiones y temperaturas.

Con este nuevo conjunto de datos, los parámetros adecuados para la ejecución del algoritmo AFCRC son, procediendo de igual modo que en el caso anterior:

- Los datos normalizados, tanto en tensión como en temperatura, varían en el rango $[-1,7, 1,7]$, con lo que $d_{max} = \sqrt{(1,7 + 1,7)^2 + (1,7 + 1,7)^2} = 4,80$.
- La distancia máxima de dos puntos de una misma clase será $d_1 = 4,80/2 = 2,40 = 6\sigma_1$, con lo que $\sigma_1^2 = 0,16$.

- Suponiendo que hacemos que la distancia a la que se encuentran la mayoría de los puntos de la clase es la mitad de d_1 , obtenemos $\sigma_2^2 = 0,04$.
- El ajuste, mediante diferentes ejecuciones, de los valores de γ_1 y γ_2 de forma que las componentes en el índice de coste de AFCRC debidas a J_{lejos} y J_{cerca} no superen el 10 % del total, lleva a $\gamma_1 = 2 \times 10^{-5}$ y $\gamma_2 = 10^{-4}$.

La aplicación del algoritmo AFCRC con dos clases y los parámetros anteriores lleva, tras 25 iteraciones, a la identificación del sistema con los modelos presentados en la tabla 10.4. Los modelos así identificados son muy similares a los que resultan al linealizar el

Pto. operación (u, y)	Función de transferencia	Offset	Ganancia	t_e
(27.6404, 74.2448)	$\frac{0,0519z^{-1}}{1-0,9465z^{-1}}$	0.9701	0.5694 °C/V	711.18 s
(62.3585, 75.7426)	$\frac{0,0907z^{-1}}{1-0,9463z^{-1}}$	-2.4709	1.6889 °C/V	709.13 s

Tabla 10.4: Modelos locales identificados por el algoritmo AFCRC.

sistema (10.3) como en (10.4), los cuales se pueden observar en la tabla 10.5 tanto en lo referente a polos y offset como a ganancia y, sobre todo, a tiempo de establecimiento t_e . Con este conjunto de datos, mucho más rico que el primero, la forma de las clases si

Pto. operación (u, y)	Función de transferencia	Offset	Ganancia	t_e
(27.6404, 74.2448)	$\frac{0,0327z^{-1}}{1-0,9464z^{-1}}$	1.0027	0.6099 °C/V	710.12 s
(62.3585, 75.7426)	$\frac{0,1099z^{-1}}{1-0,9464z^{-1}}$	-2.4709	2.0501 °C/V	710.12 s

Tabla 10.5: Modelos locales linealizados en los puntos de operación dados por el algoritmo AFCRC.

puede ser un buen indicativo de la dirección de la no linealidad identificada. Para ello se puede tomar la dirección del autovector asociado al autovalor menor de la matriz de dispersión borrosa de cada clase. La figura 10.9 muestra la asignación de los datos a los modelos identificados y las direcciones principales de las clases. Las matrices de autovalores λ_i y autovectores v_i correspondientes a este caso se muestran en las matrices (10.6). La primera columna de la matriz de autovectores corresponde a la dirección del primer autovalor y la segunda al segundo autovalor. Como los autovectores son prácticamente paralelos a los ejes de coordenadas, podemos decir en este caso que la primera fila de las matrices de autovalores será la correspondiente al eje de la tensión y la segunda al de la temperatura. Estas direcciones y la importancia relativa de cada una de ellas se representa por las elipses de la figura 10.9.

$$\begin{aligned} \lambda_1 &= \begin{bmatrix} 868,47 \\ 1114,97 \end{bmatrix}; & v_1 &= \begin{bmatrix} -0,9988 & -0,0500 \\ -0,0500 & 0,9988 \end{bmatrix} \\ \lambda_2 &= \begin{bmatrix} 868,62 \\ 1115,85 \end{bmatrix}; & v_2 &= \begin{bmatrix} -0,9985 & -0,0545 \\ -0,0545 & 0,9985 \end{bmatrix} \end{aligned} \quad (10.6)$$

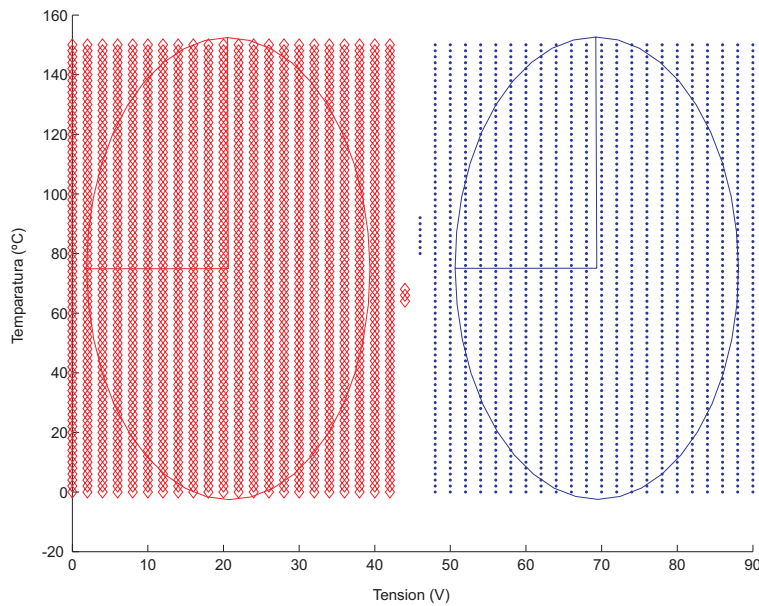


Figura 10.9: Datos asignados a la clase 1 (o) y a la clase 2 (Δ) y direcciones principales.

La dirección identificada para la no linealidad del sistema se corresponde casi perfectamente con la del eje de tensión, dado que el resultado del autovector asociado al autovalor menor en ambas clases es muy similar a $v_{min} = [1 \ 0]$.

Nota 2.1. Como se vió en el capítulo 4 desde los modelos así identificados no es posible obtener un modelo lineal de parámetros variables con garantías para realizar un posterior control, aunque si que será válida la información proporcionada en lo referente a los modelos linealizados en los puntos prototipo de las clases y la dirección de la no linealidad.

Modelo LPV por velocidad. Dado que el control se realizará, por las ventajas descritas en el capítulo 4, basado en un modelo linealizado por velocidad y que esto se convierte en el caso discreto es una linealización por incrementos del tipo $\Delta \hat{y}_k = \hat{y}_k - \hat{y}_{k-1}$, quedan para el ejemplo los modelos locales de la tabla 10.6. Como la dependencia

Pto. operación (u, y)	Modelo incremental
(27.6404, 74.2448)	$\Delta \hat{y}_{k+1} = 0,9465 \Delta \hat{y}_k + 0,0519 \Delta u_k$
(62.3585, 75.7426)	$\Delta \hat{y}_{k+1} = 0,9463 \Delta \hat{y}_k + 0,0907 \Delta u_k$

Tabla 10.6: Modelos locales por incrementos identificados por el algoritmo AFCRC.

de la no linealidad se ha mostrado clara con la tensión, una interpolación lineal de modelos locales lineales entre los valores de tensión correspondientes al par de centroides (u_1, y_1) , (u_2, y_2) será una buena opción. Esto se puede implementar de manera sencilla

redefiniendo las funciones de pertenencia de las clases como (10.7):

$$\mu_1(u) = \begin{cases} 1 & u \leq u_1 \\ \frac{1}{u_1 - u_2}u - \frac{u_2}{u_1 - u_2} & u \in [u_1, u_2] \\ 0 & u \geq u_2 \end{cases} \quad (10.7)$$

$$\mu_2(u) = 1 - \mu_1(u)$$

El modelo global no lineal (lineal de parámetros variables) del sistema será, dadas las funciones de pertenencia de (10.7) y los modelos locales por incrementos de la tabla 10.6:

$$\Delta \hat{y}_{k+1} = \mu_1(u)(0,9465\Delta \hat{y}_k + 0,0519\Delta u_k) + \mu_2(u)(0,9463\Delta \hat{y}_k + 0,0907\Delta u_k) \quad (10.8)$$

Nota 2.2. En el caso de no aparecer, como en el ejemplo, una no linealidad tan marcada en la dirección de una de las variables, se podría interpolar de igual modo en la dirección de la combinación de variables que presenten la no linealidad (interpolación de las direcciones de los autovectores asociados a los menores autovalores). Evidentemente, siempre existe la opción de emplear técnicas basadas en la información que proporciona la forma de las funciones de pertenencia identificadas (apartado 6.7.5), como se ha mostrado en el ejemplo del apartado 10.1.2.

Identificación desde datos reales

Para el estudio del sistema real se ha tenido acceso al típico ensayo basado en introducir a la entrada un conjunto de escalones de amplitud variable aleatoriamente en el rango de interés y con una duración tal que permita al sistema alcanzar su régimen permanente ante cada uno de los escalones. Los datos empleados en este caso se pueden observar en la figura 10.10. Con este nuevo conjunto de datos, los parámetros adecuados para la ejecución del algoritmo AFCRC son, procediendo del modo ya descrito en los casos anteriores:

- Los datos normalizados, tanto en tensión como en temperatura, varían en el rango $[-2, 2]$, con lo que $d_{max} = \sqrt{(2+2)^2 + (2+2)^2} = 5,66$.
- La distancia máxima de dos puntos de una misma clase será $d_1 = 5,66/2 = 2,83 = 6\sigma_1$, con lo que $\sigma_1^2 = 0,22$.
- Suponiendo que hacemos que la distancia a la que se encuentran la mayoría de los puntos de la clase es la mitad de d_1 , obtenemos $\sigma_2^2 = 0,056$.
- El ajuste, mediante diferentes ejecuciones, de los valores de γ_1 y γ_2 de forma que las componentes en el índice de coste de AFCRC debidas a J_{lejos} y J_{cerca} no superen el 10% del total, lleva a $\gamma_1 = 10^{-5}$ y $\gamma_2 = 10^{-5}$.

Tras 26 iteraciones del algoritmo AFCRC, con dos clases y los parámetros anteriores, lleva a la identificación del sistema con los modelos presentados en la tabla 10.7. En el

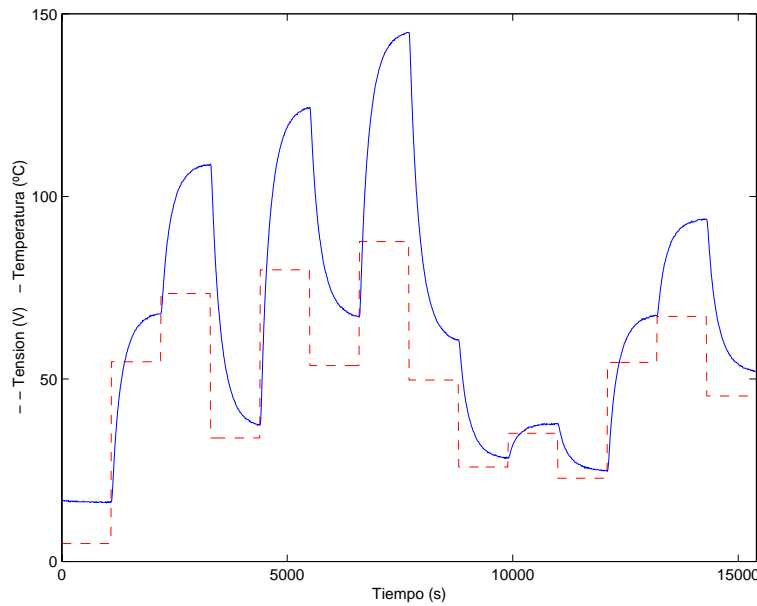


Figura 10.10: Entrada y salida del ensayo para identificación del modelo LPV.

Pto. operación (u, y)	Función de transferencia	Offset	Ganancia	t_e
(25.1154, 32.0401)	$\frac{0,0218z^{-1}}{1-0,9741z^{-1}}$	0.1981	0.8423 °C/V	1491.91 s
(64.1613, 88.0410)	$\frac{0,0961z^{-1}}{1-0,9563z^{-1}}$	-2.1926	2.20 °C/V	875.91 s

Tabla 10.7: Modelos locales identificados por el algoritmo AFCRC.

caso de identificación desde datos reales no hay modelo con el que comparar, aunque si que se puede observar en la tabla 10.8 que las ganancias estimadas por la interpolación de modelos identificados son muy similares a los del sistema real aunque los tiempos de establecimiento t_e difieren bastante. Esta discrepancia en los valores de t_e es algo que cabía esperar dadas la características de la señal empleada en el ensayo: componente frecuencial poco rica. De hecho, esto ocurrió ya en la primera identificación realizada desde una señal similar pero generada por el modelo simulado (visto en la sección 10.2.2) y, en el caso que ahora se analiza, los resultados pueden incluso verse algo empeorados por la existencia de ruido.

Pto. operación (u)	Ganancia real	t_e real	Ganancia modelo	t_e modelo
30.5	1.02 °C/V	706 s	1.03 °C/V	1405.67 s
38.65	1.34 °C/V	688 s	1.31 °C/V	1276.31 s
57.16	1.88 °C/V	669 s	1.95 °C/V	986.79 s

Tabla 10.8: Datos reales en los puntos de operación (tensiones) dados por el algoritmo AFCRC.

Con el conjunto de datos disponible, la forma de las clases (de nuevo) no es un buen indicativo de la dirección de la no linealidad identificada. La figura 10.11 muestra la

asignación de los datos a los modelos identificados y las direcciones principales de las clases.

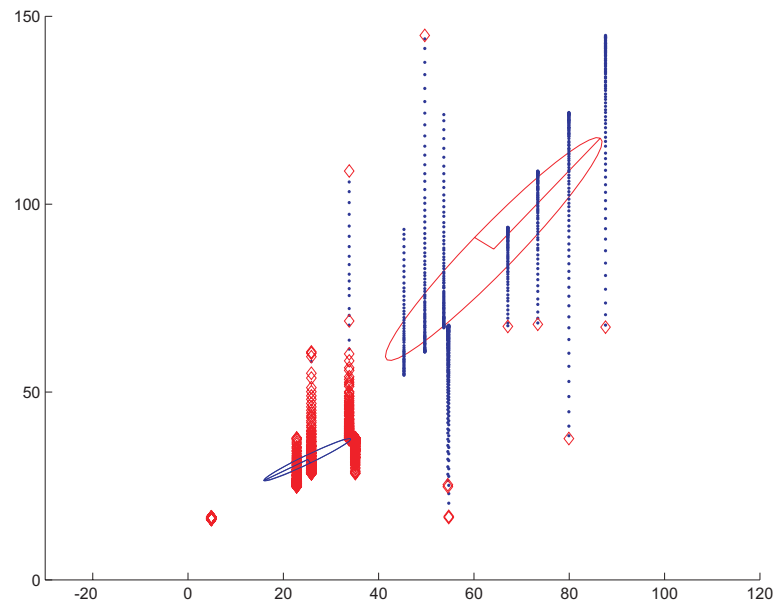


Figura 10.11: Datos asignados a la clase 1 (o) y a la clase 2 (Δ) y direcciones principales.

Las matrices de autovalores λ_i y autovectores v_i correspondientes a este caso se muestran en las matrices (10.9). La primera fila de las matrices de autovalores será la correspondiente al eje de la tensión y la segunda al de la temperatura. La primera columna de la matriz de autovectores corresponde a la dirección del primer autovalor y la segunda al segundo autovalor. Estas direcciones y la importancia relativa de cada una de ellas se representa por las elipses de la figura 10.11.

$$\begin{aligned} \lambda_1 &= \begin{bmatrix} 21,94 \\ 216,27 \end{bmatrix}; & v_1 &= \begin{bmatrix} 0,5236 & -0,8520 \\ -0,8520 & -0,5236 \end{bmatrix} \\ \lambda_2 &= \begin{bmatrix} 101,50 \\ 740,74 \end{bmatrix}; & v_2 &= \begin{bmatrix} -0,7971 & 0,6039 \\ 0,6039 & 0,7971 \end{bmatrix} \end{aligned} \quad (10.9)$$

Modelo LPV por velocidad. El modelo linealizado por velocidad, linealización por incrementos $\Delta\hat{y}_k = \hat{y}_k - \hat{y}_{k-1}$ en el caso discreto, que se empleará para el control lleva en este caso a los modelos locales de la tabla 10.9. Como los datos no cubrían todo el

Pto. operación (u, y)	Modelo incremental
(25.1154, 32.0401)	$\Delta\hat{y}_{k+1} = 0,9741\Delta\hat{y}_k + 0,0218\Delta u_k$
(64.1613, 88.0410)	$\Delta\hat{y}_{k+1} = 0,9563\Delta\hat{y}_k + 0,0961\Delta u_k$

Tabla 10.9: Modelos locales por incrementos identificados por el algoritmo AFCRC.

posible rango de operación del sistema, una interpolación lineal en la dirección sugerida

por los autovalores y autovectores de (10.9) como en (10.7) y (10.8) puede no ser una buena idea. En casos como este se sugiere realizar un mejor ensayo experimental o, si no es posible, emplear las técnicas basadas en las funciones de pertenencia identificadas del apartado 6.7.5. En este caso se ha optado por la aproximación dada por la inversión dada en el algoritmo FCM, que para este ejemplo se puede observar en la figura 10.12.

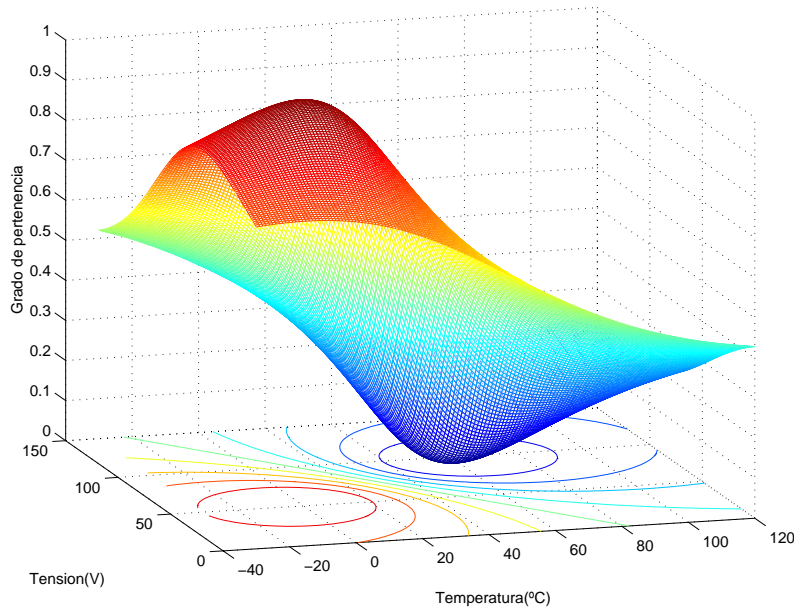


Figura 10.12: Función de pertenencia μ_1 estimada con (C.18).

De este modo se dispondrá de unas funciones de pertenencia μ_1 y μ_2 que, en combinación con los modelos locales de la tabla 10.9, llevarán al modelo lineal de parámetros variables del sistema (10.10).

$$\begin{aligned} \Delta \hat{y}_{k+1} = & \mu_1(0,9741\Delta \hat{y}_k + 0,0218\Delta u_k) + \\ & \mu_2(0,9563\Delta \hat{y}_k + 0,0961\Delta u_k) \end{aligned} \quad (10.10)$$

10.2.3. Diseño de controladores lineales

Para el diseño de los controladores, dada la desaparición del offset en los modelos incrementales (10.8) y (10.10), se pueden usar sin ningún problema las técnicas tradicionales para el control de sistemas lineales. El criterio de diseño de los controladores será, por ejemplo, asignar un polo en 0.85, lo que se puede hacer resolviendo la ecuación (10.11),

$$\Delta \hat{y}_{k+1} - \Delta r_k = 0,85(\Delta \hat{y}_k - \Delta r_k) \quad (10.11)$$

donde r es la referencia de temperatura a seguir por el horno.

Como se comentó en el apartado 9.2.3, la resolución de ecuaciones como (10.11) resulta en un controlador basado tan solo en las variaciones incrementales de las variables, lo que puede llevar a una situación en la que un error inicial en el seguimiento de una referencia constante no sea nunca subsanado. Este error puede ser evitado con la linealización por incrementos parcial propuesta en 9.24 y que en este caso, partiendo de

la especificación (10.12) y restando y_k a ambos lados, se obtiene una expresión (10.13) equivalente a (10.11) en la que se explicita el error de seguimiento de la referencia.

$$\hat{y}_{k+1} - r_k = 0,85(\hat{y}_k - r_k) \quad (10.12)$$

$$\begin{aligned} \hat{y}_{k+1} - \hat{y}_k - r_k &= 0,85(\hat{y}_k - r_k) - \hat{y}_k \\ &\Downarrow \\ \Delta\hat{y}_{k+1} &= 0,85(\hat{y}_k - r_k) - \hat{y}_k + r_k \\ &\Downarrow \\ \Delta\hat{y}_{k+1} &= (1 - 0,85)(r_k - \hat{y}_k) \end{aligned} \quad (10.13)$$

Por tanto, dado un modelo incremental genérico de la forma $\Delta\hat{y}_{k+1} = A\Delta u_k + B\Delta\hat{y}_k$, como los identificados en el proceso de agrupamiento y que aparecen en las tablas 10.6 y 10.9, la acción de control se podrá calcular para cada controlador i despejando como en (10.14)

$$\begin{aligned} A_i\Delta u_{k i} + B_i\Delta\hat{y}_k &= (1 - 0,85)(r_k - \hat{y}_k) \\ &\Downarrow \\ \Delta u_{k i} &= \frac{1 - 0,85}{A_i}(r_k - \hat{y}_k) - \frac{B_i}{A_i}\Delta\hat{y}_k \end{aligned} \quad (10.14)$$

10.2.4. Control por planificación de ganancia: interpolación de controladores

El último paso para concluir con el diseño de un controlador no lineal por planificación de ganancia para el sistema térmico identificado, consiste en realizar una apropiada interpolación de los mismos. Las opciones para la interpolación de los controladores son múltiples, tal y como comentó en profundidad en el capítulo 4. Dado un controlador por velocidad, la propuesta que se hace es seguir el mismo criterio que se haya empleado para obtener el modelo no lineal del proceso.

Para comprobar que las técnicas propuestas funcionan correctamente se va a emplear el modelo simulado, dadas las facilidades que nos ofrece en la evaluación del rendimiento y la lentitud del sistema real.

La interpolación de los controladores (10.14) basados en los modelos de la tabla 10.6 se realizará siguiendo la misma pauta empleada para interpolar dichos modelos lineales identificados para llegar al modelo no lineal del proceso 10.8. Por tanto, se aprovecharán las funciones de pertenencia definidas en (10.7) que tenían en cuenta la detección de la dependencia de la no linealidad con la tensión, y se realizará una interpolación lineal de acciones de control entre los valores de tensión extremos que marcan los centroides.

Con todas las consideraciones anteriores, la acción de control no lineal basada en una interpolación de los i controladores (10.14) dada por las funciones de pertenencia (10.7) vendrá dada por (10.15).

$$\begin{aligned} \Delta u_k = & \mu_1 \left(\frac{1 - 0,85}{0,0519} (r_k - \hat{y}_k) - \frac{0,9465}{0,0519} \Delta \hat{y}_k \right) \\ & + \mu_2 \left(\frac{1 - 0,85}{0,0907} (r_k - \hat{y}_k) - \frac{0,9463}{0,0907} \Delta \hat{y}_k \right) \end{aligned} \quad (10.15)$$

10.2.5. Rendimiento

Como se ha comentado en repetidas ocasiones a lo largo del capítulo anterior, lo habitual es que la evaluación del rendimiento del controlador por planificación de ganancia esté basada en simulaciones. En este apartado se van a mostrar los resultados de simulación del controlador diseñado en diferentes puntos de funcionamiento, todos ellos en el rango de temperaturas y tensiones para las que fue ensayado el sistema.

Para cada los puntos de funcionamiento extremos se mostrarán figuras (figura 10.13 para temperaturas bajas y figura 10.14 para temperaturas altas) compuestas por cuatro gráficas, todas ellas mostrando la referencia a seguir (pequeño escalón en el entorno del punto de funcionamiento) en rojo, y el control realizado por un controlador distinto:

- en la primera gráfica se mostrará el control realizado por el controlador Δu_{k_1} diseñado en (u_1, y_1) ,
- en la segunda gráfica se podrá ver el seguimiento de trayectoria que proporciona el controlador Δu_{k_2} diseñado en (u_2, y_2) ,
- en la tercera gráfica quedará reflejado el comportamiento del controlador no lineal diseñado por planificación de ganancia por velocidad Δu_k (interpolación de los dos anteriores), y
- en la cuarta y última gráfica aparecerá el funcionamiento deseado del controlador.

Las tres primeras gráficas, con el fin de facilitar la comparación, mostrarán también el comportamiento deseado en línea discontinua.

El comportamiento de los controladores ante un cambio en la referencia de $+1^\circ\text{C}$ en el entorno de un punto de funcionamiento a temperaturas bajas, se puede observar en la figura 10.13. Aunque todos los controladores tienen un funcionamiento aceptable en esta zona de trabajo, el mejor funcionamiento se obtiene con el primer controlador y el controlador por velocidad, ya que el segundo presenta una pequeña sobreoscilación y los otros dos siguen perfectamente el comportamiento deseado.

El comportamiento del segundo controlador va mejorando al aumentar la temperatura y el primer controlador presenta oscilaciones cada vez mayores que, como muestra la figura 10.14, hacen que se inestabilice para temperaturas altas. El comportamiento del controlador por velocidad es idéntico al deseado en todo el rango de operación, como se muestra en la figura 10.15, siendo por tanto su comportamiento el esperado y superando al controlador por velocidad tradicional como muestra la figura.

El controlador por planificación de ganancia diseñado iguala o mejora los controladores diseñados para todo el rango de funcionamiento del sistema, quedando así patente la conveniencia y aplicabilidad de las técnicas de identificación y diseño de controladores propuestas.

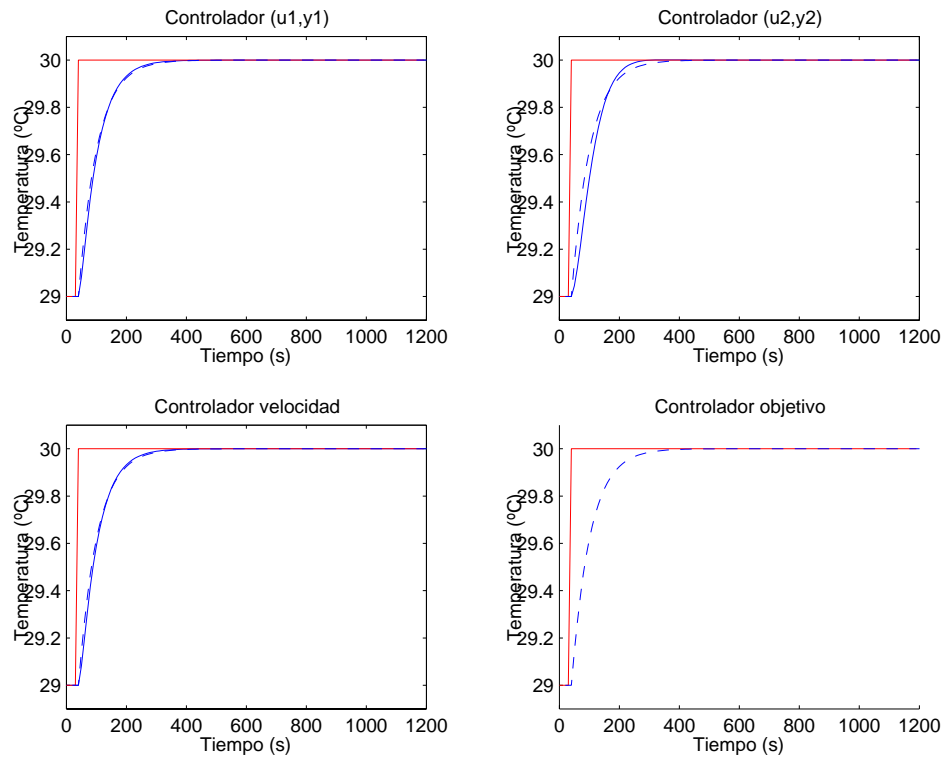


Figura 10.13: Comportamiento de los controladores en el entorno de un punto de funcionamiento a temperaturas bajas.

10.3. Conclusiones

Como se ha podido comprobar, los métodos propuestos a lo largo de esta Tesis permiten identificar y controlar de forma adecuada y sencilla un sistema no lineal, tomando como única información un conjunto de datos experimentales del sistema en estudio.

La aportación de este capítulo ha consistido en la ejemplificación, mediante el control de un sistema térmico, del empleo de todas las técnicas propuestas a lo largo del documento.

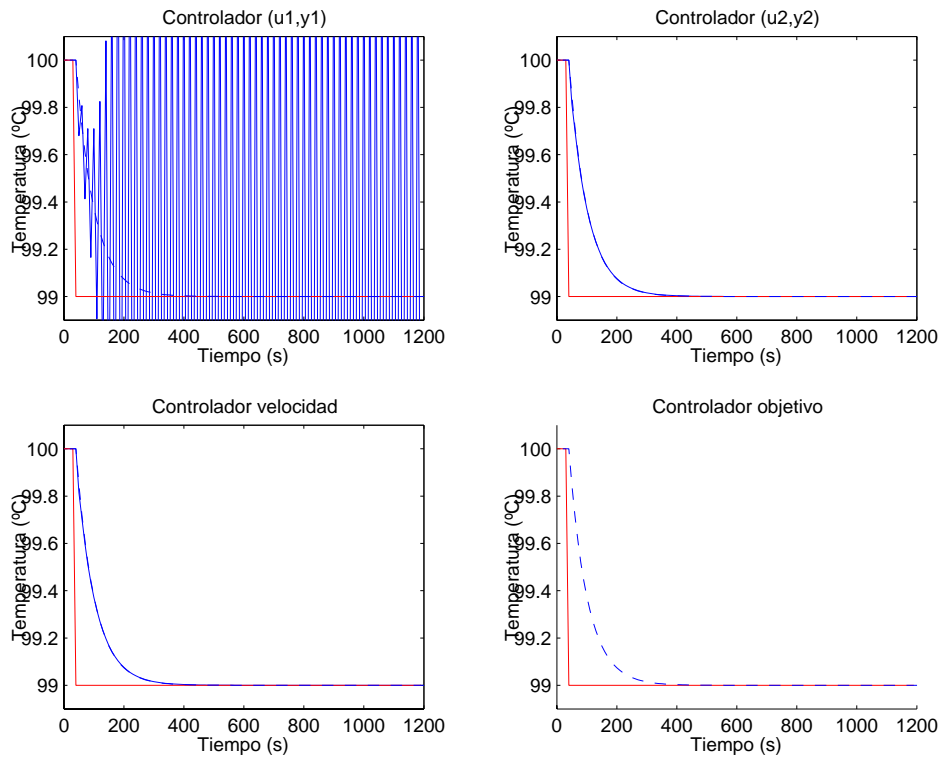


Figura 10.14: Comportamiento de los controladores en el entorno de un punto de funcionamiento a temperatura elevada.

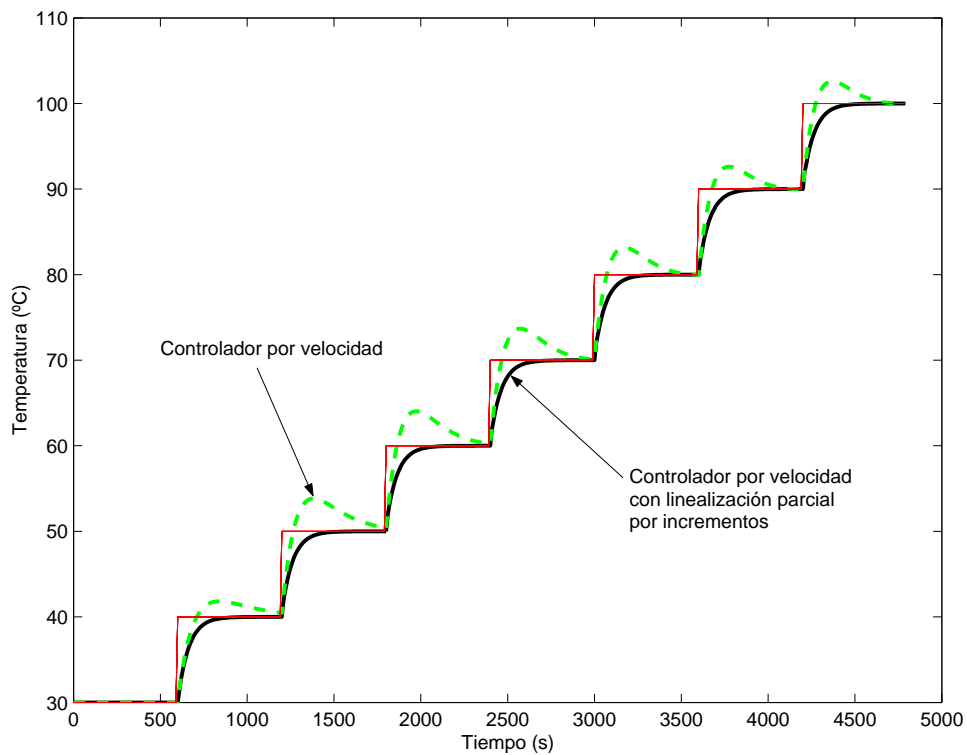


Figura 10.15: Comportamiento del controlador en el rango de funcionamiento.

Parte IV
Conclusiones

Capítulo 11

Conclusiones y cuestiones abiertas

*Me interesa el futuro
porque es el sitio donde voy
a pasar el resto de mi vida.*
Woody Allen.

Aquí, donde se cierra la Tesis, se resumirán las conclusiones que se han obtenido en el desarrollo del trabajo, resaltando las diferentes aportaciones realizadas. El capítulo concluirá con un apartado de cuestiones abiertas, donde se concretan una serie de líneas de trabajo que sugiere la línea de investigación de la Tesis.

11.1. Conclusiones de la Tesis

En esta Tesis se ha propuesto un nuevo método de identificación de modelos borrosos TS discretos de sistemas no lineales mediante la aplicación de técnicas de agrupamiento borroso, el cual resulta especialmente apropiado para una posterior aplicación de los modelos al control clásico de sistemas no lineales por planificación de ganancia, permitiendo además emplear técnicas como las LMIs.

Para ello, tras revisar el modelado de sistemas mediante las teorías del modelado por modelos locales y los modelos borrosos basados en reglas (representantes de los métodos inteligentes) observando la estrecha relación entre ellos, se ha centrado el trabajo en la identificación de estos modelos desde datos experimentales con la técnica de identificación borrosa que toma formas de trabajo de los campos de reconocimiento de formas: los algoritmos de agrupamiento o clustering.

En cuanto a la revisión de las técnicas de modelado por modelos locales, vistas las diferentes opciones para realizar particiones de modelos locales de sistemas no lineales mediante linealización en el entorno de puntos de operación empleando el Jacobiano de la planta, las opciones que se han mostrado más interesantes han sido las relacionadas con familias de linealizaciones en torno a puntos de operación. Esto es debido a que cuando los puntos de operación se corresponden con puntos de equilibrio cualquier análisis o diseño basado en esos modelos solo será válido cuando se opere cerca del equilibrio. Sin embargo, las linealizaciones en puntos de operación o la linealización basada en la velocidad permiten la linealización en torno a puntos que no son de equilibrio.

Posteriormente se han presentado las diferentes alternativas de control que son aplicables a un conjunto de modelos lineales locales de un sistema no lineal: las técnicas basadas en LMIs (en unión con las LDIs) y las de planificación de ganancia para familias de modelos LPV.

En el modelado borroso, los dos tipos de modelos basados en reglas (lingüísticos y de Takagi-Sugeno) pueden ser usados para modelar sistemas estáticos y dinámicos. Sin embargo, el modelo de Takagi-Sugeno que proporciona un posible marco para la descomposición de un sistema no lineal en modelos lineales localmente válidos (ya que tiene funciones reales en su consecuente), no es apropiado directamente para aplicar técnicas de control clásicas, por lo que el trabajo se ha centrado en estudiar modificaciones a este tipo de modelos para hacerlos adecuados.

La revisión del estado del arte ha concluido con las técnicas de agrupamiento borroso, dada su idoneidad para la aproximación de sistemas no lineales complejos con modelos borrosos desde datos numéricos. La aplicación de las técnicas de agrupamiento borroso en la determinación del modelo borroso es muy apropiada por la similitud entre las clases formadas en el proceso de agrupamiento y las funciones de pertenencia necesarias para la definición de los elementos de antecedentes y consecuentes de las reglas. En esta Tesis se ha buscado emplear los métodos de agrupamiento para identificar modelos borrosos con consecuentes afines tipo TS que integrasen modelos LPV consistentes en familias de modelos locales, por lo que se ha centrado la revisión en aquellos algoritmos de agrupamiento con capacidades de detección de subespacios lineales.

Tras las revisiones, la primera aportación de la Tesis ha sido la propuesta de un **método sistematizado para la identificación de modelos borrosos para control** en el que se integran las técnicas revisadas. Para ello se han seguido los pasos necesarios para la construcción de un modelo borroso TS de un sistema no lineal mediante técnicas de agrupamiento y en cada uno de ellos se han comentado los métodos existentes en la bibliografía que permiten llevarlos a cabo, proponiendo una técnica concreta para cada uno de ellos y llegando a un proceso sistematizado y automatizado de identificación. Sin embargo, a pesar de que el método queda validado desde el punto de vista de error cuadrático medio de la salida estimada, con la perspectiva de la aplicación a control se han detectado algunos problemas: efecto del ruido en el proceso de identificación, interpretabilidad de las funciones de pertenencia identificadas y posible no correspondencia de los modelos locales identificados con linealizaciones en el entorno de los prototipos.

Por todo lo anterior, se ha llevado a cabo un **estudio crítico de los algoritmos de agrupamiento en su aplicación a la identificación de modelos borrosos para control** tanto desde un punto de vista cualitativo como cuantitativo (análisis para el cual se han definido unos **nuevos índices de validez de los modelos borrosos identificados**), lo que ha demostrado que ninguno de los algoritmos de la literatura resuelve el problema completamente. El problema fundamental estriba en que en la identificación con técnicas de inteligencia artificial también deben ser aplicados el concepto de **modelado para control frente al de modelado para predicción**, siendo en el primer caso importante el compromiso entre interpretabilidad y error de modelado. Además, con el objetivo de la aplicación al diseño de controladores un criterio de modelado local parece en principio más adecuado y no siempre se aplica.

El objetivo de la Tesis se ha concretado entonces en encontrar un algoritmo de

agrupamiento que cumplierse con las **características deseables de un algoritmo de agrupamiento específico para la identificación de modelos borrosos para control**:

1. Los modelos locales lineales se obtuviesen en el proceso de agrupamiento.
2. Se mejorase la interpretabilidad del modelo:
 - a) incluyendo información sobre el error de modelado local en las iteraciones del algoritmo,
 - b) favoreciendo la convexidad de las funciones de pertenencia.
3. Se pudiese establecer un balance entre interpretabilidad y error de modelado.

Para ello se ha propuesto una **nueva función objetivo** basada en la del algoritmo AFRC que incluye en el proceso de agrupamiento condiciones que favorecen la interpretabilidad (convexidad) y procura la posibilidad establecer un balance entre interpretabilidad y error de modelado.

La minimización de la función objetivo propuesta se ha realizado en forma de algoritmo de agrupamiento, lo que ha llevado a definir una **nueva familia de algoritmos de agrupamiento (AFCRC)** adecuada para el modelado de sistemas no lineales con modelos borrosos TS para control: se consigue la convexidad de las funciones de pertenencia borrosas suministradas en el proceso de identificación así como la correspondencia de los modelos locales con linealizaciones en el entorno de los prototipos. De esta forma se han mejorado las técnicas de agrupamiento existentes en el sentido de la interpretabilidad y orientación a control, mientras que las características positivas del agrupamiento (adición de clases, fusión de clases, etc) siguen siendo posibles.

El nuevo algoritmo se ha aplicado a numerosos ejemplos con resultados muy interesantes, quedando así mostrada su aplicabilidad. El único problema del algoritmo está en la determinación y ajuste de los nuevos parámetros que se han definido en su función de coste, para lo cual se han propuesto reglas intuitivas que ayudarán al usuario en su determinación.

Dado que los modelos identificados con la metodología propuesta permiten la interpretación como modelos locales, en la parte final de la Tesis se han presentado las diferentes alternativas de control que son aplicables a un conjunto de modelos lineales locales de un sistema no lineal. Estas técnicas son las de planificación de ganancia para familias de modelos LPV (lineales con parámetros variables) y son especialmente aplicables, por sus diferentes ventajas, los controladores basados en modelos LPV en puntos de operación y los relacionados con la linealización por velocidad cuyo funcionamiento se ha validado con ejemplos sencillos. Finalmente se ha realizado la **comprobación experimental de su aplicabilidad**, mediante la aplicación al modelado de un sensor y un sistema no-lineal térmico sencillo, de que los métodos propuestos a lo largo de esta Tesis permiten identificar y controlar de forma adecuada y sencilla un sistema no lineal real, tomando como única información un conjunto de datos experimentales del sistema en estudio.

Las aportaciones de la Tesis podrían resumirse en:

- Propuesta de un método sistematizado de identificación basado en técnicas de agrupamiento y determinación de su aplicabilidad,
- Análisis general de la idoneidad de las funciones de coste de los algoritmos de identificación para control,
- Determinación de la función de coste adecuada en los algoritmos de agrupamiento en su aplicación a la determinación de modelos locales,
- Definición de unos índices para la evaluación objetiva del comportamiento de los algoritmos de agrupamiento,
- Aplicación de los índices anteriores en el análisis cuantitativo de algoritmos de agrupamiento,
- Análisis cualitativo de los algoritmos de agrupamiento existentes en la identificación de modelos locales para control,
- Propuesta de algoritmo de agrupamiento a emplear y modificaciones a realizar sobre el mismo para obtener un algoritmo adecuado en el marco de la Tesis,
- Propuesta de un índice de coste que reúne las características deseables en la identificación para control determinadas anteriormente,
- Minimización de la función de coste en la forma de un algoritmo de agrupamiento (AFCRC), manteniendo así las ventajas de esta forma de trabajo,
- Generación de una nueva familia de algoritmos de agrupamiento desde la función de coste, mostrando como ejemplo la versión posibilista del mismo (PAFCRC),
- Determinación de una estructura de trabajo para la determinación de los valores de los parámetros que aparecen en AFCRC a definir por el usuario,
- Aplicación del algoritmo en diferentes ejemplos, en comparación con algoritmos existentes, dando resultados apropiados,
- Análisis de la aplicabilidad de las técnicas de planificación de ganancia a los modelos identificados con el algoritmo AFCRC,
- Aplicación de controladores basados en puntos de operación, determinando los problemas generados por el término independiente de los modelos y la interpolación de éstos, proponiendo:
 - estructuras que minimizan el efecto del offset (basadas en integradores), y
 - cotas al error debido a la interpolación (que se hace pequeño con variaciones lentas en la referencia y funciones de pertenencia suaves con núcleos amplios), ya que las funciones de interpolación apropiadas no se corresponden con las funciones de pertenencia identificadas,

- Aplicación de los controladores por velocidad (en los que no hay problemas con los términos independientes), los cuales se ha manifestado que permiten (bajo condiciones de discretización apropiada) la interpolación con las funciones de pertenencia identificadas. El problema de estos controladores en cuanto al uso de variables incrementales y la imposibilidad de subsanación de errores constantes (excepto con integradores) se puede solventar con estructuras de linealización parcial por incrementos,
- Aplicación de las técnicas de LMIs al diseño y análisis de estabilidad de controladores en diferentes puntos de operación,
- Ejemplificación, mediante el modelado de un sensor y la identificación y control de un sistema térmico, del empleo de todas la técnicas propuestas a lo largo del documento.

Finalmente conviene resaltar que, aunque se ha optado en esta Tesis por los modelos borrosos, las ideas y técnicas propuestas son aplicables a cualquier enfoque de modelado por modelos locales.

11.2. Cuestiones abiertas

La discusión sobre la idoneidad, en lo que respecta a la aplicación al diseño de controladores e interpretabilidad para el usuario, de los diferentes criterios de modelado que se emplean para el modelado de sistemas no lineales mediante técnicas de agrupamiento borroso ha llevado a tratar de aclarar algunas ideas en el campo del modelado borroso para control, pero la línea de investigación que esta Tesis abre es muy amplia y algunas de las posibles líneas a seguir se concretan en una serie de cuestiones abiertas entre las que cabría destacar:

- Estudio del diseño de experimentos en la identificación de modelos borrosos para control. Este paso inicial tan importante en cualquier método de identificación, como se ha demostrado en el ejemplo real con el que se ha cerrado la Tesis, consistiría en definir las señales más apropiadas para la identificación de modelos borrosos de sistemas no lineales, ya que la elección de la señal de excitación puede depender del problema concreto y esto debe ser objeto de una investigación adicional.
- Análisis de alternativas para la integración de modelos locales. El proceso de identificación proporciona modelos locales en el entorno de prototipos y, como se ha mostrado en algunos ejemplos, la integración de estos modelos en un único modelo borroso no lineal requiere alguna técnica de interpolación apropiada (convexa). La propuesta de los modelos borrosos TS originales no se ha mostrado válida y, aunque se han presentado propuestas para el caso monovariante, es necesario estudiar alternativas válidas para casos multivariante.
- Estudio teórico de la determinación de los parámetros de AFCRC. Aunque se han propuesto reglas intuitivas que ayudan al usuario en la determinación y ajuste

de los nuevos parámetros que se han definido en la función de coste de AFCRC, un estudio teórico que respalde la determinación de dichos parámetros sería muy conveniente.

- Aplicabilidad de la familia de algoritmos de agrupamiento AFCRC. El nuevo índice de coste propuesto genera toda una nueva familia de algoritmos de agrupamiento cuya aplicabilidad sería conveniente estudiar en profundidad.
- Análisis de la estabilidad de los modelos borrosos obtenidos. Parece sugerente el tema de garantizar de algún modo la estabilidad de modelo desarrollado o estudiar las formas en las que se podría mejorar la estabilidad del controlador final durante la etapa de identificación del sistema. Algunos trabajos teóricos ya se desarrollado en el sentido de mejorar la robustez de los controladores aprovechando la forma de las funciones de pertenencia a identificar (Díez and Previdi, 2001).
- Reducción de la dimensionalidad de los modelos. La aplicación de las técnicas de detección de redundancia y validación comentadas en la Tesis podrían dar lugar a mejoras en el algoritmo propuesto, como podría ser la creación de un algoritmo AFCRC combinado con el CCM (fusión de clases compatibles).
- Estudio comparativo del algoritmo AFCRC. Cuestiones como el tiempo de ejecución o número de iteraciones necesarias para la convergencia del algoritmo no han sido estudiadas en comparación las técnicas de agrupamiento existentes y se estima como necesario para una hipotética evaluación de la aplicabilidad en un montaje de aprendizaje dinámico de modelos (en línea).
- Aplicación a casos reales e industriales. Las pruebas realizadas con las técnicas propuesta, aunque garantizan su aplicabilidad, deberían probarse en sistemas más complejos y en entornos industriales. Se propone aplicar la técnicas expuestas en los casos reales de los proyectos de investigación en los que está involucrado el Grupo de Control de Sistemas Complejos: control de un biorreactor y control de inyección en motores diesel.

Parte V
Bibliografía

Bibliografía

- Abonyi, J., R. Babuska, L.Nagy and F. Szeifert (2000). Local and global identification for fuzzy model based control. *Proc. of Intelligent Systems in Control and Measurement Symposium* pp. 111–116.
- Ahvenlampi, T., J.L. Díez and J.L. Navarro (2003). New methods for validation of local models in fuzzy clustering identification. *Proc. of IFAC Conference on Intelligent Control and Signal Processing* pp. 9–12.
- Albertos, P. and A. Sala (2002). *Iterative identification and control*. Springer-Verlag. London.
- Apkarian, P. and P. Gahinet (1995). A convex characterisation of gain-scheduled h-infinity controllers. *IEEE Transactions on Automatic Control* **40**, 853–864.
- Aström, K.J. and B. Wittenmark (1995). *Adaptive Control*. Addison-Wesley. Reading, Massachusetts.
- Babuska, R. (1996). *Fuzzy Modeling and Identification*. PhD dissertation, Delft University of Technology. Delf, The Netherlands.
- Bezdek, J.C. (1973). *Fuzzy Mathematics in Pattern Classification*. PhD Thesis, Applied Math. Center, Cornell University. Ithaca.
- Bezdek, J.C. (1987). *Pattern recognition with Fuzzy Objective Function Algorithms*. Plenum Press. New York.
- Bezdek, J.C. and J.C. Dunn (1975). Optimal fuzzy partition: a heuristic for estimating the parameters in a mixture of normal distributions. *IEEE Transactions on Computers* **24**, 835–838.
- Bezdek, J.C. and N.R. Pal (1998). Some new indexes of cluster validity. *IEEE Transactions on System, Man and Cybernetics-Part B: Cybernetics* **28**, 301–315.
- Blasco, X., J.M. Herrero, M. Martinez and J. Senent (2001). Nonlinear parametric model identification with genetic algorithms. application to a thermal process. In: *Connectionist models of neurons, learning processes and artificial intelligence. Proceedings of 6th International Conference Artificial and Neural Networks* (J. Mira and A. Prieto, Eds.). Vol. I. pp. 466–473. Springer. Granada.

- Bondía, J. (1997). *Herramientas de análisis de estabilidad de sistemas no lineales*. Internal report (in spanish), Department of Systems Engineering and Control. Polytechnic University of Valencia.
- Bonivento, C., C. Fantuzzi and R. Rovati (1998). *Fuzzy Logic Control: Advances in Methodology*. Ed. World Scientific. Singapore.
- Boyd, S., L. ElGhaoui, E. Feron and V. Balakrishnan (1994). *Linear matrix inequalities in system and control theory*. SIAM. Philadelphia.
- Carbonell, P. (1999). *Técnicas de control borroso aplicadas a convertidores dc/dc conmutados. Aplicación al convertidor buck*. PhD Thesis (in spanish), Department of Systems Engineering and Control. Polytechnic University of Valencia.
- Carbonell, P., J.L. Díez and J.L. Navarro (2000). Application of local model techniques in complex systems. *Journal of Spanish Association for Artificial Intelligence* **10**, 111–118.
- Cellier, F.E. (1991). *Continuous system modeling*. Springer. New York.
- Chan, L.A., N.M. Nasrabadi and V. Mirelli (1996). Multi-stage target recognition using modular vector quantizers and multilayer perceptrons. *Proc. IEEE Conference in Computer Vision and Pattern Recognition* **1**, 114–119.
- Davé, R.N. (1990). An adaptive fuzzy c-elliptotype clustering algorithm. *Proc. NAFIPS 90: Quarter Century of Fuzziness* **1**, 9–12.
- Davé, R.N. (1991). Characterization and detection of noise clustering. *Pattern Recognition Letters* **12**, 657–664.
- Davé, R.N. and R. Krishnapuram (1997). Robust clustering methods: a unified view. *Transactions on Fuzzy Systems* **5**, 270–293.
- Delgado, M., A. F. Gómez-Skarmeta and F. Martín (1997). A fuzzy clustering-based rapid prototyping for fuzzy rule-based modelling. *Transactions on Fuzzy Systems* **5**, 223–233.
- Desoer, C.A. (1969). Slowly varying controller $dx/dt=a(t)x$. *IEEE Transactions on Automatic Control* **14**, 780–781.
- Díez, J. L., A. Sala and J. L. Navarro (2002a). Fuzzy clustering algorithm for local model control. *Proc. IFAC 15th World Congress* pp. 60–66.
- Díez, J. L. and F. Previdi (2001). Guaranteed closed loop precision in multiple model based control. *Proc. IFAC Workshop on Adaptation and Learning in Control and Signal Processing* pp. 139–144.
- Díez, J. L., J. L. Navarro and A. Sala (2001). Identification for local model control with fuzzy clustering. *Proc. IFAC Workshop on Advanced Fuzzy-Neural Control* pp. 99–104.

- Díez, J. L., J. L. Navarro and A. Sala (2002b). Algoritmos de clustering en la identificación de modelos borrosos. *Proc. Jornadas de Automática* pp. 100–106.
- Díez, J.L. (1999). *Identificación automática de modelos borrosos mediante técnicas de clustering*. Internal report (in spanish), Department of Systems Engineering and Control. Polytechnic University of Valencia.
- Díez, J.L. and J.L. Navarro (1999). Fuzzy models of complex systems by means of clustering techniques. *Proc. 2nd Intelligent Systems in Control and Measurement* pp. 147–153.
- Dickerson, J.A. and B. Kosko (1996). Fuzzy function approximation with ellipsoidal rules. *IEEE Transactions on Systems, Man and Cybernetics - Part B: Cybernetics* **26**, 542–560.
- Duda, R. O. and P. E. Hart (1973). *Pattern Classification and Scene Analysis*. John Wiley & Sons. New York.
- Duda, R. O., P. E. Hart and D. G. Stork (2000). *Pattern Classification*. John Wiley & Sons. New York.
- Dzielinski, A. and W. Graniszewski (2002). Nonlinear control systems modelling: Global vs local. *Proc. of the American Control Conference* pp. 537–542.
- Emami, M. R., I.B. Türksen and A.A. Goldenberg (1998). Development of a systematic methodology of fuzzy logic modeling. *Transactions on Fuzzy Systems* **6**, 346–366.
- Gath, I. and A.B. Geva (1989). Unsupervised optimal fuzzy clustering. *IEEE Transactions on Pattern Analysis and Machine Intelligence* **11**, 773–781.
- Genther, H. and M. Glesner (1997). Advanced data preprocessing using fuzzy clustering techniques. *Fuzzy Sets and Systems* **85**, 155–164.
- Godfrey, K. (1990). *Perturbation signals for system identification*. Prentice Hall. New Jersey.
- Gustafson, E.E. and W.C. Kessel (1979). Fuzzy clustering with a fuzzy covariance matrix. *Proc. of IEEE CDC, San Diego, California* pp. 761–766.
- Haber, R. and H. Unbehauen (1990). Structure identification of nonlinear dynamic systems. a survey on input-output approaches. *Automatica* **26**, 651–677.
- Hathaway, R. J. and J.C. Bezdek (1991). Grouped coordinate minimization using newton's methods for inexact minimization in one vector coordinate. *Journal of Optimization Theory and Applications* **71**, 503–516.
- Hathaway, R. J. and J.C. Bezdek (1993). Switching regression models and fuzzy clustering. *Transactions on Fuzzy Systems* **1**, 195–204.
- Hirota, K. and W. Pedrycz (1996). Directional fuzzy clustering and its application to fuzzy modelling. *Fuzzy Sets and Systems* **80**, 315–326.

- Hong, T.-P. and C.-Y. Lee (1996). Induction of fuzzy rules and membership functions from training examples. *Fuzzy Sets and Systems* **84**, 33–47.
- Hopf, J. and F. Klawonn (1994). Learning the rule base of a fuzzy controller by a genetic algorithm. <http://info.et-inf.fho-empden.de/dmlab/index.html>.
- Hoppensteadt, F.C. (1966). Singular perturbations on the infinite interval. *Transactions of the American Mathematical Society* **123**, 521–535.
- Hunt, K.J. and T.A. Johansen (1997). Design and analysis of gain-scheduled control using local controller networks. *International Journal of Control* **66**, 619–651.
- Hunt, K.J., R. Haas and R. Murray-Smith (1996). Extending the functional equivalence of radial basis function networks and fuzzy inference systems. *IEEE Transactions on Neural Networks* **7**, 776–781.
- Hwang, H.-S. and K.-B. Wooc (1995). Linguistic fuzzy model identification. *IEEE Proc. Control Theory and Applications* **142**, 537–544.
- Hyde, R.A. and K. Glover (1993). The application of scheduled h-infinity controllers to a vstol aircraft. *IEEE Transactions on Automatic Control* **38(7)**, 1021–1039.
- Ilchmann, A., D.H. Owens and Pratzel-Wolters D (1987). Sufficient conditions for stability of linear time-varying systems. *Systems & Control Letters* **9**, 157–163.
- Imai, H., A. Tanaka and M. Miyakoshi (1998). A method of identifying influential data in fuzzy clustering. *IEEE Transactions on Fuzzy Systems* **6**, 90–101.
- Isermann, R., K.-H. Lachmann and D. Matko (1992). *Adaptive control systems*. Prentice Hall. New York.
- Jain, A.K. and R.C. Dubes (1988). *Algorithms for clustering data*. Prentice Hall. Englewood Cliffs.
- Jang, J.-S.R. (1993). Anfis: Adaptive-network-based fuzzy inference systems. *IEEE Transactions on System, Man and Cybernetics* **23**, 665–685.
- Jang, J.-S.R. and C.-T. Sun (1993). Functional equivalence between radial basis function networks and fuzzy inference systems. *IEEE Transactions Neural Networks* **4**, 156–159.
- Jang, J.-S.R., C.-T. Sun and E. Mizutani (1997). *Neuro-fuzzy and Soft computing*. Prentice-Hall. New Jersey.
- Johansen, T.A. and B.A. Foss (1992). A narmax model representation for adaptive control based on local models. *Modeling, Identification and Control* **13**, 25–39.
- Johansen, T.A. and R. Babuska (2002). On multi-objective identification of takagi-sugeno fuzzy model parameters. *Proc. IFAC 15th Triennial World Conference*.

- Johansen, T.A. and R. Murray-Smith (1997). The operating regime approach to nonlinear modelling and control. In: *Multiple model approaches to modelling and control* (R. Murray-Smith and T.A. Johansen, Eds.). Taylor & Francis. London.
- Johansen, T.A., R. Shorten and R. Murray-Smith (2000). On the interpretation and identification of dynamic takagi-sugeno fuzzy models. *IEEE Transactions on Fuzzy Systems* **8**, 297–313.
- Johansson, R. (1993). *System modeling and identification*. Prentice-Hall, Information and System Sciences series. New Jersey.
- Kellet, M.G. (1991). Continuous scheduling of h-infinity controllers for a ms760 paris aircraft. In: *Robust system control design using H-infinity and related methods* (P.H. Hammond, Ed.). pp. 197–223. Institute of Measurement and Control. London.
- Kelly, J.H. and J.H. Evers (1997). *An interpolation strategy for scheduling dynamic compensators*. Proc. of the AIAA Guidance, Navigation and Control Conference, Paper N° A97-37173. New Orleans.
- Khalil, H.K. (1996). *Nonlinear Systems (2nd. ed.)*. Ed. Prentice-Hall. Upper Saddle River, New Jersey.
- Kim, E., P. Minkee, S. Ji and M. Park (1997). A new approach to fuzzy modeling. *Transactions on Fuzzy Systems* **5**, 328–337.
- Kinzel, J., F. Klawonn and R. Kruse (1994). Modifications of genetic algorithms for designing and optimizing fuzzy controllers. [http : //info.et – inf.fho – emden.de/ dmlab/index.html](http://info.et-inf.fho-emden.de/dmlab/index.html).
- Kiriakidis, K. (1999). Nonlinear contro system design via fuzzy modelling and lmis. *International Journal of Control* **72**, 676–685.
- Klawonn, F. and A. Keller (1997). Fuzzy clustering and fuzzy rules. [http : //info.et – inf.fho – emden.de/ dmlab/index.html](http://info.et-inf.fho-emden.de/dmlab/index.html).
- Klawonn, F. and R. Kruse (1995a). Clustering methods in fuzzy control. [http : //info.et – inf.fho – emden.de/ dmlab/index.html](http://info.et-inf.fho-emden.de/dmlab/index.html).
- Klawonn, F. and R. Kruse (1995b). Derivation of fuzzy clasification rules from multidimensional data. [http : //info.et – inf.fho – emden.de/ dmlab/index.html](http://info.et-inf.fho-emden.de/dmlab/index.html).
- Klawonn, F. and R. Kruse (1997). Constructing a fuzzy controller from data. [http : //info.et – inf.fho – emden.de/ dmlab/index.html](http://info.et-inf.fho-emden.de/dmlab/index.html).
- Klawonn, F., R. Kruse and H. Timm (1997). Fuzzy shell cluster analysis. [http : //info.et – inf.fho – emden.de/ dmlab/index.html](http://info.et-inf.fho-emden.de/dmlab/index.html).
- Källström, C.G., K.J. Aström, N.E. Thorell and J. Eriksson amd L. Sten (1979). Adaptive autopilots for tankers. *Automatica* **15**, 241–254.

- Kohonen, T. (1990). The self-organising map. *Proceedings of the IEEE* **78(9)**, 1464–1480.
- Kosko, B. (1986). Fuzzy cognitive maps. *Int. Journal of Man-Machine Studies* **24**, 65–75.
- Krishnapuram, R. and C.P. Freg (1992). Fitting an unknown number of lines and planes to image data through compatible cluster merging. *Pattern Recognition* **25**, 385–400.
- Krishnapuram, R. and J.M. Keller (1993). A possibilistic approach to clustering. *Transactions on Fuzzy Systems* **1**, 98–110.
- Lawrence, D.A. and W.J. Rugh (1990). On a stability theorem for nonlinear systems with slowly varying inputs. *IEEE Transactions on Automatic Control* **35**, 860–864.
- Lawrence, D.A. and W.J. Rugh (1995). Gain scheduling dynamic linear controllers for a nonlinear plant. *Automatica* **31(3)**, 381–390.
- Lee, S. and G. E. Yen (2002). On the local interpretation of takagi-sugeno fuzzy models from a dynamical systems view. *Proc. of the American Control Conference* pp. 519–524.
- Leith, D. J. and W. E. Leithead (1996). Appropriate realisation of gain scheduled controllers with application to wind turbine regulation. *International Journal of Control* **65**, 223–248.
- Leith, D. J. and W. E. Leithead (2000). Survey of gain scheduling analysis and design. *International Journal of Control* **73**, 1001–1025.
- Leith, D.J. and W.E. Leithead (1998*a*). Appropriate realisation of mimo gain scheduled controllers. *International Journal of Control* **70**, 13–50.
- Leith, D.J. and W.E. Leithead (1998*b*). Gain-scheduled and nonlinear systems: dynamic analysis by velocity-based linearisation families. *International Journal of Control* **70**, 289–317.
- Leith, D.J. and W.E. Leithead (1998*c*). Gain-scheduled controller design: an analitic framework directly incorporating non-equilibrium plant dynamics. *International Journal of Control* **70**, 249–269.
- Leith, D.J. and W.E. Leithead (1999*a*). Analytic framework for blended multiple model systems using linear local models. *International Journal of Control* **72**, 605–619.
- Leith, D.J. and W.E. Leithead (1999*b*). Input-output linearisation by velocity-based gain-scheduling. *International Journal of Control* **72**, 229–246.
- Ljung, L. (1999). *System Identification*. Ed. Prentice-Hall. New Jersey.
- Maravall, D. (1993). *Algoritmos de agrupación de clases: clustering*. RA-MA. Barcelona.

- Martinetz, T.M. and K.J. Schulten (1994). Topology representing networks. *Neural Networks* **7**, 518–522.
- Nakamori, Y. and M. Ryoike (1994). Identification of fuzzy prediction models through hyperellipsoidal clustering. *IEEE Transactions on System, Man and Cybernetics* **24**, 1153–1173.
- Nelles, O. (1999). *Nonlinear system identification with local linear neuro-fuzzy models*. Shaker Verlag, PhD dissertation, Darmstadt University of Technology. Darmstadt, Germany.
- Neurofuzzy computing* (1996). NIBS Pte. Ltd. Technical Report TR-960308, [http : //www.singapore.com/productos/mf/ga/neufuzzy.htm](http://www.singapore.com/productos/mf/ga/neufuzzy.htm). New Jersey.
- Nichols, R.A., R.T. Reichert and W.J. Rugh (1993). Gain scheduling for h-infinity controllers: a flight control example. *IEEE Transactions on Control Systems Technology* **1(2)**, 69–79.
- Ohashi, Y. (1984). *Fuzzy clustering and rpbust estimation*. 9th Meeting SAS Users Grp. Int.. Hollywood Beach, Florida.
- Patterson, D. W. (1996). *Artificial neural networks*. Prentice-Hall. New Jersey.
- Pedrycz, W. and J. Waletzky (1997). Fuzzy clustering with partial supervision. *IEEE Transactions on System, Man and Cybernetics-Part B: Cybernetics* **27**, 787–795.
- Previdi, F. (1998). *Identification and control with local linear models*. PhD dissertation, Politecnico di Milano. Milano, Italy.
- Reichert, R. (1992). Dynamic scheduling of modern-robust-control autopilot designs for missiles. *IEEE Control Systems Magazine* **12(5)**, 35–42.
- Ronen, M., Y. Shabtai and H. Guterman (1998). Rapid process modeling-model building methodology combining unsupervised fuzzy-clustering and supervised neural networks. *Computers Chem. Engineering* **22**, S1005–S1008.
- Rugh, W. J. and J. S. Shamma (2000). Research on gain scheduling. *Automatica* **36**, 1401–1425.
- Rugh, W.J. (1983). *Linearization about constant operating points: an input-output viewpoint*. Proc. of the IEEE Conference on Decision and Control. San Antonio, Texas.
- Rugh, W.J. (1991). Analytical framework for gain-scheduling. *IEEE Control Systems Magazine* **11(1)**, 79–84.
- Ruspini, E. (1970). Numerical methods for fuzzy clustering. *Inf. Science* **2**, 319–350.
- Ryoike, M. and Y.Nakamori (1996). Simultaneous analysis of classification and regression by adaptive fuzzy clustering. *Japanese Journal of Fuzzy Theory and Systems* **8**, 99–113.

- Sandberg, I. W. (1964). In the l2-boundedness of solutions of nonlinear functional equations. *Bell Sys. Tech. J.* **43**, 1581–1597.
- Söderström, T. and P. Stoica (1989). *System identification*. Prentice-Hall. New York.
- Seber, G.A.F. and C.J. Wild (1989). *Nonlinear regression*. John Wiley & Sons. New York.
- Setnes, M. (1995). *Fuzzy rule base simplification using similarity measures*. M.Sc. thesis, Delft University of Technology. Delft, The Netherlands.
- Shahruz, S.M. and S. Behtash (1992). Design of controllers for linear parameter-varying systems by the gain scheduling technique. *Journal of Mathematical Analysis and Applications* **168(1)**, 195–217.
- Shamma, J.S. and M. Athans (1990). Analysis of nonlinear gain scheduled control systems. *IEEE Transactions on Automatic Control* **35(8)**, 898–907.
- Shamma, J.S. and M. Athans (1991). Guaranteed properties of gain-scheduled control for linear parameter-varying plants. *Automatica* **27**, 559–564.
- Shorten, R., R. Murray-Smith, R. Bjorgan and H. Golle (1999). On the interpretation of local models in blended multiple model structures. *International Journal of Control* **72**, 620–628.
- Smith, S. M., B.R. Nokleby and D. J. Comer (1993). A computational approach to fuzzy logic controller design and analysis using cell state space methods. In: *Fuzzy Control Systems* (A. Kandel and G. Langholz, Eds.). pp. 398–427. CRC Press. London.
- Sontag, E.D. (1987). Controllability and linearised regulation. *IEEE Transactions on Automatic Control* **32**, 877–878.
- Stein, G., G.L. Hartmann and R. Hendrick (1977). Adaptive control laws for f-8 flight test. *IEEE Transactions on Automatic Control* **22(5)**, 758–767.
- Stilwell, D.J. and W. J. Rugh (1999). Interpolation of observer state feedback controllers for gain scheduling. *IEEE Transactions on Automatic Control* **44(6)**, 1225–1229.
- Stilwell, D.J. and W. J. Rugh (2000). Stability preserving interpolation methods for gain scheduled controllers. *Automatica* **36(5)**, 665–671.
- Sugeno, M. and T. Yasukawa (1993). A fuzzy-logic-based approach to qualitative modeling. *Transactions on Fuzzy Systems* **1**, 7–31.
- Sutanto, E.L., J.D. Mason and K. Warwick (1997). Mean-tracking clustering algorithm for radial basis function centre selection. *International Journal of Control* **67**, 961–977.

- T-M.Guerra and L. Vermeiren (2001). Conditions for non quadratic stabilization of discrete fuzzy models. *Proc. IFAC Workshop on Advanced Fuzzy-Neural Control* pp. 1–6.
- Takagi, T. and M. Sugeno (1985). Fuzzy identification of systems and its applications to modeling and control. *IEEE Transactions on System, Man and Cybernetics* **15**, 116–132.
- Tanaka, K. and H. O. Wang (2001). *Fuzzy control systems design and analysis*. John Wiley & Sons. New York.
- Tanaka, K. and M. Sano (1994). Robust stabilisation problem of fuzzy control systems and its application to backing-up of a truck trailer. *IEEE Transactions on Fuzzy Systems* **2**, 119–134.
- Tanaka, K. and M. Sugeno (1992). Stability analysis and design of fuzzy control systems. *Fuzzy Sets and Systems* **45**, 135–156.
- Tanaka, K., T. Ikeda and H. Wang (1998). Fuzzy regulators and observers: relaxed stability conditions and lmi-based designs. *IEEE Transactions on Fuzzy Systems* **6**, 250–265.
- Visyasagar, M. (1993). *Nonlinear Systems Analysis*. Ed. Prentice-Hall. New Jersey.
- Wang, H. O., K. Tanaka and M. F. Griffin (1996). An approach to fuzzy control of nonlinear systems: Stability and design issues. *Transactions on Fuzzy Systems* **4**, 14–23.
- Wang, J. and W. Rugh (1987a). Feedback linearisation families for nonlinear systems. *IEEE Transactions on Automatic Control* **32**, 935–940.
- Wang, J. and W. Rugh (1987b). Parameterised linear systems and linearisation families for nonlinear systems. *IEEE Transactions on Circuits and Systems* **34**, 650–657.
- Wang, L.-X. (1994). *Adaptive Fuzzy Systems and Control: Design and Stability Analysis*. Prentice-Hall. New Jersey.
- Wang, L.-X. (1997). *A Course in Fuzzy Systems and Control*. Prentice-Hall. New Jersey.
- Yager, R.R. and D.P. Filev (1994). Approximate clustering via the mountain method. *IEEE Transactions on Systems, Man and Cybernetics* **24**, 1279–1284.
- Yang, M.-S. and C.-H. Ko (1997). On cluster-wise fuzzy regression analysis. *IEEE Transactions on Systems, Man and Cybernetics - Part B: Cybernetics* **27**, 1–13.
- Zadeh, L.A. (1965). Fuzzy sets. *Information Control* **8**, 338–353.
- Zadeh, L.A. (1973). Outline of a new approach to the analysis of complex systems and decision processes. *IEEE Transactions on Systems, Man and Cybernetics* **1**, 28–44.

Parte VI

Anexos

Apéndice A

Modelos locales

En este anexo se plantean las diferentes formas de realizar particiones de modelos locales de sistemas no lineales, todas ellas mediante linealización (desarrollo en serie) en el entorno de puntos de operación empleando el Jacobiano de la planta.

Para ello, en primer lugar, se definirán los modelos matemáticos de los sistemas no lineales que se van a estudiar. A continuación se verá el enfoque más habitual de partición que es aquel en el que los puntos de operación se corresponden con uno o varios (familias de linealizaciones) puntos de equilibrio. La limitación de estas opciones es que cualquier análisis o diseño basado en esos modelos solo será válido cuando se opere cerca del equilibrio. Por ello se revisarán a continuación las técnicas que permiten la linealización en torno a puntos que no son de equilibrio, como las linealizaciones en el entorno de puntos de operación o la linealización basada en la velocidad.

A.1. Introducción

El modelado y control basado en modelos locales (Johansen and Murray-Smith, 1997) consiste en dividir un sistema no complejo en subsistemas más sencillos que puedan ser analizados de forma individual. La clave, por tanto, consiste en la elección de aquellos parámetros adecuados que permitan realizar dicha partición. Por otra parte, se hace necesario un elemento supervisor que sea de algún modo capaz de integrar de nuevo todos estos subsistemas en un sistema global. Si se eligen modelos lineales como modelos locales (todos ellos con la misma estructura, manteniéndose así las variables de estado), la integración de todos ellos nos lleva a un *modelo lineal de parámetros variables* LPV (linear parameter-varying), los cuales son la base de las técnicas clásicas para el desarrollo de controladores por planificación de ganancia (Shamma and Athans, 1990; Rugh, 1991; Aström and Wittenmark, 1995).

Cuanto mayor sea el número de modelos locales elegidos, más sencillo será encontrar un hipotético controlador que funcione adecuadamente para cada uno de los regímenes. Sin embargo, esta propiedad se enfrenta a la denominada maldición de la dimensionalidad: una partición uniforme lleva a que el número de particiones crezca de forma exponencial con el número de variables.

Para obtener un modelo LPV, la opción más común está basada en la linealización del *Jacobiano* de la planta no lineal en el entorno de una familia de puntos de operación,

que consiste en aplicar la teoría del desarrollo en serie en torno a esos puntos, olvidando los términos de orden superior (Leith and Leithead, 2000). Lo más habitual es que los puntos de operación sean puntos de equilibrio (Hunt and Johansen, 1997), pero en cualquier caso se llega a una familia parametrizada de plantas linealizadas y es la base de lo que se llama planificación por linealización. Generalmente, la parametrización corresponde a valores fijos de las variables de planificación o funciones de ellas, que son variables internas de la planta y/o señales exógenas.

A.2. Modelos de sistemas no lineales

El modelo que se va a considerar de los sistemas no lineales que se desean estudiar (identificar y controlar) será el de la planta genérica no lineal dada por:

$$\begin{aligned}\dot{x} &= f(x, u), \\ y &= g(x, u).\end{aligned}\tag{A.1}$$

donde x es el vector de estado $n \times 1$, u es el vector de entrada $m \times 1$, y es el vector de salida $p \times 1$ medible para el controlador (variables penalizadas, señales de mando para referencias, algunas variables de estado), y se ha asumido que x y u incluyen todas las dependencias con variables exógenas. Finalmente, $f(\cdot)$ y $g(\cdot)$ son funciones vectoriales suaves, es decir, con derivadas continuas (primera y siguientes) en cualquier instante.

Si se considera (como ocurre en muchas aplicaciones reales para control) que no hay dependencia directa de la salida con la entrada (Rugh and Shamma, 2000), la variable u desaparece de los argumentos de g en (3.1). Finalmente, con ánimo de simplificar, muchas veces se asume que el vector de estado es medible en su totalidad. Teniendo en cuenta estas dos últimas consideraciones (Johansen and Foss, 1992), la ecuación de salida resultante es la identidad y la representación del sistema se reduce tan solo a una ecuación (la primera) del sistema:

$$\begin{aligned}\dot{x} &= f(x, u), \\ y &= x,\end{aligned}\tag{A.2}$$

con $x = y \in R^n$, $u \in R^m$, siendo $f(\cdot)$ una función suave. Esta única ecuación consiste, en el caso discreto, en una dependencia del valor de la salida actual con un vector de salidas (anteriores) y entradas (actual y anteriores), dado que el estado se podrá reconstruir completamente si dicho vector es lo suficientemente amplio (Söderström and Stoica, 1989). Dicha ecuación, en su versión discreta, no es más que el conocido modelo no lineal autoregresivo con entrada exógena NARX (nonlinear autoregressive model with exogenous input) de un sistema:

$$y_{k+1} = f(y_k, \dots, y_{k-n_y}, u_k, \dots, u_{k-n_u}).\tag{A.3}$$

Desde estos modelos genéricos de los sistemas no lineales, existen diferentes opciones para asociarles un sistema lineal (o conjunto de ellos). Como ya se ha comentado,

la opción más usada (y que se va a desarrollar aquí) es la que emplea la teoría de linealización por desarrollo en serie (mediante el Jacobiano) de la planta no lineal en el entorno de una familia puntos de operación.

Dentro de la linealización en el entorno de puntos de operación empleando el Jacobiano de la planta, existen diferentes posibilidades. Lo más habitual (por la gran cantidad de literatura existente al respecto (Rugh and Shamma, 2000)) es que los puntos de operación se correspondan con una trayectoria deseada o un punto de equilibrio, con la desventaja de que el desarrollo en serie solo es válido en el entorno de esa trayectoria específica o ese punto de equilibrio. Esta condición se puede relajar desarrollando familias de linealizaciones en torno a conjuntos de trayectorias deseadas o puntos de equilibrio, con lo que se aumenta el rango de operación del sistema.

La limitación común a las opciones descritas sobre linealización en el entorno de puntos de equilibrio es que cualquier análisis o diseño basado en esos modelos, generalmente, solo será válido cuando se opere cerca del equilibrio. Afortunadamente, existen algunas técnicas que permiten la linealización en torno a puntos que no son de equilibrio, como las linealizaciones en el entorno de puntos de operación (Hunt and Johansen, 1997) o la linealización basada en la velocidad (Leith and Leithead, 2000; Rugh and Shamma, 2000). Dichas técnicas también serán comentadas en este capítulo.

A.3. Linealización en torno a puntos de equilibrio

Los diseños de planificación de ganancia emplean habitualmente una estrategia de descomposición de las tareas no lineales en un conjunto de subtareas lineales. Esta descomposición depende del establecimiento de una relación entre el sistema no lineal y la familia de sistemas lineales. Los resultados teóricos que existen para relacionar las características dinámicas del sistema no lineal con las de la familia lineal que se le asocia, y que se van a ver a continuación, son de uno de los siguientes tipos:

- resultados de estabilidad, que establecen una relación entre la estabilidad del sistema no lineal y la del sistema lineal asociado, y
- resultados de aproximación, que establecen una relación directa entre la solución del sistema no lineal y la solución del sistema lineal que se le asocia.

Los primeros son mucho más limitados que los segundos, reduciéndose a especificar condiciones bajo las que si la solución del sistema lineal es acotada implica que la solución del sistema no lineal también es acotada para un conjunto adecuado de entradas y condiciones iniciales (Leith and Leithead, 2000).

A.3.1. Linealización clásica en torno a un punto de equilibrio: linealización por Jacobiano

Aunque es un caso poco útil en la práctica de la identificación de sistemas no lineales y, además, queda incluido en la generalización a la linealización en el entorno de una

familia de puntos de equilibrio que se verá posteriormente, conviene revisarlo por su importancia como base para desarrollos posteriores.

La linealización por desarrollo en serie de un sistema no lineal es bien conocida. Consideremos un sistema no lineal como el descrito en (A.1) y sea (x_e, u_e) un punto de equilibrio del sistema no lineal, es decir, que $f(x_e, u_e) = 0$. Obviando los términos de orden superior, el sistema puede ser aproximado localmente por el sistema lineal invariante en el tiempo:

$$\begin{aligned}\delta\dot{x} &= \nabla_x f(x_e, u_e)\delta x + \nabla_u f(x_e, u_e)\delta u, \\ \delta\dot{y} &= \nabla_x g(x_e, u_e)\delta x + \nabla_u g(x_e, u_e)\delta u,\end{aligned}\tag{A.4}$$

donde $\delta x = x - x_e$, $\delta u = u - u_e$, $\delta y = y - y_e$ y, por ejemplo, $\nabla_x f$ indica la matriz Jacobiana de f respecto de x (lo mismo es aplicable al resto de ∇).

El estudio de la estabilidad es bien conocida en el caso de un sistema lineal e invariante en el tiempo (Visyasagar, 1993): el sistema no lineal (A.1) es estable en (x_e, u_e) siempre que la dinámica dada por (A.4) sea estable respecto a la aproximación del error que aparece al truncar el desarrollo en serie (nos movamos en un entorno lo suficientemente pequeño del punto de equilibrio).

En el caso discreto, el punto de equilibrio del sistema no lineal es (x_e, u_e) , en el que $x_e = f(x_e, u_e)$. Obviando los términos de orden superior, este sistema también puede ser aproximado localmente mediante desarrollo en serie por un sistema discreto lineal invariante en el tiempo, que en el caso de trabajar con un modelo discreto como (A.3) se llega a la ecuación:

$$\delta y_{k+1} = \nabla_y f(y_e, u_e)\delta y_k + \nabla_u f(y_e, u_e)\delta u_k,\tag{A.5}$$

donde $\delta u_k = u_k - u_e$ y $\delta y_k = y_k - y_e$.

A.3.2. Múltiples linealizaciones en torno a un punto de equilibrio

En este caso se representa muy bien el comportamiento transitorio del sistema, ya que se asocian diversos modelos lineales al punto de equilibrio (Wang *et al.*, 1996), dependiendo del valor de una variable adicional no incluida en el modelo y que caracteriza los posibles diferentes modos de funcionamiento. Con esta opción se puede llegar a controladores con un buen comportamiento durante el transitorio, al permitir asociar controladores con regímenes de operación transitorios. Sin embargo, dado que solo se considera un punto de equilibrio, esta opción es más adecuada para la regulación y la estabilización que para los problemas de control y seguimiento con múltiples equilibrios.

A.3.3. Familias de linealizaciones por Jacobiano en torno a diferentes puntos de equilibrio

La aplicación de la teoría de linealización clásica requiere que la linealización se corresponda con un punto de equilibrio del sistema no lineal. Esta es una restricción

muy seria que puede llevar a un mal comportamiento transitorio del sistema de control por planificación de ganancia, que en este caso son planificados con respecto a la salida, el estado o alguna variable auxiliar. El sistema puede incluso llegar a ser inestable si se opera muy lejos del equilibrio.

Consideremos de nuevo el sistema no lineal descrito en (A.1). Supongamos que este sistema tiene un conjunto de puntos de equilibrio (x_e, u_e) tales que para todos ellos $f(x_e, u_e) = 0$. Todos estos puntos de equilibrio se pueden parametrizar en función de una variable de planificación σ , que puede incluir todas las variables medibles del sistema (internas o externas) con la única restricción de que $\sigma(t)$ siempre esté disponible (Rugh and Shamma, 2000), ya que podría ser usada posteriormente para el diseño del control

Las funciones $x_e(\sigma)$ y $u_e(\sigma)$ definen una *familia de equilibrio* para la planta en el conjunto S si

$$f(x_e(\sigma), u_e(\sigma)) = 0, \quad \sigma \in S \quad (\text{A.6})$$

Obviando los términos de orden superior, el sistema puede ser aproximado localmente por el sistema lineal variante en el tiempo:

$$\begin{aligned} \delta \dot{x} &= \nabla_x f(x_e(\sigma), u_e(\sigma)) \delta x + \nabla_u f(x_e(\sigma), u_e(\sigma)) \delta u, \\ \delta y &= \nabla_x g(x_e(\sigma), u_e(\sigma)) \delta x + \nabla_u g(x_e(\sigma), u_e(\sigma)) \delta u, \end{aligned} \quad (\text{A.7})$$

donde, $\delta x = x - x_e(\sigma)$, $\delta u = u - u_e(\sigma)$, $\delta y = y - y_e(\sigma)$, $\sigma \in S$ y, por ejemplo, $\nabla_x f$ indica la matriz Jacobiana de f respecto de x (lo mismo es aplicable al resto de ∇). La interpretación es que para cada valor de linealización σ , el sistema (A.7) describe el comportamiento local de la planta no lineal en el entorno del equilibrio correspondiente. En el contexto de la planificación de ganancia, se suele emplear (Desoer, 1969; Ilchmann *et al.*, 1987) la teoría de tiempo detenido (frozen-time) (Hoppensteadt, 1966; Lawrence and Rugh, 1990) para establecer condiciones suficientes (y, en general, muy conservadoras) de estabilidad del sistema lineal variante en el tiempo (A.4). Es importante hacer notar que en todos los resultados sobre robustez en tiempo detenido, un incremento en la robustez requiere un decremento en el ratio de variación de $\nabla_x f(x_e(\sigma), u_e(\sigma))$, heredando el sistema lineal variante en el tiempo la robustez de la familia en tiempo detenido solo en el caso de variación arbitrariamente lenta (?; Shamma and Athans, 1991; Leith and Leithead, 1998b). Además, es habitual que las condiciones de estabilidad que se obtienen sean difíciles de comprobar, pues pueden incluir cantidades difíciles de evaluar (Shamma and Athans, 1990).

Con las mismas consideraciones que la aproximación al sistema continuo, se puede realizar una representación análoga en su versión discreta. En este caso, las funciones $x_e(\sigma)$ y $u_e(\sigma)$ definen una *familia de equilibrio* para la planta discreta en el conjunto S si

$$f(x_e(\sigma), u_e(\sigma)) = x_e(\sigma), \quad \sigma \in S \quad (\text{A.8})$$

Obviando los términos de orden superior, el sistema puede ser aproximado localmente mediante desarrollo en serie por un sistema discreto lineal invariante en el tiempo.

Considerando un modelo discreto como (A.3) queda:

$$\delta y_{k+1} = \nabla_y f(y_e(\sigma), u_e(\sigma)) \delta y_k + \nabla_u f(y_e(\sigma), u_e(\sigma)) \delta u_k, \quad (\text{A.9})$$

donde $\delta u_k = u_k - u_e$, $\delta y_k = y_k - y_e$, y $\sigma \in S$ (según (A.8)).

Algunos ejemplos de modelado con las aproximaciones descritas se pueden encontrar en (Nichols *et al.*, 1993; Rugh and Shamma, 2000).

A.4. Linealizaciones en puntos fuera del equilibrio

La restricción de la operación en el entorno de los puntos de equilibrio que tienen los métodos clásicos de planificación de ganancia es el resultado de las limitaciones de la teoría convencional de linealización por desarrollo en serie. Específicamente, como se ha visto, la teoría clásica de linealización asocia un sistema lineal solo a puntos de operación en el equilibrio. Esta limitación se resuelve considerando marcos alternativos de linealización.

A.4.1. Linealización dinámica en torno a una trayectoria nominal

La teoría de linealización en serie no solo puede aproximar un sistema no lineal por otro lineal en el entorno de un punto de equilibrio, sino también en el entorno de una trayectoria (Sontag, 1987). Sin embargo, la linealización en el entorno de una trayectoria es un sistema lineal variante con el tiempo en lugar de los sistemas lineales invariantes en el tiempo que aparecen en la linealización en torno a un punto de equilibrio.

Además, el diseño del control para el sistema lineal (variante con el tiempo) que resulta es considerablemente más difícil que el que resulta por la linealización en el entorno de puntos de equilibrio. Las linealizaciones en el entorno de puntos de operación que se estudiarán a continuación pueden considerarse una generalización de esta opción de trabajo.

A.4.2. Linealización en torno a puntos de operación

Consideremos de nuevo el sistema no lineal dado por (A.1). Lo habitual es considerar la familia de puntos de equilibrio del sistema dados por (A.6) y aproximarlos localmente por el sistema lineal variante en el tiempo (A.7). Una interpretación de esta dinámica clásica linealizada es que aproxima la dinámica del sistema no lineal para valores iniciales y entradas que se desvían ligeramente de x_e y u_e respectivamente. La razón más importante de la popularidad de la linealización clásica es la existencia de una teoría completa y perfectamente desarrollada para el análisis de las propiedades de estabilidad local en el equilibrio de los sistemas no lineales en términos de sus versiones linealizadas, así como la existencia de la conocida teoría de control basada en la linealización.

Supongamos que la dinámica del sistema no lineal se aproxima en el entorno de un punto de operación $(x_o, u_o) \in R^{n+m}$, el cual no es necesario que sea un punto de

equilibrio (Wang *et al.*, 1996; Hunt and Johansen, 1997; Shorten *et al.*, 1999). Es decir, los posibles puntos de operación del sistema vienen dados por:

$$\psi = \{(x, u) \in R^{n+m}\} \quad (\text{A.10})$$

con lo que la linealización del sistema no lineal en torno al punto (x_o, u_o) es:

$$\begin{aligned} \dot{x} &= \nabla_x f(x_o, u_o)\delta x + \nabla_u f(x_o, u_o)\delta u + f(x_o, u_o), \\ y &= \nabla_x g(x_o, u_o)\delta x + \nabla_u g(x_o, u_o)\delta u + g(x_o, u_o), \end{aligned} \quad (\text{A.11})$$

donde $\delta x = x - x_o$, $\delta u = u - u_o$, por ejemplo, $\nabla_x f$ indica la matriz Jacobiana de f respecto de x (lo mismo es aplicable al resto de ∇). Esta ecuación se puede reescribir de una forma más adecuada para una posterior extensión a un conjunto de puntos de operación:

$$\begin{aligned} \dot{x} &= \nabla_x f(x_o, u_o)x + \nabla_u f(x_o, u_o)u + d_f(x_o, u_o), \\ y &= \nabla_x g(x_o, u_o)x + \nabla_u g(x_o, u_o)u + d_g(x_o, u_o), \end{aligned} \quad (\text{A.12})$$

donde

$$\begin{aligned} d_f(x_o, u_o) &= f(x_o, u_o) - \nabla_x f(x_o, u_o)x_o - \nabla_u f(x_o, u_o)u_o, \\ d_g(x_o, u_o) &= g(x_o, u_o) - \nabla_x g(x_o, u_o)x_o - \nabla_u g(x_o, u_o)u_o. \end{aligned} \quad (\text{A.13})$$

Desde un punto de vista matemático, estas linealizaciones en el entorno de puntos de operación (Hunt and Johansen, 1997) corresponden a aproximar las funciones f y g por sus planos tangentes en el punto (x_o, u_o) . Sin embargo, la interpretación del sistema (A.12) es conceptualmente diferente a la clásica, ya que (x_o, u_o) solo puede ser un estado transitorio y nunca un punto de equilibrio. Una interpretación de (A.12) es que aproxima la (posiblemente transitoria) dinámica del sistema no lineal cuando la trayectoria está cerca a (x_o, y_o) . Otra interpretación que está relacionada, es que el flujo de la dinámica linealizada aproxima el flujo del sistema no lineal para estados y entradas que se desvían ligeramente de x_o y u_o . Esta aproximación no es de inmediata utilidad para el análisis de estabilidad, pero es muy útil para el diseño de control estable de altas prestaciones por planificación de ganancia. El beneficio final de esta forma de trabajo es que la dinámica del transitorio (fuera del equilibrio) del sistema de control puede ser mejorada significativamente.

Esta linealización en un punto de operación fuera del equilibrio se puede extender a un conjunto de puntos Ψ del tipo (A.10), es decir:

$$\psi = \{(x, u) \in R^{n+m}\}, \quad \psi \in \Psi. \quad (\text{A.14})$$

Todos estos puntos de operación se pueden parametrizar en función de una variable de planificación ϕ , que puede incluir todas las variables medibles del sistema (internas o externas, estén en ψ o no) con la única restricción de que $\phi(t)$ siempre esté disponible (Rugh and Shamma, 2000), ya que podría ser usada posteriormente para el diseño del

control. A $\phi \in \Phi$ se le llama régimen de operación y al conjunto de todos ellos, Φ , conjunto de operación.

La diferencia entre la linealización en el entorno de puntos fuera del equilibrio y la linealización dinámica a lo largo de una trayectoria (brevemente comentada anteriormente) depende únicamente de la granularidad del conjunto de puntos elegido en la primera (Leith and Leithead, 2000). La razón de esto es que el sistema lineal variante con el tiempo LTV (linear time variant) resultante por linealización dinámica solo depende del punto por el que pasa la trayectoria en cierto instante de tiempo. Así pues, la linealización en puntos fuera del equilibrio lleva a una aproximación arbitrariamente cercana al sistema LTV en términos de conjuntos de sistemas LTI.

En definitiva, eliminando los términos de orden superior, el sistema (A.1) puede ser aproximado por el sistema lineal variante en el tiempo llamado red de modelos locales:

$$\begin{aligned} \dot{x} &= \nabla_x f(x_o(\phi), u_o(\phi))x + \nabla_u f(x_o(\phi), u_o(\phi))u + d_f(x_o(\phi), u_o(\phi)), \\ y &= \nabla_x g(x_o(\phi), u_o(\phi))x + \nabla_u g(x_o(\phi), u_o(\phi))u + d_g(x_o(\phi), u_o(\phi)), \end{aligned} \quad (\text{A.15})$$

donde $\phi \in \Phi$, $d_f(\cdot)$ y $d_g(\cdot)$ siguen la definición de (A.13) y, por ejemplo, $\nabla_x f$ indica la matriz Jacobiana de f respecto de x (lo mismo es aplicable al resto de ∇). La interpretación es que para cada valor de linealización ϕ , el sistema (A.15) describe el comportamiento local de la planta no lineal en el entorno del punto de operación correspondiente.

La versión discreta de esta aproximación viene dada por:

$$\begin{aligned} x_{k+1} &= \nabla_x f(x_o(\phi), u_o(\phi))x_k + \nabla_u f(x_o(\phi), u_o(\phi))u_k + d_f(x_o(\phi), u_o(\phi)), \\ y_k &= \nabla_x g(x_o(\phi), u_o(\phi))x_k + \nabla_u g(x_o(\phi), u_o(\phi))u_k + d_g(x_o(\phi), u_o(\phi)), \end{aligned} \quad (\text{A.16})$$

donde $\phi \in \Phi$, $d_f(\cdot)$ y $d_g(\cdot)$ siguen la definición de (A.13) y, por ejemplo, $\nabla_u g$ indica la matriz Jacobiana de g respecto de u (lo mismo es aplicable al resto de ∇). La aplicación de todo lo visto al modelo discreto (A.3) lleva a la expresión:

$$y_{k+1} = \nabla_y f(y_o(\phi), u_o(\phi))y_k + \nabla_u f(y_o(\phi), u_o(\phi))u_k + d_f(y_o(\phi), u_o(\phi)). \quad (\text{A.17})$$

El análisis de los resultados de estabilidad de estos modelos en los puntos de operación realizado, por ejemplo, empleando la teoría de tiempo detenido (frozen-time) para establecer condiciones suficientes de estabilidad del sistema lineal variante en el tiempo (A.15) o (A.16) no tiene sentido, ya que (x_o, u_o) puede ser, como ya se ha comentado, un estado transitorio y no un punto de equilibrio. Sin embargo, se podrá llegar a resultados como cotas (Hunt and Johansen, 1997) (aplicables a herramientas de control robusto) en las que interviene la granularidad del modelo y la suavidad del cambio entre puntos de operación para asegurar un buen comportamiento del control basado en estos modelos (Díez and Previdi, 2001).

En (Johansen and Foss, 1992; Hunt and Johansen, 1997) se pueden encontrar ejemplos en los que se aplican los modelos descritos en este apartado.

A.4.3. Linealización basada en la velocidad

La limitación de la teoría clásica de linealización en la que solo se asocia un sistema lineal en puntos de equilibrio, se aborda de forma directa por la teoría de linealización basada en la velocidad (Leith and Leithead, 2000; Rugh and Shamma, 2000).

En (Leith and Leithead, 1998b) se demuestra que la solución a la linealización basada en la velocidad:

$$\begin{aligned}\dot{x} &= v, \\ \dot{v} &= (\nabla_x f(x_o, u_o) + \nabla_x d_f(x_o, u_o))v + (\nabla_u f(x_o, u_o) + \nabla_u d_f(x_o, u_o))\dot{u}, \\ \dot{y} &= (\nabla_x g(x_o, u_o) + \nabla_x d_g(x_o, u_o))v + (\nabla_u g(x_o, u_o) + \nabla_u d_g(x_o, u_o))\dot{u},\end{aligned}\quad (\text{A.18})$$

aproxima localmente la solución del sistema no lineal (A.1) en un punto de operación genérico (x_o, u_o) (no necesariamente un punto de equilibrio).

La obtención de (A.18) desde el modelo del sistema no lineal es muy sencilla. Para ello simplemente se reescribe (A.1) separando la parte lineal de la no lineal:

$$\begin{aligned}\dot{x} &= v, \\ \dot{v} &= (\nabla_x f(x, u) + \nabla_x d_f(x, u))v + (\nabla_u f(x, u) + \nabla_u d_f(x, u))\dot{u}, \\ \dot{y} &= (\nabla_x g(x, u) + \nabla_x d_g(x, u))v + (\nabla_u g(x, u) + \nabla_u d_g(x, u))\dot{u},\end{aligned}\quad (\text{A.19})$$

y se diferencia, siendo (A.18) la versión de (A.19) para un instante temporal concreto y en el punto de operación (x_o, u_o) . Además, se puede observar que (A.19) coincide con (A.12) en los puntos de operación.

La solución de la linealización basada en la velocidad (A.18) es solo una aproximación local, pero se puede encontrar una solución asociada a cada punto de operación de un sistema no lineal, con grado de precisión arbitrario, y parametrizarlas en función de una variable de planificación ρ (con las mismas características de ϕ en (A.15)). Así pues, una familia de linealizaciones basadas en la velocidad parametrizada por la variable de planificación ρ engloba toda la dinámica del sistema no lineal (ρ captura la estructura no lineal del sistema) quedando:

$$\begin{aligned}\dot{x} &= v, \\ \dot{v} &= (\nabla_x f(x_o(\rho), u_o(\rho)) + \nabla_x d_f(x_o(\rho), u_o(\rho)))v + \\ &\quad + (\nabla_u f(x_o(\rho), u_o(\rho)) + \nabla_u d_f(x_o(\rho), u_o(\rho)))\dot{u}, \\ \dot{y} &= (\nabla_x g(x_o(\rho), u_o(\rho)) + \nabla_x d_g(x_o(\rho), u_o(\rho)))v + \\ &\quad + (\nabla_u g(x_o(\rho), u_o(\rho)) + \nabla_u d_g(x_o(\rho), u_o(\rho)))\dot{u},\end{aligned}\quad (\text{A.20})$$

Esta representación tiene la ventaja, con respecto a (A.15), de que los offsets (funciones no lineales d_f y d_g) han desaparecido. La relación entre la solución de un sistema no lineal y las soluciones de los miembros de la familia de linealizaciones basadas en la velocidad asociada a él, se puede usar para derivar condiciones que relacionen la estabilidad del sistema no lineal con la estabilidad de sus linealizaciones basadas en la velocidad (Leith and Leithead, 1998b). Métodos generales de análisis de la estabilidad

como la teoría de pequeña ganancia o la de Lyapunov se pueden aplicar para derivar condiciones de estabilidad basadas en la velocidad.

La linealización basada en la velocidad se convierte en una linealización basada en incrementos (Δ) para el caso discreto:

$$\begin{aligned}\Delta x_k &= x_k - x_{k-1}, \\ \Delta x_{k+1} &= (\nabla_x f(x_o, u_o) + \nabla_x d_f(x_o, u_o))\Delta x_k + (\nabla_u f(x_o, u_o) + \nabla_u d_f(x_o, u_o))\Delta u_k, \\ \Delta y_k &= (\nabla_x g(x_o, u_o) + \nabla_x d_g(x_o, u_o))\Delta x_k + (\nabla_u g(x_o, u_o) + \nabla_u d_g(x_o, u_o))\Delta u_k,\end{aligned}\tag{A.21}$$

que aproxima (para periodos de muestreo pequeños) al sistema no lineal localmente en un punto de operación (x_o, u_o) , de nuevo con la ventaja de que los offsets (funciones no lineales d_f y d_g) han desaparecido. (A.21) se puede obtener, de forma análoga al caso continuo, aplicando incrementos en todas las variables (por ejemplo, $\Delta \hat{x}_k = \hat{x}_k - \hat{x}_{k-1}$ para $x(k)$) del modelo discreto reescrito de forma que queden separadas la parte lineal de la no lineal.

En este caso discreto también se puede encontrar una aproximación local del tipo linealización basada en la velocidad asociada a cada punto de operación del sistema no lineal, con grado de precisión arbitrario (dependiente de un elevado número de puntos y un bajo periodo de muestreo). Así pues, una familia de linealizaciones basadas en la velocidad parametrizada otra vez por la variable de planificación ρ engloba toda la dinámica del sistema no lineal discreto, quedando:

$$\begin{aligned}\Delta x_k &= x_k - x_{k-1}, \\ \Delta x_{k+1} &= (\nabla_x f(x_o(\rho), u_o(\rho)) + \nabla_x d_f(x_o(\rho), u_o(\rho)))\Delta x_k + \\ &\quad + (\nabla_u f(x_o(\rho), u_o(\rho)) + \nabla_u d_f(x_o(\rho), u_o(\rho)))\Delta u_k, \\ \Delta y_k &= (\nabla_x g(x_o(\rho), u_o(\rho)) + \nabla_x d_g(x_o(\rho), u_o(\rho)))\Delta x_k + \\ &\quad + (\nabla_u g(x_o(\rho), u_o(\rho)) + \nabla_u d_g(x_o(\rho), u_o(\rho)))\Delta u_k,\end{aligned}\tag{A.22}$$

Tomando modelos discretos como (A.3), la aplicación de la linealización basada en velocidad proporciona un resultado como:

$$\begin{aligned}\Delta x_k &= x_k - x_{k-1}, \\ \Delta y_{k+1} &= (\nabla_x f(x_o(\rho), u_o(\rho)) + \nabla_x d_f(x_o(\rho), u_o(\rho)))\Delta x_k + \\ &\quad + (\nabla_u f(x_o(\rho), u_o(\rho)) + \nabla_u d_f(x_o(\rho), u_o(\rho)))\Delta u_k.\end{aligned}\tag{A.23}$$

Algunos ejemplos de linealizaciones basadas en velocidad se pueden revisar en (Leith and Leithead, 1998b).

Apéndice B

Planificación de ganancia

En este anexo se verán técnicas de control aplicables a modelos identificados desde los datos experimentales que sean realmente linealizaciones del proceso en el entorno de un conjunto de puntos: el control por planificación de ganancia.

B.1. Introducción

Cuando se dispone de un modelo borroso lo suficientemente preciso de un proceso no lineal, éste se puede usar para diseñar un controlador para el proceso. En (Babuska, 1996) se comenta la posibilidad de diseñar un controlador no lineal por inversión del modelo borroso del sistema (siempre que la dinámica del proceso invertido sea estable), o usar las capacidades de predicción del modelo borroso para diseñar un control predictivo basado en el modelo.

Si los modelos lineales identificados desde los datos experimentales son realmente linealizaciones del proceso (como busca el algoritmo de agrupamiento AFCRC), se pueden emplear las mismas técnicas de control que se basan en representar sistemas no lineales por un modelo lineal que lo aproxime y diseñar un controlador para él. La restricción de que dicho modelo solo puede obtener una representación local del comportamiento del sistema, tradicionalmente se elimina diseñando diferentes controladores para diferentes modelos lineales en diferentes puntos de operación e interpolando estos controladores, con lo que se llega a un controlador no lineal global. A grandes rasgos, esto es lo que se conoce como *control por planificación de ganancia* y parece directamente aplicable a los modelos identificados con los algoritmos de agrupamiento comentados en los capítulos anteriores.

La planificación de ganancia es una de las técnicas más habituales para el control de procesos no lineales y ha sido empleada satisfactoriamente desde hace años (Källström *et al.*, 1979; Stein *et al.*, 1977) en aplicaciones que van desde la ingeniería aeroespacial al control de procesos. Sin embargo, hasta los años noventa es raro encontrar un tratamiento teórico de la planificación de ganancia (Shamma and Athans, 1990; Rugh, 1991; Åström and Wittenmark, 1995), aunque afortunadamente en los últimos años han aparecido revisiones completas sobre el tema (Leith and Leithead, 2000; Rugh and Shamma, 2000).

B.2. Control por planificación de ganancia

El diseño de un controlador por planificación de ganancia para una planta no lineal se puede describir como un procedimiento de cuatro pasos (Aström and Wittenmark, 1995; Shamma and Athans, 1990; Hyde and Glover, 1993), con diferentes opciones técnicas en cada uno de ellos (Rugh and Shamma, 2000; Leith and Leithead, 2000). Estos cuatro pasos son:

- calcular un modelo lineal de parámetros variables de la planta,
- diseñar controladores mediante métodos lineales,
- realizar la planificación de ganancia propiamente dicha, y
- evaluar el rendimiento del control diseñado.

Para obtener un **modelo lineal de parámetros variables** LPV (linear parameter-varying), la opción más común está basada en la linealización del *Jacobiano* de la planta no lineal en el entorno de una familia puntos de operación (ver anexo A), que consiste en aplicar la teoría del desarrollo en serie en torno a esos puntos, olvidando los términos de orden superior (Leith and Leithead, 2000). Lo más habitual es que los puntos de operación sean puntos de equilibrio (Hunt and Johansen, 1997), pero en cualquier caso se llega a una familia parametrizada de plantas linealizadas y es la base de lo que se llama planificación por linealización. Generalmente, la parametrización corresponde a valores fijos de las variables de planificación o funciones de ellas, que son variables internas de la planta y/o señales exógenas. La segunda opción, mucho más novedosa y que comenzó a desarrollarse en la pasada década (Shamma and Athans, 1991), es la planificación *quasi-LPV* (quasi linear parameter-varying), en la que la dinámica de la planta se reescribe para disfrazar las no linealidades como parámetros variables en el tiempo que serán usados como variables de planificación. Dependiendo de la estructura de la planta, la planificación quasi-LPV puede no involucrar ninguna linealización por Jacobiano de la dinámica del sistema. El concepto es, en la mayoría de métodos de planificación quasi-LPV, totalmente distinto al caso de modelos lineales, ya que con esta nueva técnica se puede desarrollar directamente el controlador mediante medidas de rendimiento basadas en normas y sin emplear los métodos de diseño de controladores lineales invariantes con el tiempo. Como regla general, cuanto mayor sea el número de modelos locales elegidos, más sencillo será encontrar un controlador que funcione adecuadamente para cada uno de los regímenes. Sin embargo, esta propiedad se enfrenta a la denominada maldición de la dimensionalidad: una partición uniforme lleva a que el número de particiones crezca de forma exponencial con el número de variables.

El segundo paso para el diseño de un controlador por planificación de ganancia consiste en usar métodos lineales para diseñar **controladores lineales** adecuados al modelo lineal de parámetros variables diseñado, independientemente del método que se haya empleado para ello. Este proceso de diseño puede llevar directamente a una familia de controladores lineales correspondientes a la planta lineal parametrizada, o puede ser un proceso de interpolación que lleve a una familia de controladores lineales partiendo

de un conjunto de controladores diseñados para valores aislados de las variables de planificación.

La **planificación de ganancia** del tercer paso consiste en implementar la familia de controladores lineales de forma que los coeficientes de los controladores (ganancias) varíen (sean planificados) de acuerdo con los valores actualizados de las variables de planificación, si no se ha hecho de una forma integrada con el diseño de controladores lineales del paso anterior. En este tercer paso, dependiendo de si se ha optado por una representación quasi-LPV o por una basada en el Jacobiano como opción de representación del sistema, las opciones a seguir son muy distintas.

El cuarto y último paso consiste en asegurar un adecuado **rendimiento**. En el mejor de los casos, cuando son tenidos en cuenta criterios de rendimiento en el proceso de diseño, este estudio puede ser relativamente sencillo. Sin embargo, lo más habitual es que solo se pueda realizar un estudio analítico a nivel local de las propiedades de estabilidad y rendimiento del controlador por planificación de ganancia, mientras que la evaluación de su rendimiento no local está basado en simulaciones.

B.3. Planificación de ganancia desde el diseño de controladores con métodos lineales.

El diseño de un controlador por planificación de ganancia, consiste en diseñar una familia de controladores lineales correspondiente a la familia de modelos lineales obtenida con alguno de los métodos descritos en el anexo A. El resultado es una familia de controladores parametrizada con la misma variable que la familia de modelos del sistema.

En general, existen dos opciones de trabajo para desarrollar el controlador no lineal para el sistema (Rugh and Shamma, 2000):

- interpolar las linealizaciones que existen en los puntos de operación y diseñar un controlador apropiado para cada una de esas interpolaciones
- diseñar controladores en las linealizaciones e interpolar éstos posteriormente

La primera de las opciones es el mejor de los casos y consistiría en diseñar una familia de controladores por medio de algún método de diseño particular, como LPV (Shamma and Athans, 1991) (linear parameter-varying, basados en funciones de Lyapunov) o LFT (Apkarian and Gahinet, 1995) (linear fractional transformation, que emplea la teoría de pequeña ganancia, ver B.3.4), donde se tiene en cuenta en todo momento la variabilidad con el tiempo de todos los parámetros del sistema. Sin embargo, estos métodos difieren bastante de lo que tradicionalmente se entiende como planificación de ganancia, ya que se diseña directamente el controlador en lugar de diseñar una familia de controladores por métodos lineales invariantes en el tiempo. Además, en lugar de proporcionar métodos genéricos de diseño, se basan en índices de rendimiento a determinar en cada caso.

La segunda opción es lo que se toma en la mayoría de los casos de diseño de control (Rugh and Shamma, 2000; Leith and Leithead, 2000) por planificación de ganancia y

consiste en tomar un conjunto de *puntos de diseño* correspondientes a una selección de puntos de operación, con la única restricción (por simplificar) de que todos los diseños tengan la misma dimensión. Además, este método es genérico y aplicable, por ejemplo, a los modelos identificados por técnicas de agrupamiento. La idea subyacente de este enfoque de la planificación de ganancia consiste en tratar la variable de planificación como un parámetro a lo largo de todo el proceso de diseño de controladores lineales, pasando a ser una señal de entrada variable en el tiempo cuando se implemente el controlador definitivo. La familia de controladores lineales se calcula interpolando los diseños de los puntos elegidos.

Los modelos LPV de mayor interés en el caso de modelos identificados por técnicas de agrupamiento, son las familias de linealizaciones clásicas en el entorno de puntos de equilibrio (A.7), las familias de linealizaciones en torno a puntos de operación (A.15) y las familias de linealización basadas en la velocidad (A.18) y, más concretamente, sus versiones discretas (A.16) y (A.21), respectivamente.

B.3.1. Controladores basados en modelos LPV del sistema en puntos de equilibrio

Para el desarrollo de estos controladores se toma (A.7) como modelo del sistema, el cual se definió en el punto relativo al desarrollo de modelos LPV de familias de linealizaciones por Jacobiano en torno a diferentes puntos de equilibrio de un sistema.

Diseño de controladores

Lo habitual es diseñar un controlador para cada uno de los puntos de equilibrio de la forma (x_e, u_e) tales que $f(x_e, u_e) = 0$, parametrizados con la variable de planificación σ y que constituyen la familia de equilibrio (A.6) de la planta en el conjunto S . Esto es debido a que para cada valor de la variable de planificación en la linealización (σ_i tal que $i \in [1, c]$, siendo c el número de puntos de equilibrio seleccionados), el sistema (A.7) describe el comportamiento local de forma lineal para la planta no lineal en el entorno del equilibrio correspondiente. En el caso de la identificación de modelos mediante técnicas de agrupamiento, esto tendrá sentido siempre que los prototipos identificados tengan la característica de pertenecer al conjunto S .

Para una adecuada entrada e al controlador (con valor e_e en el equilibrio, correspondiente a x_e), el resultado es un conjunto indexado ($i \in [1, c]$) de controladores lineales:

$$\begin{aligned}\delta\dot{x} &= H_i\delta x + E_i\delta e, \\ \delta u &= J_i\delta x + K_i\delta e,\end{aligned}\tag{B.1}$$

cada uno de los cuales asegura un funcionamiento adecuado en bucle cerrado cuando se emplea con la linealización de la planta que le corresponde. Se asume que las dimensiones de las variables son las mismas para todos los controladores (para simplificar la notación) y las variables, por supuesto, son variaciones en el entorno del punto de equilibrio correspondiente (por ejemplo, $\delta e = e - e_e(\sigma_i)$). Hay que resaltar que σ_i se asume constante en el diseño del controlador.

Para conseguir, en base a un conjunto de controladores lineales (B.1) en los puntos de equilibrio, un controlador lineal único de parámetros variables parametrizado con σ como:

$$\begin{aligned}\delta\dot{x} &= H(\sigma)\delta x + E(\sigma)\delta e, \\ \delta u &= J(\sigma)\delta x + K(\sigma)\delta e,\end{aligned}\tag{B.2}$$

hay muchas opciones diferentes (Rugh and Shamma, 2000). Por ejemplo, la familia de equilibrio puede ser dividida en secciones disjuntas $S = S_1 \cup \dots \cup S_c$ y que los coeficientes del controlador varíen con la sencilla regla: $\sigma \in S_i$ implica el uso de los coeficientes H_i, E_i, J_i, K_i . Esto tiene la ventaja de la simplicidad en la definición de la familia de controladores, pero la desventaja de las discontinuidades (saltos) en los coeficientes del controlador, así como un posible comportamiento de *chattering*.

Normalmente, los controladores indexados se interpolan con respecto al parámetro σ de forma suave o, al menos, continua. Una opción es interpolar los coeficientes del controlador en el espacio de estados. Otra posibilidad es interpolar alguna característica de las representaciones entrada-salida del conjunto de controladores indexados, por ejemplo la interpolación de coeficientes de la función de transferencia.

Gran parte de la literatura sobre planificación de ganancia por linealización asume que está disponible un controlador interpolado de forma continua e ignora la necesidad de interpolar controladores en el caso de puntos de diseño aislados. Sin embargo, se pueden encontrar varias propuestas en este sentido. Entre ellas se puede destacar:

- interpolación de funciones de transferencia del controlador por interpolación lineal de polos, ceros y ganancias (Nichols *et al.*, 1993),
- interpolación de controladores H_∞ por interpolación lineal de las soluciones de las ecuaciones de Riccati (Reichert, 1992),
- interpolación lineal de los coeficientes de las matrices en el espacio de estados de controladores balanceados (Kellet, 1991),
- interpolación lineal de ganancias de observadores y estados (Hyde and Glover, 1993), y
- combinación de controladores (Kelly and Evers, 1997).

Todas estas posibilidades tienen la atracción de la intuición y funcionan correctamente en la aplicación para la que fueron diseñadas. Por otra parte, cada propuesta puede generar controladores no estables en equilibrios intermedios, una situación que puede llevar a inestabilidad en el sistema de planificación de ganancia en bucle cerrado (Rugh and Shamma, 2000). Lo que se suele hacer es incrementar el número de miembros de la familia de controladores empleada hasta que se obtienen las características deseadas en los puntos interpolados, ya que hay muy pocos resultados teóricos al respecto de la interpolación lineal de controladores.

Además de estos métodos ad-hoc, aunque son los más usados en la práctica, se han propuesto algunos métodos justificados teóricamente, como:

- un algoritmo para la interpolación lineal de las ganancias por realimentación del estado para posicionamiento de autovalores (Shahruz and Behtash, 1992),
- interpolación lineal dinámica de controladores satisfaciendo requerimientos de estabilización lineal (el controlador interpolado estabiliza la planta linealizada para cada $\sigma \in S$) (Stilwell and Rugh, 1999; Stilwell and Rugh, 2000).

En estos casos se demuestra que si se puede calcular una cota de $\dot{\sigma}(t)$ se garantizan propiedades de estabilidad local para el sistema no lineal de planificación de ganancia.

Planificación de ganancia

El último y definitivo paso consiste en implementar un controlador no lineal basado en la familia de controladores lineales (B.2), obteniendo un único controlador como:

$$\begin{aligned} \dot{x} &= f^c(x, e), \\ u &= g^c(x, e). \end{aligned} \tag{B.3}$$

Este paso es de gran importancia ya que la elección de la realización definitiva del controlador no lineal puede influir mucho en el comportamiento en bucle cerrado (Leith and Leithead, 1996).

Conviene recordar que, dados los modelos en los que se basa la planificación de ganancia clásica, para los controladores definitivos que se puedan implementar solo se emplea la información referente a la dinámica de la planta en el equilibrio. Por supuesto, se han desarrollado otros métodos que explotan información adicional, como la dinámica de la planta en puntos fuera del equilibrio, lo que lleva a controladores que alcanzan mejor rendimiento pero que se escapan del concepto clásico de planificación de ganancia. Este tipo de controladores se verán en apartados sucesivos.

En el caso que nos ocupa, el problema de la obtención de un controlador no lineal por planificación de ganancia se puede abordar de tres formas: con el método clásico, aplicando el principio de equivalencia local, o mediante el principio de equivalencia local extendida.

El **método clásico** consiste, en primer lugar, en implementar sobre (B.1) las transformaciones de entrada y salida ($\delta e = e - e_e(\sigma_i)$, $\delta u = u - u_e(\sigma_i)$). Hay que resaltar que este paso, a menudo, está implícito. Habitualmente, la entrada del controlador $e = y - y_{ref}$ es nula (o cercana a cero) en el equilibrio (es decir, $e_e(\sigma_i) = 0$ y $\delta e = e$) y, o bien la planta se muestra como un integrador puro (con lo que $u_e(\sigma_i) = 0$), o bien cada controlador lineal contiene una acción integral que genera implícitamente $u_e(\sigma_i)$ con la acción del bucle de realimentación (Aström and Wittenmark, 1995; Shamma and Athans, 1990). El segundo paso consiste en sustituir σ por σ_i en la familia de controladores locales lineales (B.2), con lo que se obtiene un controlador no lineal. Hay que resaltar que la variable de planificación no es necesario que sea continua y su selección depende mucho del conocimiento del comportamiento físico del sistema (Aström and Wittenmark, 1995).

El **principio de equivalencia lineal local** se basa en la demostración (Khalil, 1996) de que si una planta no lineal y un controlador no lineal tienen ambos el equilibrio

en el origen, entonces el correspondiente sistema en bucle cerrado tendrá un equilibrio en el origen. Además, la linealización del sistema en bucle cerrado alrededor del origen podrá ser calculada tomando el sistema en bucle cerrado formado por la planta linealizada y el controlador linealizado. El mismo resultado se mantiene en puntos de equilibrio diferentes al origen si el equilibrio del controlador coincide con el de la planta (Rugh, 1983). Así pues, el objetivo de la planificación por linealización deberá ser que la *familia de equilibrio del controlador* coincida con la de la planta, es decir, que el sistema en bucle cerrado mantenga los puntos de equilibrio y que, en dichos puntos, la linealización de la realización no lineal del controlador coincida con la familia de controladores lineales. Esto, evidentemente, no era necesario con el método clásico.

A efectos prácticos (Wang and Rugh, 1987b; Wang and Rugh, 1987a) la aplicación del principio de equivalencia lineal local consiste en, tras sustituir $\sigma(t)$ en (B.2) por una función $\sigma(t) = \sigma(x(t), e(t))$ que muestre su dependencia con otras variables (pueden ser la indicadas o cualquier otra disponible), conseguir un controlador tal que:

- sea capaz de proporcionar la entrada constante adecuada al sistema para que se mantenga en los puntos de equilibrio, es decir, que para que los equilibrios de planta y controlador coincidan este último deberá cumplir (si para las condiciones de equilibrio de la planta, existe una función $x_e(\sigma)$) que:

$$\begin{aligned} 0 &= f^c(x_e(\sigma), e(\sigma)), \\ u_e(\sigma) &= g^c(x_e(\sigma), e(\sigma)), \end{aligned} \tag{B.4}$$

- la linealización del controlador no lineal coincida con los controladores lineales diseñados, por ejemplo, haciendo:

$$\begin{aligned} \frac{\partial f^c}{\partial x^c}(x_e(\sigma), e(\sigma)) &= H(\sigma), \\ \frac{\partial g^c}{\partial u}(x_e(\sigma), e(\sigma)) &= K(\sigma). \end{aligned} \tag{B.5}$$

La forma más sencilla de hacerlo es especificar, en términos de los coeficientes de (B.2), la familia de controladores

$$\begin{aligned} \dot{x} &= H(\sigma)[x - x_e(\sigma)] + E(\sigma)[e - e_e(\sigma)], \\ u &= J(\sigma)[x - x_e(\sigma)] + K(\sigma)[e - e_e(\sigma)] + u_e(\sigma), \end{aligned} \tag{B.6}$$

que satisface la condición de equilibrio (B.4) y, tomando σ como constante y diferenciando, cumple con los requisitos de linealización planteados en (B.5). Por supuesto, hay otras muchos controladores del tipo (B.3) que satisfacen (B.4) y (B.5). Sin embargo, (B.6) es lo habitual en aplicaciones.

La implementación del controlador por planificación de ganancia basado en (B.6) es, sencillamente, reemplazar el parámetro $\sigma(t)$ en (B.2) por su dependencia con otras variables de acuerdo con $\sigma(t) = \sigma(x(t), e(t))$, con lo que (B.6) lleva a:

$$\begin{aligned}
\dot{x} &= H(\sigma(x(t), e(t)))[x - x_e(\sigma(x(t), e(t)))] \\
&\quad + E(\sigma(x(t), e(t)))[e - e_e(\sigma(x(t), e(t)))]], \\
u &= J(\sigma(x(t), e(t)))[x - x_e(\sigma(x(t), e(t)))] \\
&\quad + K(\sigma(x(t), e(t)))[e - e_e(\sigma(x(t), e(t)))] + u_e(\sigma(x(t), e(t))),
\end{aligned} \tag{B.7}$$

cuya versión discreta será:

$$\begin{aligned}
x_{k+1} &= H(\sigma(x_k, e_k))[x_k - x_e(\sigma(x_k, e_k))] \\
&\quad + E(\sigma(x_k, e_k))[e_k - e_e(\sigma(x_k, e_k))], \\
u_k &= J(\sigma(x_k, e_k))[x_k - x_e(\sigma(x_k, e_k))] \\
&\quad + K(\sigma(x_k, e_k))[e_k - e_e(\sigma(x_k, e_k))] + u_e(\sigma(x_k, e_k)),
\end{aligned} \tag{B.8}$$

que en el caso de sistemas NARX como (A.9) llegamos al controlador:

$$\begin{aligned}
u_k &= J(\sigma(y_k, e_k))[y_k - y_e(\sigma(y_k, e_k))] \\
&\quad + K(\sigma(y_k, e_k))[e_k - e_e(\sigma(y_k, e_k))] + u_e(\sigma(y_k, e_k)).
\end{aligned} \tag{B.9}$$

La última forma para obtener un controlador no lineal por planificación de ganancia es aplicar el **principio de equivalencia lineal local extendida**, el cual da una guía (Leith and Leithead, 1998a) para seleccionar una apropiada realización, de entre las infinitas que pueden existir (Leith and Leithead, 1996), del controlador no lineal que cumple con el principio de equivalencia lineal local. Para ello se seleccionará aquel controlador en el que la vecindad de sus linealizaciones sean una aproximación más precisa de la dinámica del controlador. De esta forma, la realización del controlador que se consigue al aplicar el principio de equivalencia lineal local extendida es, en cierto sentido, el mejor que se puede conseguir con las limitaciones impuestas por el conocimiento limitado del sistema tan solo en el equilibrio. A pesar de todo, los requerimientos formales de este principio son muy duros y difícilmente aplicables en un caso general (Leith and Leithead, 2000).

Acoplamiento oculto

Para analizar el comportamiento del sistema de planificación de ganancia del bucle cerrado correspondiente a (B.7), un primer paso puede ser linealizar en el entorno de los puntos de equilibrio para analizar el comportamiento local. Esto se puede hacer de forma separada para el controlador y la planta. Sin embargo, la linealización de (B.7) no coincide con la de (B.6), ya que no aparece explícitamente el parámetro artificial σ en el controlador. Esta diferencia puede llevar a que aparezcan nuevos términos en la linealización de (B.7) que no aparecían en la linealización de (B.6).

La linealización de (B.7) en el entorno de un equilibrio especificado por σ da lugar, tras una cuidadosa diferenciación y evaluación (Rugh and Shamma, 2000), a la expresión

$$\begin{aligned}
\delta \dot{x} &= H(\sigma)\delta x + E(\sigma)\delta e \\
&\quad - \left[H(\sigma)\frac{\partial x_e(\sigma)}{\partial \sigma} + E(\sigma)\frac{\partial e_e(\sigma)}{\partial \sigma} \right] \\
&\quad \times \left[\frac{\partial \sigma}{\partial x}(x_e(\sigma), e_e(\sigma))\delta x + \frac{\partial \sigma}{\partial e}(x_e(\sigma), e_e(\sigma))\delta e \right], \\
\delta \dot{u} &= J(\sigma)\delta x + K(\sigma)\delta e \\
&\quad + \left[\frac{\partial u_e(\sigma)}{\partial \sigma} - J(\sigma)\frac{\partial x_e(\sigma)}{\partial \sigma} - K(\sigma)\frac{\partial e_e(\sigma)}{\partial \sigma} \right] \\
&\quad \times \left[\frac{\partial \sigma}{\partial x}(x_e(\sigma), e_e(\sigma))\delta x + \frac{\partial \sigma}{\partial e}(x_e(\sigma), e_e(\sigma))\delta e \right].
\end{aligned} \tag{B.10}$$

Los términos adicionales que aparecen al comparar (B.10) con (B.2) actúan como realimentaciones adicionales (en el caso de las variables de estado) o como perturbaciones externas. A éstos términos se les llama *términos de acoplamiento oculto*. Durante muchos años, los usuarios de estos métodos conocían este fenómeno y lo han llamado *ganancias ocultas* o *bucles ocultos*, aunque no han sido analizados en un contexto general hasta hace poco tiempo (Rugh and Shamma, 2000).

La forma de conseguir un controlador por planificación de ganancia como (B.7) sin términos de acoplamientos ocultos es, dada la familia de controladores (B.2), encontrar una familia de equilibrio del controlador $x_e(\sigma)$ que satisfaga (Rugh, 1991; Nichols *et al.*, 1993; Lawrence and Rugh, 1995)

$$\begin{aligned}
H(\sigma)\frac{\partial x_e(\sigma)}{\partial \sigma} + E(\sigma)\frac{\partial e_e(\sigma)}{\partial \sigma} &= 0 \\
\frac{\partial u_e(\sigma)}{\partial \sigma} - J(\sigma)\frac{\partial x_e(\sigma)}{\partial \sigma} - K(\sigma)\frac{\partial e_e(\sigma)}{\partial \sigma} &= 0,
\end{aligned} \tag{B.11}$$

para $\sigma \in S$. Las condiciones que (B.11) da para diseñar controladores sin que aparezcan los términos de acoplamiento oculto son, sin embargo, muy restrictivas y solo es aplicable a casos sencillos (Rugh and Shamma, 2000).

B.3.2. Controladores basados en modelos LPV del sistema en puntos de operación

A pesar de la gran utilización de los controladores por planificación de ganancia clásica, se desprende de lo comentado en la sección anterior que estas técnicas todavía están poco desarrolladas. En particular, la limitación más clara de estas técnicas es que tan solo explotan el comportamiento del sistema en la vecindad de puntos de equilibrio, lo que generalmente obliga a la imposición del requerimiento de variación lenta del sistema para asegurar que el estado permanece cerca del equilibrio, el cual es adicional al requerimiento de variación lenta asociado con el cambio de la dinámica linealizada cuando el sistema se mueve de la vecindad de un punto de equilibrio a otro.

Sin embargo, para cumplir con las cada vez más exigentes especificaciones de funcionamiento de los sistemas, a los controladores por planificación de ganancia se les suele requerir operar tanto durante las transiciones entre puntos de equilibrio (que puede llevar de forma transitoria al sistema a operar lejos del equilibrio) como lejos del equilibrio de forma continuada. Diferentes aproximaciones (Hunt and Johansen, 1997) intentan relajar las condiciones de operación en el entorno del punto de equilibrio aunque están muy ligadas a la filosofía de diseño de la planificación de ganancia clásica.

En este caso se considera el desarrollo de un controlador para cada uno de los puntos de operación $(x_o, u_o) \in R^{n+m}$ en el entorno de los cuales se ha aproximado la dinámica del sistema no lineal. Todos estos puntos de operación ψ , definidos según (A.10), deberán pertenecer a un espacio de operación Ψ como (A.14). Dichos puntos de operación se parametrizan en función de una variable de planificación $\phi \in \Phi$ a la que se le llama régimen de operación y al conjunto Φ de todos ellos conjunto de operación. En el caso que nos ocupa, dado que como se ha comentado los modelos de la planta con los que se trabaja están desarrollados en el marco de los modelos borrosos o neuronales, la representación del sistema de la que se dispone es una representación combinación de múltiples modelos como los de Takagi-Sugeno o las redes de modelos locales (equivalentes bajo ciertas condiciones (Hunt *et al.*, 1996)), llevando a sistemas aproximadores universales de (A.1) de la forma:

$$\begin{aligned} \dot{x} &= \sum_{i=1}^c \mu_i(\phi) f_i(x, u), \\ u &= \sum_{i=1}^c \mu_i(\phi) g_i(x, u), \end{aligned} \tag{B.12}$$

en los que μ representa la función de validez (o pertenencia) que se suele normalizar de la forma $\sum_i \mu_i(\phi) = 1$. Con estos modelos se puede llegar a aproximar el sistema no lineal (A.1) por sus planos tangentes en el punto (x_o, u_o) , con lo que para cada valor de la variable de planificación en la linealización (ϕ_i tal que $i \in [1, c]$, siendo c el número de puntos de operación seleccionados), el sistema (B.12) describe el comportamiento local de forma lineal para la planta no lineal en el entorno del punto correspondiente. En el caso de la identificación de modelos mediante técnicas de agrupamiento, esto será aplicable siempre que los modelos identificados representen la dinámica de la planta en el entorno de los puntos elegidos. Este tipo de modelo se definió en el punto relativo al desarrollo de modelos LPV de familias de linealizaciones en torno a diferentes puntos de operación (no necesariamente de equilibrio) de un sistema, considerando modelos afines (A.15).

El resultado será, en lo que respecta al control, un conjunto indexado ($i \in [1, c]$) de controladores lineales como el descrito en (B.1), cada uno de los ellos diseñado para asegurar un funcionamiento adecuado en bucle cerrado cuando se emplea con la linealización de la planta que le corresponde. Para que el conjunto indexado de controladores lineales sea aplicable, se ha de conseguir en base a ellos un controlador lineal único de parámetros variables parametrizado con ϕ como:

$$\begin{aligned}\dot{x} &= H(\phi)\delta x + E(\phi)\delta e, \\ u &= J(\phi)\delta x + K(\phi)\delta e + u_o(\phi).\end{aligned}\tag{B.13}$$

Para la combinación de los controladores se puede usar (y de hecho se usa) la misma regla de variación que dan las funciones de pertenencia (validez) para el modelo de la planta (B.12), ya que es de suponer que de algún modo capturan la dinámica de variación entre modelos para la planta, quedando:

$$\begin{aligned}\dot{x} &= \sum_i^c [\mu_i(\phi)H_i]\delta x + \sum_i^c [\mu_i(\phi)E_i]\delta e, \\ u &= \sum_i^c [\mu_i(\phi)J_i]\delta x + \sum_i^c [\mu_i(\phi)K_i]\delta e + \sum_i^c [\mu_i(\phi)u_{oi}(\phi)],\end{aligned}\tag{B.14}$$

o su equivalente discreto:

$$\begin{aligned}x_{k+1} &= \sum_i^c [\mu_i(\phi)H_i]\delta x_k + \sum_i^c [\mu_i(\phi)E_i]\delta e_k, \\ u &= \sum_i^c [\mu_i(\phi)J_i]\delta x_k + \sum_i^c [\mu_i(\phi)K_i]\delta e_k + \sum_i^c [\mu_i(\phi)u_{oi}(\phi)],\end{aligned}\tag{B.15}$$

que queda como sigue al tomar el modelo NARX (A.17):

$$u = \sum_i^c [\mu_i(\phi)J_i]\delta y_k + \sum_i^c [\mu_i(\phi)K_i]\delta e_k + \sum_i^c [\mu_i(\phi)u_{oi}(\phi)].\tag{B.16}$$

Sin embargo, los modelos como (B.12) son la combinación de los modelos locales que se desprenden de (A.1), pero la relación entre las soluciones de ambos sistemas no es directa. El desarrollo en serie de (B.12) en el entorno del punto de operación aproxima perfectamente a (A.1), pero contiene términos cruzados de las derivadas de las funciones de pertenencia (Hunt and Johansen, 1997), con lo que en aquellas zonas en las que la derivada de alguna de las funciones de pertenencia tome valores elevados, la dinámica del desarrollo en serie estará muy relacionada con las funciones de pertenencia y poco con el modelo local.

De forma general, los métodos basados en las representaciones de Takagi-Sugeno o redes de modelos locales no tienen por qué seguir la filosofía de división del problema en un conjunto de subproblemas lineales, sino que los modelos obtenidos son una combinación de múltiples modelos (solo válidos en el punto de operación) cada uno de los cuales no tiene por qué responder a un comportamiento lineal del sistema en el entorno de un punto de operación, sino que es la combinación de todos ellos la que aproxima el modelo al sistema real (Díez *et al.*, 2001). La interpolación se convierte en este caso en la parte más importante de este método, ya que no hay modelos que dominen

en cada región de operación sino que varios de ellos contribuyen a la descripción del funcionamiento del sistema. Esta interpretación se aleja de la idea de los métodos de planificación de ganancia clásicos y los métodos de diseño se asemejan más a los que trabajan con representaciones quasi-LPV y se basan en la resolución de sistemas LMI (Linear Matrix Inequalities) (Tanaka and Sano, 1994; Wang *et al.*, 1996; Tanaka *et al.*, 1998; Kiriakidis, 1999).

Si lo que se pretende es mantener la continuidad con los métodos lineales de la planificación de ganancia clásica se deberán conseguir modelos que realmente describan el comportamiento local de forma lineal para la planta no lineal en un entorno del punto correspondiente y considerar representaciones de la planta con modelos locales lineales (Tanaka and Sano, 1994; Wang *et al.*, 1996; Tanaka *et al.*, 1998; Kiriakidis, 1999), lo que da lugar a representaciones en las que los modelos locales son no homogéneos, como cuando se consideran modelos afines y resulta el modelo (A.15) con el que se suele trabajar en la literatura. A pesar de todo, este tipo de modelos no siguen el principio de superposición y, por tanto, no son lineales (Hunt and Johansen, 1997).

Para el diseño de los controladores no se puede obviar la existencia de los términos no homogéneos aunque el controlador posea acción integral (Leith and Leithead, 2000), ya que se pueden encontrar ejemplos en los que un controlador así diseñado lleve a sistemas inestables en bucle cerrado. Lo que sí que se puede hacer es evaluar la influencia de los términos que aparecen debido a la no linealidad de los modelos elegidos, pudiendo llegar a conseguir una función temporal $\epsilon(t)$ en la que se mide el alejamiento con respecto al modelo lineal (Hunt and Johansen, 1997). El valor de dicha función se puede mantener relativamente pequeño cuando hay un suficiente número de modelos y la variación del punto de operación es lo suficientemente lenta y $\epsilon(t)$ se puede emplear como una cota superior en el diseño de controladores con técnicas de control robusto.

Si se construyen los modelos del sistema con un rango suficientemente amplio en el entorno del punto de operación, es decir, la planta realmente responde a un sistema linealizabile, el valor de $\epsilon(t)$ se reduce de forma considerable (Díez and Previdi, 2001). Además, si se dispone de cierta libertad en la selección de las formas de las funciones de pertenencia, su forma suave, plana en la cercanía del punto de operación y su variación temporal lenta ayudan a mantener $\epsilon(t)$ en valores pequeños. La identificación con una técnica de agrupamiento como AFCRC, en la que la forma de las funciones de pertenencia se incluye en su índice de coste, ayuda a encontrar (si existe) el modelo del sistema más adecuado en este sentido (Díez *et al.*, 2001).

En definitiva, la aplicabilidad de la filosofía clásica de control por planificación de ganancia en los controladores basados en modelos LPV del sistema en diferentes puntos de operación se restringe a situaciones en las que hay una combinación de modelos mínima o el sistema es lo suficientemente lento como para que la influencia de los términos cruzados en los que aparecen las derivadas de las funciones de pertenencia es pequeña (Shorten *et al.*, 1999; Leith and Leithead, 1999a). Por tanto, se puede decir que la eficiencia del controlador desarrollado es muy dependiente de las características del sistema así como de su modelo disponible y hay que recordar que, a todas estas consideraciones, hay que añadir las condiciones de variación lenta asociadas con los métodos de planificación de ganancia clásicos.

B.3.3. Controladores basados en modelos LPV del sistema linealizados por velocidad

La limitación del desarrollo de modelos clásicos al entorno de puntos de equilibrio, también se puede superar mediante los modelos de linealización basada en velocidad (A.18). De este modo se puede asociar un modelo lineal del sistema en cualquier punto de operación del sistema no lineal, sea de equilibrio o no. La linealización basada en la velocidad de la combinación en cascada de dos sistemas no lineales es simplemente la combinación de las correspondientes linealizaciones basadas en la velocidad de los dos sistemas no lineales (Leith and Leithead, 1998c). Igualmente ocurre en la combinación en paralelo o en bucle cerrado de dos sistemas no lineales. Estas propiedades sugieren un método generalizado de diseño de reguladores por planificación de ganancia.

Una vez determinados los modelos del tipo (A.18) para cada uno de los puntos de operación que se hayan elegido, pueden parametrizarse en un único modelo (A.20). Con esta familia de linealizaciones de la planta basada en la velocidad, se debe determinar la familia de linealizaciones del controlador basado en la velocidad tal que la familia en bucle cerrado alcance los requerimientos de diseño deseados. Como cada miembro de los que componen la familia de la planta es lineal, se pueden emplear métodos de diseño lineales para el diseño de cada uno de los controladores.

Para cada uno de los valores de ρ en (A.20) podemos tener un modelo linealizado del sistema, con lo que la familia de linealizaciones por velocidad puede tener un número infinito de miembros. Por supuesto, es más atractivo considerar en el proceso de diseño de controladores un pequeño conjunto finito de miembros de la familia de linealizaciones de la planta basada en la velocidad al igual que se hace con los métodos clásicos de planificación de ganancia que ya se han descrito. En la planificación de ganancia tradicional esto se resuelve interpolando entre un número limitado de controladores y algo similar se podría aplicar en el diseño de los controladores basados en las linealizaciones por velocidad. Desafortunadamente, tal y como se ha visto, los métodos de interpolación empleados en la planificación de ganancia convencional son bastante arbitrarios. Una alternativa es considerar una aproximación a la familia de linealización basada en la velocidad de la planta en la que se interpolen un número finito de linealizaciones basadas en la velocidad (modelos locales). Los modelos locales y las funciones de interpolación se seleccionan para asegurar que la familia de aproximación sea lo suficientemente precisa, de la misma forma que las aproximaciones desarrolladas en el marco de la lógica borrosa y las redes neuronales que se han comentado en el apartado anterior. Con todo ello, el modelo global consistirá en la ponderación de un conjunto indexado ($i \in [1, c]$) de modelos locales (uno de ellos para cada punto de operación (x_i, u_i) elegido) como:

$$\begin{aligned}
\dot{x} &= v, \\
\dot{v} &= \sum_i^c \mu_i(\rho) [(\nabla_x f(x_o(\rho), u_o(\rho)) + \nabla_x d_f(x_o(\rho), u_o(\rho)))]v + \\
&\quad + \sum_i^c \mu_i(\rho) [(\nabla_u f(x_o(\rho), u_o(\rho)) + \nabla_u d_f(x_o(\rho), u_o(\rho)))]\dot{u}, \\
\dot{y} &= \sum_i^c \mu_i(\rho) [(\nabla_x g(x_o(\rho), u_o(\rho)) + \nabla_x d_g(x_o(\rho), u_o(\rho)))]v + \\
&\quad + \sum_i^c \mu_i(\rho) [(\nabla_u g(x_o(\rho), u_o(\rho)) + \nabla_u d_g(x_o(\rho), u_o(\rho)))]\dot{u},
\end{aligned} \tag{B.17}$$

en la que μ sigue la misma definición que en (B.12) o (B.14). Ahora se podrá diseñar un controlador para cada uno de los modelos locales según:

$$\begin{aligned}
\dot{v} &= H_i v + E_i \dot{e}, \\
u &= J_i v + K_i \dot{e},
\end{aligned} \tag{B.18}$$

donde se asume que las derivadas del estado de la planta y del error se pueden medir o son calculables. El resultado definitivo será una familia de controladores linealizados por velocidad que se podrá obtener interpolando entre los controladores lineales. Para ello se usarán las mismas funciones de interpolación que en la aproximación de la planta (B.17), con lo que la familia de controladores linealizados por velocidad e interpolados reflejará las características de la planta:

$$\begin{aligned}
\dot{x} &= v, \\
\dot{v} &= \sum_i^c [\mu_i(\rho) H_i] v + \sum_i^c [\mu_i(\rho) E_i] \dot{e}, \\
u &= \sum_i^c [\mu_i(\rho) J_i] v + \sum_i^c [\mu_i(\rho) K_i] \dot{e},
\end{aligned} \tag{B.19}$$

e igualmente en su versión discreta:

$$\begin{aligned}
\Delta v_k &= v_k - v_{k-1}, \\
\Delta v_{k+1} &= \sum_i^c [\mu_i(\rho) H_i] \Delta v_k + \sum_i^c [\mu_i(\rho) E_i] \Delta e_k, \\
\Delta u_k &= \sum_i^c [\mu_i(\rho) J_i] \Delta v_k + \sum_i^c [\mu_i(\rho) K_i] \Delta e_k,
\end{aligned} \tag{B.20}$$

o la que parte del modelo NARX (A.23):

$$\begin{aligned}\Delta v_k &= v_k - v_{k-1}, \\ \Delta u_k &= \sum_i^c [\mu_i(\rho) J_i] \Delta v_k + \sum_i^c [\mu_i(\rho) K_i] \Delta e_k.\end{aligned}\tag{B.21}$$

En estos casos, en contraste con las representaciones de Takagi-Sugeno o redes de modelos locales, al desaparecer el término independiente (offset) se mantiene una relación directa (Leith and Leithead, 1999a) entre las características dinámicas de (B.17) y de (B.19).

El último paso del proceso de diseño (Leith and Leithead, 1998c; Leith and Leithead, 1999a; Leith and Leithead, 1999b) será la realización de un controlador no lineal desde la familia de controladores linealizados por velocidad e interpolados (B.19). El controlador en su forma basada en velocidad se puede obtener directamente desde la familia de controladores lineales simplemente permitiendo que la variables de planificación ρ varíe con el punto de operación. Por supuesto, hay que tener un cuidado especial con las realizaciones basadas en velocidad, pues hay que recordar que sus entradas son derivadas de variables. Además, debido a las operaciones de diferenciación e integración asociadas con los modelos locales en su representación por velocidad, el orden de estos modelos puede ser mayor que el de la representación directa (A.18).

La ventaja del proceso de diseño que se acaba de describir radica en que mantiene una filosofía de división del problema en subproblemas conservando la continuidad con los métodos lineales de diseño de controladores, sin restricciones al respecto de la técnica de diseño que se haya de emplear (los términos de offset han desaparecido). Además de las ventajas anteriores, compartidas con los métodos clásicos de diseño, la planificación de ganancia basada en la velocidad proporciona un marco de análisis y diseño con un único tipo de linealización. Adicionalmente, el controlador no lineal que resulta es válido en todos los posibles puntos de funcionamiento de la planta, no solo en la vecindad de los puntos de equilibrio (Leith and Leithead, 1999b).

Nota. Es interesante resaltar que para aquellas plantas no lineales que satisfacen el principio de equivalencia local lineal extendida, los métodos de planificación de ganancia clásico y basado en velocidad coinciden si se elige un realización adecuada del controlador (Leith and Leithead, 1998c).

B.3.4. Estabilidad y rendimiento

Al considerar los métodos de planificación de ganancia en los modelos de la planta solo las linealizaciones en algunos puntos de operación, son obviadas transiciones entre regímenes y linealidades de segundo orden. Las guías heurísticas que se suelen considerar simplemente establecen que mientras los términos obviados no sean muy significantes el comportamiento global puede esperarse que sea bueno, lo que se comprobará mediante simulaciones.

Sin embargo, la estabilidad y rendimiento de los sistemas no lineales diseñados por el método de planificación de ganancia basada en la velocidad puede analizarse mediante cualquier método de propósito general para sistemas no lineales como la teoría de

Lyapunov (aplicación vista en el apartado 4.3), o los métodos de pequeña ganancia o teoría de tiempo detenido que se comentan brevemente a continuación. Por supuesto, los resultados obtenidos son bastante conservadores y, generalmente, solo consisten en condiciones suficientes de estabilidad.

Estabilidad mediante la teoría del tiempo detenido. En el contexto de planificación de ganancia es muy usada la teoría de tiempo detenido (frozen-time) para establecer condiciones de estabilidad en sistemas lineales variantes con el tiempo. En concreto, puede verse (Desoer, 1969) que la estabilidad del sistema no lineal está garantizada (hereda las propiedades de estabilidad del peor de los casos, es decir, el peor comportamiento en los puntos de operación) si la variación temporal de la variable de planificación ρ es suficientemente lenta en algún sentido matemático (por ejemplo, que $\sup_{t>0} |\frac{d}{dt}\rho|$ sea lo suficientemente pequeño).

Todos los resultados de robustez que se puedan concluir desde la teoría del tiempo detenido llevarán a que un incremento de la robustez implica una disminución en el ratio de variación de ρ . El sistema lineal invariante solo heredará las propiedades del sistema en tiempo detenido si se permite una variación arbitrariamente lenta de la variable de planificación.

Estabilidad mediante el teorema de pequeña ganancia. El teorema de pequeña ganancia (Sandberg, 1964) establece que el sistema no lineal $y = H(u)$ que se esté estudiando será estable en bucle cerrado si la ganancia del sistema $\gamma_H = \sup_{t>0, u \neq 0} \frac{|H_t u_t|}{|u_t|}$ es menor que la unidad, es decir, que $\gamma_H < 1$. Además establece que, dada la entrada u al sistema, el error será $|e_t| = \frac{1}{1-\gamma_H} |u_t|$ y la salida $y_t = \frac{\gamma_H}{1-\gamma_H} |u_t|$.

El valor de γ_H debe evaluarse para todas las posibles entradas u del sistema, lo que conlleva dos importantes restricciones: algunos sistemas de interés práctico pueden no poseer estas ganancias y, aunque existan, pueden ser difíciles de evaluar.

Apéndice C

Algoritmos de agrupamiento

En este anexo se revisan las técnicas de agrupamiento existentes, comenzando con las técnicas más clásicas para seguir con las más apropiadas en la identificación de modelos borrosos como son los algoritmos de agrupamiento borroso. De entre los múltiples algoritmos de agrupamiento borroso existentes, se centrará esta revisión en aquellos con capacidades de detección de subespacios lineales dada su idoneidad para la identificación de modelos borrosos TS.

Posteriormente se muestran algoritmos que no son de interés en el marco de la Tesis pero de relevancia en la bibliografía, como las clases con prototipos no lineales o los robuustos.

Finalmente, se plantean diferentes opciones para la determinación del número de clases, paso previo en la mayoría de algoritmos de agrupamiento.

C.1. Algoritmos de agrupamiento clásico

En este punto se van a comentar los algoritmos de *agrupamiento clásico* (Duda and Hart, 1973; Maravall, 1993; Duda *et al.*, 2000) más utilizados, en sentido creciente de complejidad de cálculo.

C.1.1. Algoritmo de las distancias encadenadas (chain-map)

Es un algoritmo muy simple que no requiere ningún tipo de información a priori respecto a la distribución por clases de los objetos a clasificar. Aunque los resultados no pueden ser, para determinadas situaciones, los óptimos, es recomendable como procedimiento inicial para tantear la agrupación de los objetos.

Dados los vectores a clasificar z_1, z_2, \dots, z_N , se escoge uno de ellos al azar, por ejemplo el z_i , y a continuación se ordenan de manera que el siguiente vector de la cadena es el más próximo al anterior, quedando una sucesión:

$$z_i(0), z_i(1), z_i(2), \dots, z_i(k), z_i(k+1), \dots, z_i(N-1) \quad (\text{C.1})$$

en la que $z_i(1)$ será el vector más próximo al $z_i(0)$, $z_i(2)$ el más cercano a $z_i(1)$, y así sucesivamente. Obsérvese que se ha indicado explícitamente que la sucesión formada

depende del elemento inicial, de ahí que en (C.1) se mantenga el subíndice i perteneciente al primer elemento de la cadena.

El siguiente paso es calcular la sucesión de distancias euclídeas relativas:

$$d_1, d_2, \dots, d_{k+1}, \dots, d_{N-1} \quad (\text{C.2})$$

siendo:

$$\begin{aligned} d_1 &= \|z_i(1) - z_i(0)\| \\ d_2 &= \|z_i(2) - z_i(1)\| \\ &\dots \\ d_{N-1} &= \|z_i(N-1) - z_i(N-2)\| \end{aligned} \quad (\text{C.3})$$

A partir de su representación gráfica se pueden detectar fácilmente las clases, como se puede observar en el histograma de la figura C.1.

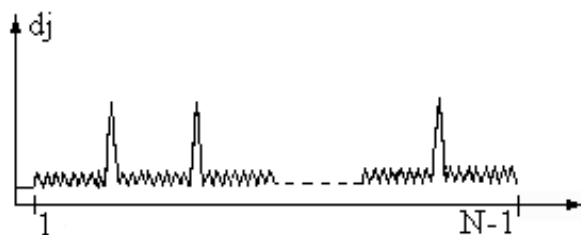


Figura C.1: Representación de un histograma distancia euclídea versus índice de la distancia.

La detección de clases se realiza mediante el salto significativo de la distancia euclídea. El parámetro más delicado del algoritmo es el umbral de detección de una nueva clase. Un umbral excesivamente bajo dará lugar a clases ficticias, mientras que un umbral alto tenderá a agrupar en una misma clase objetos de diferentes clases. En cuanto a la elección del primer elemento de la cadena, el algoritmo no es excesivamente sensible, salvo para distribuciones muy caprichosas.

C.1.2. Algoritmo max-min

Es un algoritmo heurístico que emplea como único elemento formal la distancia euclídea. No requiere ninguna información a priori del número de clases existentes. Partiendo del conjunto de muestras o vectores a agrupar, los pasos del algoritmo son:

- Se escoge al azar un elemento de los N disponibles, por ejemplo el z_1 , y se crea la primera clase A_1 .
- Se busca el elemento más alejado z_2 (tomando distancias euclídeas) y se forma la segunda clase A_2 .

- Se asigna a cada elemento sin clasificar la menor distancia a las dos clases ya definidas A_1 y A_2 . De las $N-2$ distancias mínimas obtenidas, se toma la máxima y si

$$d_{MAX} > f \times d(z_1, z_2), \quad 0 < f < 1$$

se crea una nueva clase A_3 .

- Se van creando nuevas clases hasta que no se cumpla la condición

$$d_{MAX} > f \times (\text{valor medio de las distancias entre prototipos}), \quad 0 < f < 1$$

- Se agrupan los restantes elementos a las clases cuyo prototipo esté más cercano.

El mayor inconveniente de este algoritmo es la elección de f , ya que el algoritmo es muy sensible a este coeficiente. La única forma de realizar una buena elección es ensayar con diferentes valores hasta que se produzca una elección correcta.

C.1.3. Algoritmo K-medias (K-means)

El nombre de este algoritmo hace referencia a que existen K clases diferentes, por lo que es necesario conocer a priori el número de clases existentes. El algoritmo ha sido ampliamente utilizado por su sencillez, eficiencia y robustez, siempre que se conozca el número de clases con exactitud.

Los pasos del algoritmo, dado un conjunto Z de N elementos a clasificar son:

- Se escogen K elementos al azar que forman las clases iniciales, siendo ellos, como únicos elementos, los centroides (c_i) de las mismas.
- Se asignan los restantes elementos a la clase A_i cuyo centro esté más cerca.
- Se recalcula el centro de cada clase como la media de los elementos que componen la clase, buscando minimizar el índice:

$$J = \sum_{i=1}^K \left(\sum_{j, z_j \in A_i} \|z_j - c_i\| \right) \quad (\text{C.4})$$

- Se vuelve al segundo paso hasta que la distribución sea estable (J no varíe), según una condición de finalización predefinida.

Este algoritmo es extremadamente sensible al parámetro K , ya que un valor superior al número real de clases dará lugar a clases ficticios y un valor de K inferior producirá menos clases de las reales. Se podría detectar una mala elección del parámetro K analizando las dispersiones estadísticas de las clases formadas, aunque en realidad esa dispersión también podría deberse a la naturaleza de las variables estudiadas.

C.1.4. Algoritmo ISODATA

El algoritmo ISODATA (Iterative Self-Organizing Data Analysis Techniques) está basado en el algoritmo K-means al que se le ha añadido conocimiento heurístico mediante parámetros y operaciones. Tiene una naturaleza altamente interactiva con el usuario, participando éste activamente durante la ejecución.

Los parámetros que maneja el algoritmo son:

- N_C : número de clases.
- K : número estimado, a priori, de clases.
- θ_N : número mínimo de miembros de una clase para constituirse como tal.
- θ_S : desviación típica máxima. Cuando la desviación típica de un grupo sea mayor a θ_S se dividirá la clase en dos.
- θ_C : distancia mínima entre clases. Si la distancia mínima entre dos clases es menor que θ_C se fusionarán las clases.
- L : número máximo de uniones en una iteración.
- I : número máximo de iteraciones.

Una vez definidos los parámetros que intervienen, el algoritmo ISODATA consiste en:

- Se hace $N_C = K$. Se escogen K elementos al azar de entre los N a clasificar, formándose con cada uno de ellos, como centroide provisional, una clase inicial.
- Se agrupan las muestras con las clases formados según la mínima distancia.
- Se eliminan las clases con un número de elementos N_i inferior a θ_N , actualizando N_C .
- Se actualizan los centroides de las clases c_i calculando la media de los elementos de cada clase A_i .
- Se obtienen una serie de parámetros estadísticos de cada clase:
 - D_i : Media de las distancias de los z_i elementos pertenecientes a una clase A_i con respecto a su centroide c_i .
 - $\sigma_{i1}, \sigma_{i2}, \dots, \sigma_{in}$: Desviaciones típicas de cada grupo para cada una de sus dimensiones.
 - D : Distancia media de todas las clases.
 - σ_{iMAX} : Desviación típica máxima de cada clase.
- Se ha de tener en cuenta la posible división de clases. Para dividir una clase A_i se debe cumplir la condición:

$$\sigma_{iMAX} > \theta_S \text{ y } [(D_i > D \text{ y } N_i > 2(\theta_N + 1)) \text{ o } N_C \leq K/2] \quad (\text{C.5})$$

- Se ha de estudiar la posible unión de clases. Para ello:
 - Se calculan todas las distancias entre parejas de clases D_{ij} .
 - Se seleccionan las L (si las hay) menores distancias $\{D_1, D_2, \dots, D_L\}$ que sean menores que θ_C .
 - Se unen las clases A_i y A_j que no hayan sido fusionadas previamente en la misma iteración, obteniendo su nuevo centroide mediante:

$$c_{ij} = \frac{1}{N_i + N_j} (N_i c_i + N_j c_j) \quad (\text{C.6})$$

y actualizando el parámetro N_C en cada unión.

- Se comprueba si se ha llegado a la última iteración I . Si no es así se vuelve al segundo paso del algoritmo.

El algoritmo ISODATA se publicó por primera vez en 1965 y desde entonces se ha aplicado profusamente.

C.1.5. Algoritmo de agrupación por eliminación

Este algoritmo de agrupamiento borroso es una modificación *del método de agrupamiento de montaña* propuesto por Yager (Yager and Filev, 1994), con una carga computacional mucho menos elevada. El método de agrupación por eliminación se puede usar, por su sencillez y efectividad, con el fin de obtener unos valores iniciales para los centros de las clases que se requieren por otros algoritmos más sofisticados, como por ejemplo, el algoritmo fuzzy c-means que se verá en el punto siguiente. El método se basa en lo que hacemos visualmente los humanos para la formación de grupos en un conjunto de datos.

Considérese un conjunto de N datos n -dimensionales Z como en (C.12). Cada dato es un candidato a centro de clase (en el método del agrupamiento de montaña los candidatos son las intersecciones de una rejilla definida por el usuario). Una *medida de densidad* para el dato z_i es:

$$D_i = \sum_{j=1}^N \exp \left(-\frac{\|x_i - x_j\|^2}{(r_a/2)^2} \right) \quad (\text{C.7})$$

donde r_a es una constante positiva. Un punto tendrá una alta densidad si tiene muchos puntos en su vecindad. El radio r_a define esta vecindad, haciendo que los puntos fuera de ese radio contribuyan muy poco a la medida de densidad.

Una vez se han calculado las densidades para cada dato, se selecciona el punto con mayor densidad como centro de la primera clase. Si el punto elegido es z_{C1} y su densidad D_{C1} , se recalcula la densidad de cada punto con:

$$D_i = D_i - D_{C1} \exp \left(-\frac{\|x_i - x_{C1}\|^2}{(r_b/2)^2} \right) \quad (\text{C.8})$$

siendo r_b una constante positiva (generalmente $1.5 \times r_a$). De este modo, los puntos cercanos al primer centro de clase z_{C_1} disminuyen su densidad y se hace así muy difícil su elección como centro de otra clase. Se elige el nuevo centro como la medida de densidad más alta que quede y se repite el proceso hasta que se de la condición de parada que se especifique.

El algoritmo de agrupación por eliminación quedaría de la siguiente manera:

- Calcular la densidad D_i para cada punto de Z según:

$$D_i = \sum_{j=1}^N \exp \left(-\frac{\|x_i - x_j\|^2}{(r_a/2)^2} \right)$$

- Elegir el punto z_{C_i} con valor de densidad más alta.
- Recalcular los valores de densidades con:

$$D_i = D_i - D_{C_i} \exp \left(-\frac{\|x_i - x_{C_i}\|^2}{(r_b/2)^2} \right)$$

- Si no se da la condición de finalización (número de clases, medida de densidad menor que cierto umbral, ...) volver al paso segundo.

Relación entre parámetros del algoritmo de agrupación por eliminación y el algoritmo AFCRC. Como ya se comentando, los valores de σ_1 y σ_2 en el algoritmo AFCRC (ver sección 8.5 del capítulo 8) están relacionados con la distancia máxima del entorno del prototipo para la que deben estar la mayoría de los puntos de alta pertenencia (σ_2) y la distancia máxima a partir de la cual no deben haber altas pertenencia a la clase (σ_1). Esta idea es en cierto modo similar a la del algoritmo de agrupación por eliminación y el significado de σ_1 y σ_2 se puede relacionar con el de r_a en (C.7) y r_b (C.8).

El algoritmo de agrupación por eliminación emplea penalizaciones exponenciales en sus medidas de densidad donde, por ejemplo, para la primera de ellas se tiene $\exp \left(-\frac{L_{ik}^2}{(r_a/2)^2} \right)$ y el radio r_a define la vecindad del prototipo haciendo que los puntos fuera de ese radio contribuyan muy poco. De esta forma, se podrían reescribir (8.1) y (8.2) en la forma anterior y aprovechar el mismo significado.

Dado un radio R de la clase, los resultados de ambos criterios son muy similares ya que el denominador *den* del término exponencial resulta (C.9) con el criterio del algoritmo de agrupación por eliminación y (C.10) con el criterio estadístico. En términos relativos, (C.9) resulta un 2.78% mayor que (C.10).

$$den = \frac{R^2}{2} = \frac{R^2}{4} \quad (C.9)$$

$$R = 3\sigma \Rightarrow \sigma = \frac{R}{3} \Rightarrow den = 2\sigma^2 = 2\frac{R^2}{9} = \frac{R^2}{4.5} \quad (C.10)$$

En cuanto a la elección de σ_2 se puede emplear también la misma idea que en el algoritmo de agrupación por eliminación y tomar una relación aproximada de 1.5 entre los radios mayor y menor de la clase.

C.2. Algoritmo de agrupamiento borroso fuzzy c-means

En los algoritmos de agrupamiento clásico cada dato debe ser asignado a una clase. El análisis con agrupamiento borroso relaja este requerimiento permitiendo pertenencias graduales y ofreciendo la oportunidad de trabajar con datos que pertenecen a más de una clase al mismo tiempo. La mayoría de las técnicas analíticas de agrupamiento borroso se basan en la optimización de la función objetivo c-means o alguna modificación de ésta.

C.2.1. Función objetivo c-means

La función objetivo base de una gran familia de algoritmos de agrupamiento borroso es la siguiente:

$$J(Z; U, C) = \sum_{i=1}^c \sum_{k=1}^N (\mu_{ik})^m \|z_k - c_i\|_B^2 \quad (\text{C.11})$$

donde

$$Z = \{z_1, z_2, \dots, z_N\} \quad (\text{C.12})$$

son los datos que deben ser clasificados,

$$U = [\mu_{ik}] \in M_{fc} \quad (\text{C.13})$$

es una matriz partición borrosa (ver (6.15)) de Z ,

$$C = [c_1, c_2, \dots, c_c], c_i \in \mathbf{R}^n \quad (\text{C.14})$$

es el vector de centros (centroides, prototipos) a determinar,

$$D_{ik B}^2 = \|z_k - c_i\|_B^2 = (z_k - c_i)^T B (z_k - c_i) \quad (\text{C.15})$$

es una norma, y

$$m \in [1, \infty) \quad (\text{C.16})$$

es un exponente que determina la "borrosidad" de las clases resultantes. El valor de la función de coste (C.11) es una medida ponderada del error cuadrático que se comete al representar las c clases por los prototipos c_i .

C.2.2. Algoritmo fuzzy c-means (FCM)

La minimización de la función objetivo (C.11) es un problema de optimización no lineal que puede ser resuelto de muchas formas, pero la más habitual es la conocida como algoritmo fuzzy c-means o fuzzy ISODATA.

Los puntos estacionarios de la función objetivo (C.11) se encuentran mediante su minimización en dos pasos independientes:

- minimizar para U con C fija, lo que genera un problema de minimización con restricciones por la condición de suma 1 de las funciones de pertenencia. Este problema se resuelve añadiendo la condición (6.13) a J mediante los multiplicadores de Lagrange (Bezdek, 1987):

$$\bar{J}(Z; U, C, \lambda) = \sum_{i=1}^c \sum_{k=1}^N (\mu_{ik})^m D_{ik B}^2 + \sum_{k=1}^N \lambda_k \left(\sum_{i=1}^c \mu_{ik} - 1 \right) \quad (\text{C.17})$$

La minimización de (C.17) se lleva a cabo igualando a cero las derivadas parciales de \bar{J} con respecto a U y λ .

- minimizar para C con U fija, que es un problema simple de minimización sin restricciones. Esto se resuelve igualando a cero la derivada parcial de J con respecto a C .

La combinación de todo esto lleva a que las condiciones necesarias para que (C.11) alcance su mínimo (posiblemente local) sean:

$$\mu_{ik} = \frac{1}{\sum_{j=1}^c (D_{ik B} / D_{jk B})^{2/(m-1)}}, \quad 1 \leq i \leq c \quad 1 \leq k \leq N \quad (\text{C.18})$$

$$c_i = \frac{\sum_{k=1}^N (\mu_{ik})^m z_k}{\sum_{k=1}^N (\mu_{ik})^m}, \quad 1 \leq i \leq c \quad (\text{C.19})$$

Esta solución también satisface las condiciones (6.12) y (6.14). La ecuación (C.19) nos da un valor para c_i como la media ponderada de los datos que pertenecen a una clase, donde los pesos son las funciones de pertenencia.

Los pasos que debe seguir el algoritmo c-means, dado un conjunto de datos como (C.12), y seleccionados un número $c \in [1, N]$ de clases, un exponente de ponderación $m > 1$, una tolerancia $\varepsilon > 0$ (normalmente entre 0.01 y 0.001) y una norma B , son los siguientes:

- Inicializar la matriz de partición borrosa U con valores aleatorios $U^{(0)}$, haciendo $l=1$ (índice de iteración).
- Calcular los centros de las clases:

$$c_i^{(l)} = \frac{\sum_{k=1}^N (\mu_{ik}^{(l-1)})^m z_k}{\sum_{k=1}^N (\mu_{ik}^{(l-1)})^m}, \quad 1 \leq i \leq c$$

- Calcular las distancias:

$$D_{ik B}^2 = (z_k - c_i^{(l)})^T B (z_k - c_i^{(l)}), \quad 1 \leq i \leq c \quad 1 \leq k \leq N$$

- Recalcular la matriz de partición:
 - si $D_{ikB} > 0$ para todo $1 \leq i \leq c, 1 \leq k \leq N$,

$$\mu_{ik}^{(l)} = \frac{1}{\sum_{j=1}^c (D_{ikB} / D_{jkB})^{2/(m-1)}}$$

- en cualquier otro caso:
 - $\mu_{ik}^{(l)} = 0$ si $D_{ikB} > 0$, y
 - $\mu_{ik}^{(l)} \in [0,1]$ con $\sum_{i=1}^c \mu_{ik}^{(l)} = 1$ para el resto.

- Si $\|U^{(l)} - U^{(l-1)}\| < \varepsilon$, parar. Si no, volver al segundo paso del algoritmo.

C.2.3. Normas

La forma de las clases se determina por la elección de la matriz B en (C.15). Una elección habitual es tomar $B = I$, lo que lleva a la norma Euclidea:

$$D_{ik}^2 = (z_k - c_i)^T (z_k - c_i) \tag{C.20}$$

Esta elección induce clases hiperesféricas (figura 6.2.a). B también puede ser una matriz diagonal de dimensiones $n \times n$ (n es la dimensión de los elementos de Z) que toma diferentes varianzas en cada una de las direcciones de los ejes coordenados de Z:

$$B = \begin{bmatrix} (1/\sigma_1)^2 & 0 & \cdot & \cdot & 0 \\ 0 & (1/\sigma_2)^2 & \cdot & \cdot & 0 \\ \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot \\ 0 & 0 & \cdot & \cdot & (1/\sigma_n)^2 \end{bmatrix} \tag{C.21}$$

Esta matriz induce una norma diagonal en \mathbf{R}^n y una forma hiperelipsoidal en las clases (figura 6.2.b). La última opción es definir B como la inversa de la matriz de covarianzas de las muestras de Z (\bar{z} es la media de los datos):

$$B = R^{-1} \quad \text{con} \quad R = \frac{1}{N} \sum_{k=1}^N (z_k - \bar{z})(z_k - \bar{z})^T \tag{C.22}$$

En este caso se induce la norma de Mahalanobis (Bezdek, 1987), que también da lugar a clases hiperelipsoidales, pero con una orientación arbitraria de sus ejes (figura 6.2.c).

Algunas normas son sensibles a los rangos numéricos de los datos, con lo que una normalización previa puede ser una buena opción. La norma Euclidea, por ejemplo, asigna un mayor peso a aquellas características con rangos amplios que a las de rangos estrechos. Una simple elección de unidades (metros o centímetros) puede influenciar negativamente los resultados del agrupamiento. La literatura de reconocimiento de patrones sugiere normalizar los datos previamente al agrupamiento (Jain and Dubes, 1988). La normalización más simple es la dada por la resta de la media de la característica \bar{z}_j :

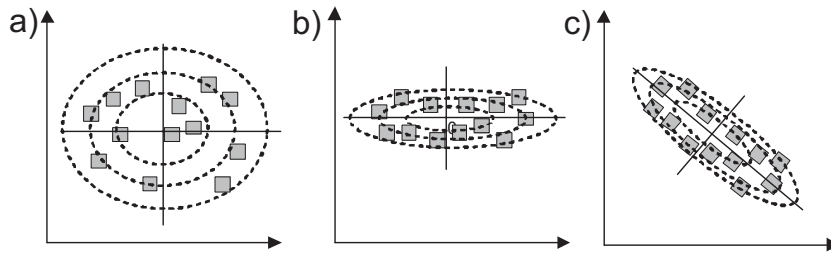


Figura C.2: Normas usadas en el agrupamiento borroso.

$$z_{jk} = z_{jk}^* \bar{z}_j \quad (\text{C.23})$$

que hace los valores de la característica invariantes a desplazamientos rígidos de las coordenadas. El asterisco denota los datos "sin escalar". Otro tipo de normalización traslada y escala los ejes de forma que todas las características tengan media nula y varianza unitaria:

$$z_{jk} = \frac{z_{jk}^* \bar{z}_j}{\sigma_j} \quad (\text{C.24})$$

Sin embargo, la normalización no siempre es deseable y puede alterar la separación entre las clases y afectar negativamente los resultados del agrupamiento. Afortunadamente, los algoritmos basados en medidas de distancia adaptativas (apartado C.3) no son sensibles al escalado de los datos, ya que la adaptación de la medida de distancia compensa automáticamente las diferencias de escala.

C.2.4. Extensiones al algoritmo FCM

En algunas aplicaciones el algoritmo FCM no da unos buenos resultados, principalmente por identificar todas las clases con dimensiones y orientación similar y una forma hipersférica o hiperelipsoidal.

Sin embargo, existen numerosas extensiones y modificaciones al algoritmo básico fuzzy c-means que se ha descrito. Algunas de ellas pueden hacer que sus resultados sean más adecuados para otros fines, como por ejemplo el modelado y control por modelos locales. Estos métodos de posible interés para aplicaciones en las que la detección de espacios lineales sea de importancia, pueden ser clasificados en dos grandes grupos:

- Algoritmos que utilizan una medida de la distancia adaptativa (una norma diferente para cada clase), que serán estudiados en el punto C.3.
- Algoritmos basados en prototipos lineales (norma constante y prototipos variables). Es una alternativa a la solución anterior de las restricciones de FCM que se tratará en el punto C.4.

De entre los algoritmos con distancia adaptativa, cabe destacar el de Gustafson-Kessel (GK), algoritmo que extiende el algoritmo básico fuzzy c-means eligiendo una norma diferente B_i para cada clase. Este algoritmo, aunque no tanto como el FCM, se

usa bastante en la bibliografía para la obtención de modelos borrosos, dado que las clases hiperlipsoydales que busca detectan de forma bastante correcta los comportamientos quasi-lineales de los diversos regímenes de funcionamiento que pueden existir en un conjunto de datos.

Finalmente, aunque no sea de aplicación al ámbito de trabajo de esta tesis, conviene resaltar la existencia de un tercer tipo de modificaciones al algoritmo FCM básico que lo hace válido para la detección de clases huecas y que podríamos englobar con el nombre de algoritmos basados en prototipos no lineales, que se pueden consultar en el Anexo C.

C.3. Algoritmos con norma adaptativa

Existen muchos algoritmos que, basados en el algoritmo básico fuzzy c-means, adaptan (C.15) con el fin de detectar clases de diferentes formas geométricas y tamaños en un mismo conjunto de datos. En este apartado se recogen los más apropiados para la detección de subespacios lineales.

C.3.1. Algoritmo de Gustafson-Kessel (GK)

Este algoritmo extiende el algoritmo básico fuzzy c-means eligiendo una norma diferente B_i para cada clase, con lo que (C.15) pasa a ser (Gustafson and Kessel, 1979):

$$D_{ik B_i}^2 = \|z_k - c_i\|_{B_i}^2 = (z_k - c_i)^T B_i (z_k - c_i) \quad (\text{C.25})$$

Estas matrices son ahora tomadas como posibles variables para la optimización de la función (C.11), con lo que se adaptará la norma a cada clase según sus características. Sea $B = \{B_1, B_2, \dots, B_c\}$ el vector que contiene las c normas. La nueva función a minimizar será:

$$J(Z; U, C, B) = \sum_{i=1}^c \sum_{k=1}^N (\mu_{ik})^m D_{ik B_i}^2 \quad (\text{C.26})$$

cumpliendo (C.13), (C.14) y (C.16). Para obtener una solución viable B_i debe ser limitada de alguna forma. La forma más habitual es fijar el determinante de B_i , lo que es equivalente a optimizar la forma dlla clase manteniendo su volumen constante:

$$|B_i| = \rho_i, \quad \rho_i > 0 \quad (\text{C.27})$$

con ρ_i constante para cada clase. La expresión que se obtiene añadiendo, a los pasos descritos en C.2 para la minimización de la función objetivo de FCM, un tercer paso para la minimización de B_i con U y C fijos con la restricción (C.27) de volumen constante es:

$$B_i = [\rho_i \det(F_i)]^{1/n} F_i^{-1} \quad (\text{C.28})$$

siendo F_i la matriz de covarianzas de clase i definida por:

$$F_i = \frac{\sum_{k=1}^N (\mu_{ik})^m (z_k - c_i)(z_k - c_i)^T}{\sum_{k=1}^N (\mu_{ik})^m} \quad (\text{C.29})$$

Dado un conjunto de datos como (C.12), y seleccionados un número $c \in [1, N]$ de clases, un exponente de ponderación $m > 1$, un valor para cada determinante ρ_i ($\rho_i = 1$ si no hay información previa en ese sentido) y una tolerancia $\varepsilon > 0$ para la terminación, los pasos del algoritmo GK completo son los siguientes:

- Inicializar la matriz de partición borrosa U con valores aleatorios $U^{(0)}$, haciendo $l=1$ (índice de iteración).

- Calcular los centros de las clases:

$$c_i^{(l)} = \frac{\sum_{k=1}^N (\mu_{ik}^{(l-1)})^m z_k}{\sum_{k=1}^N (\mu_{ik}^{(l-1)})^m}, \quad 1 \leq i \leq c$$

- Calcular la matriz de covarianzas de cada clase:

$$F_i^{(l)} = \frac{\sum_{k=1}^N (\mu_{ik}^{(l-1)})^m (z_k - c_i^{(l)})(z_k - c_i^{(l)})^T}{\sum_{k=1}^N (\mu_{ik}^{(l-1)})^m}, \quad 1 \leq i \leq c$$

- Calcular las distancias:

$$B_i = (z_k - c_i^{(l)})^T \left[[\rho_i \det(F_i)]^{1/n} F_i^{-1} \right] (z_k - c_i^{(l)}), \quad 1 \leq i \leq c \quad 1 \leq k \leq N$$

- Recalcular la matriz de partición:

- si $D_{ikB_i} > 0$ para todo $1 \leq i \leq c, 1 \leq k \leq N$,

$$\mu_{ik}^{(l)} = \frac{1}{\sum_{j=1}^c (D_{ikB_i} / D_{jkB_i})^{2/(m-1)}}$$

- en cualquier otro caso:

$$\mu_{ik}^{(l)} = 0 \text{ si } D_{ikB_i} > 0, \text{ y}$$

$$\mu_{ik}^{(l)} \in [0, 1] \text{ con } \sum_{i=1}^c \mu_{ik}^{(l)} = 1 \text{ para el resto.}$$

- Si $\|U^{(l)} - U^{(l-1)}\| < \varepsilon$, parar. Si no, volver al segundo paso del algoritmo.

C.3.2. Algoritmo de estimación de la máxima probabilidad (FMLE)

La norma utilizada en este algoritmo, basada una estimación de máxima probabilidad que incluye un término exponencial que hace decrecer más rápido que (C.25), es (Bezdek and Dunn, 1975):

$$D_{ik}^2_{G_i} = \frac{[\det G_i]^{1/2}}{P_i} \exp [(z_k - c_i)^T G_i^{-1} (z_k - c_i)/2] \quad (C.30)$$

con:

$$G_i = \frac{\sum_{k=1}^N \mu_{ik} (z_k - c_i)(z_k - c_i)^T}{\sum_{k=1}^N \mu_{ik}} \quad (C.31)$$

y:

$$P_i = \frac{1}{N} \sum_{k=1}^N \mu_{ik} \quad (C.32)$$

Las diferencias de cálculo entre el algoritmo FMLE y el ya visto de GK estriban en que la norma (C.30) tiene un término exponencial y la matriz de covarianzas (C.31) no incluye el término ponderante m , pero el esquema iterativo es prácticamente el mismo. FMLE es capaz de detectar clases con variaciones en forma, tamaños y densidad, pero es muy sensible a los valores de inicialización.

C.3.3. Algoritmo de Gath y Geva (GG)

Si el interés está en encontrar clases elipsoidales de tamaño variable, el *algoritmo de Gath y Geva* (Gath and Geva, 1989) es una buena herramienta.

El algoritmo de Gath y Geva fue desarrollado con el fin de obtener una solución satisfactoria al problema de la gran variabilidad en formas y densidades de las clases, desarrollando una estrategia de agrupamiento en dos capas. En el primer paso, se utiliza el algoritmo FCM. No se dan condiciones iniciales sobre la localización de los centros de las clases y los prototipos son identificados en un proceso de aprendizaje no supervisado. Usando esos prototipos, un segundo paso busca una partición borrosa óptima mediante FMLE. Este esquema de trabajo es iterativo, incrementando el número de clases en el conjunto de datos y calculando medidas de comportamiento y validez de las clases en cada paso hasta que la partición óptima es obtenida.

Los pasos que debe seguir el algoritmo de Gath y Geva, dado un conjunto de datos como (C.12), seleccionados los exponentes de ponderación $m > 1$, las tolerancias $\varepsilon > 0$ (normalmente entre 0.01 y 0.001) para cada algoritmo de agrupamiento y norma Euclídea $B = I$ para FCM, son los siguientes:

1. Elegir un número $c \in [1, N]$ de clases si se tiene información a priori del proceso. Si no es así, estimar un valor inicial:

- a) Calcular la media y la desviación típica de los datos.
 - b) Elegir un prototipo en la media de la distribución.
 - c) Elegir un prototipo adicional equidistante a todos los puntos.
 - d) Calcular la partición borrosa de los datos con FCM.
 - e) Si el número c de clases es menor que un valor máximo dado, volver al paso 1.3. Si no, continuar.
2. Aplicar el método de agrupamiento borroso FCM.
 3. Aplicar el método de agrupamiento borroso FMLE con los centros obtenidos del paso anterior.
 4. Calcular índices de validez de las clases basados en criterios de hipervolumen y densidad en función del número de clases.
 5. Aumentar el número c de clases y repetir los pasos 2 a 4 hasta que se obtenga un valor óptimo de los índices.

C.4. Algoritmos con prototipos lineales

En los algoritmos vistos hasta el momento, las clases han venido representados por unos puntos prototipo (centroides, centros) c_i con estructura geométrica del mismo tipo que los datos. La limitación de forma y tamaño de las clases impuesta por FCM ha sido resuelta con GK mediante normas variables. La otra opción es mantener la norma constante pero definir prototipos r -dimensionales ($0 \leq r \leq n-1$), lineales o no lineales, en subespacios del espacio de datos. Los algoritmos referentes a espacios subespacios lineales serán comentados a continuación. Los algoritmos con prototipos en subespacios no lineales (búsqueda de clases huecos) se comentan en C.6.

C.4.1. Algoritmo fuzzy c -varieties (FCV)

La idea principal de este algoritmo (Bezdek, 1987) es medir las distancias entre los datos y formas lineales r -dimensionales (*variedades*), es decir, entre los datos y líneas ($r = 1$), planos ($r = 2$) e hiperplanos ($2 < r < n$). De esta forma se pueden encontrar clases de datos en los subespacios lineales r -dimensionales de \mathbf{R}^n . El criterio se da esta vez como:

$$J_C(Z; U, C_r) = \sum_{i=1}^c \sum_{k=1}^N (\mu_{ik})^m D_{rik}^2 \quad (\text{C.33})$$

donde $C_r = \{C_{r1}, C_{r2}, \dots, C_{rc}\}$ es el conjunto de variedades lineales r -dimensionales y D_{rik}^2 es una distancia ortogonal de z_k a la variedad C_{ri} :

$$D_{rik}^2 = \|z_k - C_{ri}\|^2 - \sum_{j=1}^r \langle z_k - C_{ri}, s_{ij} \rangle^2 \quad (\text{C.34})$$

donde C_i es un punto incluido en la variedad, $(s_{i1}, s_{i2}, \dots, s_{ir})$ es un conjunto de vectores linealmente independientes (autovectores principales) en el espacio r -dimensional de la variedad y $\langle \cdot, \cdot \rangle$ es el producto escalar.

Dado un conjunto de datos como (C.12), y seleccionados un número $c \in [1, N]$ de clases, un exponente de ponderación $m > 1$, la dimensión de las variedades $r \in [0, n]$ y una tolerancia $\varepsilon > 0$ para la terminación, los pasos a seguir para implementar el algoritmo FCV son los siguientes:

- Inicializar la matriz de partición borrosa U con valores aleatorios $U^{(0)}$, haciendo $l=1$ (índice de iteración).
- Calcular los centros de las clases:

$$C_{ri}^{(l)} = \frac{\sum_{k=1}^N (\mu_{ik}^{(l-1)})^m z_k}{\sum_{k=1}^N (\mu_{ik}^{(l-1)})^m}, \quad 1 \leq i \leq c$$

- Calcular la matriz de covarianzas de cada clase:

$$F_i^{(l)} = \frac{\sum_{k=1}^N (\mu_{ik}^{(l-1)})^m (z_k - C_{ri}^{(l)})(z_k - C_{ri}^{(l)})^T}{\sum_{k=1}^N (\mu_{ik}^{(l-1)})^m}, \quad 1 \leq i \leq c$$

- Cálculo de autovectores principales. De cada $F_i^{(l)}$ se obtienen sus autovectores principales $s_{ij}^{(l)}$, $j = 1, 2, \dots, r$ (autovectores correspondientes a sus r autovalores más grandes).
- Calcular las distancias:

$$D_{rik}^2 = \left\| z_k - C_{ri}^{(l)} \right\|^2 - \sum_{j=1}^r \left\langle z_k - C_{ri}^{(l)}, s_{ij}^{(l)} \right\rangle^2, \quad 1 \leq i \leq c \quad 1 \leq k \leq N$$

- Recalcular la matriz de partición:

- si $D_{rik} > 0$ para todo $1 \leq i \leq c$, $1 \leq k \leq N$,

$$\mu_{ik}^{(l)} = \frac{1}{\sum_{j=1}^c (D_{rik}/D_{rjk})^{2/(m-1)}}$$

- en cualquier otro caso:

$$\mu_{ik}^{(l)} = 0 \text{ si } D_{rik} > 0, \text{ y}$$

$$\mu_{ik}^{(l)} \in [0, 1] \text{ con } \sum_{i=1}^c \mu_{ik}^{(l)} = 1 \text{ para el resto.}$$

- Si $\|U^{(l)} - U^{(l-1)}\| < \varepsilon$, parar. Si no, volver al segundo paso del algoritmo.

Los problemas más habituales de este algoritmo vienen porque las variedades no están limitadas en tamaño y el algoritmo tiende a conectar clases colineales que deberían quedar separados. Además, el algoritmo no hace las particiones correctamente si los tamaños de clases son muy dispares.

C.4.2. Algoritmo fuzzy c-regression models (FCRM)

Este algoritmo ajusta los parámetros del modelo de regresión:

$$y_k = f_i(x_k; \beta_i) \quad (\text{C.35})$$

con las funciones f_i parametrizadas por $\beta_i \in \mathbf{R}^{p_i}$. El grado de pertenencia $\mu_{ik} \in U$ se interpreta en este caso como la cercanía existente entre el valor predicho por el modelo (C.35) e y_k . El error de predicción suele calcularse como:

$$E_{ik}(\beta_i) = (y_k - f_i(x_k; \beta_i))^2 \quad (\text{C.36})$$

La función objetivo a minimizar con el método que se presenta, definidas $U \in M_{fc}$ y $(\beta_1, \dots, \beta_c) \in \mathbf{R}^{p_1} \times \mathbf{R}^{p_2} \times \dots \times \mathbf{R}^{p_c}$ es:

$$E_m(U, \{\beta_i\}) = \sum_{i=1}^c \sum_{k=1}^N (\mu_{ik})^m E_{ik}(\beta_i) \quad (\text{C.37})$$

Una posibilidad para minimizar (C.37) se presenta en (Hathaway and Bezdek, 1991) y queda resumida en el siguiente algoritmo. Dado un conjunto de datos $Z = \{(x_1, y_1), \dots, (x_N, y_N)\}$, seleccionados un número $c \in [1, N]$ de clases, un exponente de ponderación $m > 1$ y una tolerancia $\varepsilon > 0$ para la terminación:

- Inicializar la matriz de partición borrosa U con valores aleatorios $U^{(0)}$, haciendo $l=1$ (índice de iteración).
- Calcular valores para los parámetros β_i^* del modelo que minimicen la función $E_m(U^{(l)}, \{\beta_i\})$, mediante mínimos cuadrados.
- Recalcular la matriz de partición para todo $1 \leq i \leq c$, $1 \leq k \leq N$,

$$\mu_{ik}^{(l)} = \frac{1}{\sum_{j=1}^c (E_{ik}/E_{jk})^{2/(m-1)}}$$

si $E_{ik} = 0$ para algún $i = s$, hacer $\mu_{sk} = 1$ y $\mu_{ik} = 0$ para $\forall i \neq s$.

- Si $\|U^{(l)} - U^{(l-1)}\| < \varepsilon$, parar. Si no, volver al segundo paso del algoritmo.

C.5. Algoritmos con prototipos mixtos

Existe otro tipo de algoritmos (además de los que tienen norma adaptativa o los que emplean prototipos lineales) que intentan superar los problemas de los algoritmos con prototipos lineales mediante la combinación de las dos opciones vistas y que se podrían llamar algoritmos con prototipos mixtos. Estos algoritmos son el algoritmo fuzzy c-elliptotypes (Bezdek, 1987) (FCE) y el algoritmo adaptive fuzzy c-regression models (Ryoke and Y.Nakamori, 1996) (AFCR).

C.5.1. Algoritmo fuzzy c-elliptotypes (FCE)

La extensión está basada en (Davé, 1990), donde se calcula un mínimo local de una función objetivo basada en FCM (C.11) en la que se sustituye D_{ik}^2 (C.15) por una combinación convexa de distancias.

Con este algoritmo se pretenden superar algunos de los problemas del FCV. Para ello, se fuerza a cada clase a tener un centro de gravedad cdg_i y se mide la distancia como combinación de las distancias de FCM y FCV:

$$D_{eik} = \alpha D_{ik} + (1 - \alpha) D_{rik} \quad (C.38)$$

con $\alpha \in [0,1]$. Sustituyendo (C.20) y (C.34) en (C.38) tenemos:

$$D_{eik} = (z_k - cdg_i)^T (z_k - cdg_i) - \alpha \sum_{j=1}^r \langle z_k - cdg_i, s_{ij} \rangle^2 \quad (C.39)$$

El problema en este algoritmo es la elección del α correcto para cada clase.

C.5.2. Adaptive fuzzy c-regression models (AFCR)

El algoritmo AFCR (Ryoke and Y.Nakamori, 1996) (del que tan solo se han encontrado referencias en la literatura japonesa) proporciona un tratamiento similar a FCRM del que hace FCE con FCV. En este caso el criterio queda:

$$J(Z; U, \beta_1, \dots, \beta_c, c_1, \dots, c_c) = \sum_{i=1}^c \sum_{k=1}^N (\mu_{ik})^m D_{aik} \quad (C.40)$$

y se toma la distancia como combinación de las distancias de FCRM (E_{ik}) y FCM (D_{ik}):

$$D_{aik} = \alpha_i E_{ik}(\beta_i) + (1 - \alpha_i) \eta D_{ik} \quad (C.41)$$

con $\alpha_i \in [0,1]$. El primer término proporciona el mismo criterio que FCRM y el segundo incrementa la capacidad de partición en las variables ya que tiene en cuenta la distancia de los datos al prototipo de las clases. La elección del α_i se hace de forma dinámica en este algoritmo y se acerca a 1 según la estructura descubierta en la clase i es más lineal. Su determinación se basa en

$$\alpha_i = 1 - \frac{\min_l \{\lambda_{il}\}}{\max_l \{\lambda_{il}\}}, \quad i = 1, 2, \dots, c \quad (\text{C.42})$$

con λ_{il} los autovalores de la matriz de covarianzas definida en (C.29) para GK. Este procedimiento es muy importante para evitar mínimos locales no deseados pero, dependiendo de la complejidad de los datos, se podría pensar en otros índices (más simples) que variaran de 0 a 1 según las clases se hiciesen más lineales.

De este modo se conjuga en un solo algoritmo ventajas de FCRM, FCM y GK. El parámetro η sirve de balance entre los términos cuando su tamaño medio es muy diferente y no hay ningún estudio respecto a su determinación.

C.6. Algoritmos con prototipos no lineales

Los algoritmos de agrupamiento borroso basados en FCM discutidos en el capítulo 6 buscan clases situadas en subespacios lineales del espacio de los datos. Sin embargo, aunque no sea de aplicación al ámbito de trabajo de esta Tesis, también es posible (con modificaciones al algoritmo FCM básico) detectar clases que están en subespacios no lineales, similares a cáscaras, caparazones o superficies sin puntos interiores, que llamaremos *clases huecas*.

La única diferencia entre los algoritmos vistos hasta el momento y los que se van a ver en esta apartado es que los prototipos de los *algoritmos borrosos para la búsqueda de clases huecas* son curvas, superficies o hipersuperficies. Se van a describir dos algoritmos de este tipo (Klawonn *et al.*, 1997), uno para la búsqueda de superficies de elipsoides y otro para la búsqueda de cuádricas o hipercuádricas.

C.6.1. Algoritmo fuzzy c ellipsoidal shells (FCES)

Los algoritmos de este tipo pueden buscar elipses y todas sus posibles representaciones r-dimensionales. Se va a describir, por ejemplo, el algoritmo de búsqueda de elipses. Una elipse viene dada por:

$$(x - c_i)^T B_i (x - c_i) = 1 \quad (\text{C.43})$$

donde x es un punto perteneciente a la elipse, c_i es el centro de la elipse, y B_i es una matriz simétrica positiva que determina los ejes de la elipse y su orientación. Con esta descripción de la elipse, se generan prototipos $C_{ei} = C_{ei}(c_i, B_i)$ para las clases. La medida de la distancia será la norma (C.20), quedando:

$$D_{e\ ik}^2 = \|z_k - X_{ik}\|^2 = (z_k - X_{ik})^T (z_k - X_{ik}) \quad (\text{C.44})$$

donde X_i es el punto de intersección de la elipse C_{ei} con la línea que va de su centro c_i al punto z_k del conjunto de datos. La distancia (C.44) se transforma en (C.45) haciendo uso de (C.43):

$$D_{e\ ik}^2 = \frac{\left(\sqrt{(z_k - c_i)^T B_i (z_k - c_i)} - 1\right) \|z_k - X_{ik}\|^2}{(z_k - c_i)^T B_i (z_k - c_i)} \quad (C.45)$$

Minimizando la función objetivo (C.46) que se obtiene desde (C.33) utilizando como variedad $C_e = \{C_{e1}, \dots, C_{ec}\}$ y la distancia (C.45):

$$J_e(Z; U, C_e) = \sum_{i=1}^c \sum_{k=1}^N (\mu_{ik})^m D_{e\ ik}^2 \quad (C.46)$$

se obtiene el sistema de ecuaciones

$$\sum_{k=1}^N (\mu_{ik})^m (z_k - c_i)(z_k - c_i)^T \left(\frac{\|z_k - c_i\|}{d_{ik}}\right)^2 (\sqrt{d_{ik}} - 1) = 0 \quad (C.47)$$

$$\sum_{k=1}^N \frac{(\mu_{ik})^m (\sqrt{d_{ik}} - 1)}{d_{ik}^2} \left[\|z_k - c_i\|^2 B_i + (\sqrt{d_{ik}} - 1) d_{ik} I \right] (z_k - c_i) = 0 \quad (C.48)$$

donde $d_{ik}^2 = (z_k - c_i)^T B_i (z_k - c_i)$ e I es la matriz identidad. El sistema de ecuaciones formado por (C.47) y (C.48) se ha de resolver, por supuesto, por técnicas numéricas. El resultado será la obtención de los c_i centros de las clases, así como los valores μ_{ik} de la matriz de partición borrosa. Es necesario, antes de comenzar la resolución, determinar el número de clases que existen en el conjunto de datos.

C.6.2. Algoritmo fuzzy c quadric shells (FCQS)

Con este algoritmo la búsqueda da por resultado clases con forma de cuádrlica o hipercuádrlica. Las hipercuádrlicas se definen por:

$$p_i^T q = 0 \quad (C.49)$$

donde: $p_i^T = (p_{i1}, p_{i2}, \dots, p_{iN}, p_{i(N+1)}, \dots, p_{ir}, p_{i(r+1)}, \dots, p_{is})$,
 $q^T = (x_1^2, x_2^2, \dots, x_n^2, x_1 x_2, \dots, x_{n-1} x_n, \dots, x_1, x_2, \dots, x_n, 1)$,
 $s = N(N+1)/2 + N + 1 = r + N + 1$,
 $r = N(N+1)/2$,

y N es la dimensión del vector de datos. De este modo, los prototipos del algoritmo FCQS son s -dimensionales. La distancia del punto z_k al prototipo C_{qi} será:

$$D_{q\ ik}^2 = p_i^T q_k q_k^T p_i = p_i^T M_k p_i \quad (C.50)$$

con $M_k = q_k q_k^T$. Para evitar que el algoritmo caiga en la solución trivial $p_i^T = (0, \dots, 0)$ se necesita una condición adicional. En el caso bidimensional la restricción recomendada es:

$$\left\| p_{i1}^2 + p_{i2}^2 + \dots + p_{iN}^2 + \frac{1}{2} p_{i(N+1)}^2 + \dots + p_{ir}^2 \right\|^2 = 1 \quad (C.51)$$

aunque, sin embargo, esta condición hace que el algoritmo interprete las clases lineales como hipérbolas o elipses con eje mayor muy grande con respecto a su eje menor. La solución pasa por ejecutar un algoritmo para la detección de líneas después del FCQS. En (Klawonn *et al.*, 1997) se comenta que la aplicación de CCM da muy buenos resultados cuando se inicializa con esas clases detectados pos FCQS como hipérbolas o elipses con eje mayor muy grande con respecto a su eje menor.

Con el fin de simplificar la notación, se definen $a_i = \{a_{i1}, \dots, a_{iN}\}$ y $b_i = \{b_{i1}, \dots, b_{iN}\}$ donde:

$$a_{ik} = \begin{cases} p_{ik} & 1 \leq k \leq N \\ p_{ik} & N + 1 \leq k \leq r \\ b_{ik} = p_{i(r+k)} & 1 \leq k \leq s - r \end{cases}$$

quedando simplificada la condición (C.51) a $\|a_i\|^2 = 1$. La función objetivo (C.33) se minimiza con respecto a los prototipos a_i y b_i , que se calculan como:

$$a_i = \text{autovector correspondiente al menor autovalor de } (F_i - G_i^T H_i^{-1} G_i)$$

$$b_i = -H_i^{-1} G_i a_i,$$

donde:

$$F_i = \sum_{j=1}^N \mu_{ij}^m R_j, \quad R_j = r_j r_j^T$$

$$G_i = \sum_{j=1}^N \mu_{ij}^m S_j, \quad S_j = r_j t_j^T$$

$$H_i = \sum_{j=1}^N \mu_{ij}^m T_j, \quad T_j = t_j t_j^T$$

$$r_j^T = [x_{j1}^2, x_{j2}^2, \dots, x_{jN}^2, \sqrt{2}x_{j1}x_{j2}, \dots, \sqrt{2}x_{jk}x_{jl}, \dots, \sqrt{2}x_{j(N-1)}x_{jN}],$$

$$t_j^T = [x_{j1}, x_{j2}, \dots, x_{jN}, 1].$$

De esta forma el cálculo de los prototipos se reduce a un problema trivial de cálculo de autovectores de dimensión $N(N+1)/2$. Con los prototipos ya calculados, el problema se reduce a la minimización de la función objetivo (C.33) con respecto a éstos.

Sin embargo, la medida de la distancia algebraica $D_{q,ik}^2$ elegida es altamente no lineal y depende mucho de la posición del dato con respecto al prototipo, con lo que los grados de pertenencia calculados pueden que no tengan un significado muy claro. Es decir, que dependiendo del dato, puede que se llegue a malos resultados. Para evitar dicho problema, el *algoritmo FCQS modificado* hace uso de la distancia perpendicular definida por $D_{p,ij}^2$. Para calcular esta distancia se reescribe (C.49) como $x^T A_i x + x^T b_i + c_i = 0$ y se calcula la distancia más corta entre el dato z_j y la clase β_i mediante:

$$D_{p,ij}^2 = \min_x \|z_j - x\|^2$$

sujeto a:

$$x^T A_i x + x^T b_i + c_i = 0$$

donde x es un punto de la cuádrica β_i . Mediante el multiplicador de Lagrange λ se obtiene la solución:

$$x = \frac{1}{2}(I - \lambda A_i)^{-1}(\lambda b_i + 2z_j)$$

donde I es la matriz identidad. Sustituyendo la última ecuación en la penúltima se llega a una ecuación en λ de cuarto grado. Cada raíz real λ_k del polinomio representa un valor posible de λ . Calculando los correspondientes vectores x_k , la distancia se determina con:

$$D_{p\ ij}^2 = \min_k \|z_j - x_k\|^2$$

Con esta distancia se evita el problema comentado de la obtención grados de pertenencia no válidos, pero la carga computacional es muy elevada por los métodos numéricos que se han de emplear para resolver la ecuación anterior y recalculando los prototipos. Para evitar este problema adicional, una solución válida la da el *algoritmo FCQS modificado y simplificado*. Este algoritmo utiliza la distancia $D_{q\ ik}^2$ para actualizar los prototipos y la distancia $D_{p\ ik}^2$ para actualizar los grados de pertenencia.

Para dimensiones más elevadas (más allá de 3 dimensiones) se usa el algoritmo llamado *fuzzy c planoquadric shells* (FCPQS), que utiliza la distancia aproximada $D_{a\ ij}^2$:

$$D_{a\ ij}^2 = \frac{D_{q\ ij}^2}{|\nabla D_{q\ ij}|^2} = \frac{p_i^T M_j p_i}{p_i^T (D(q_j) D(q_j)^T) p_i}$$

donde $\nabla D_{q\ ij}$ es el gradiente de $p_i^T q$ evaluado en z_j y $D(q_j)$ es el Jacobiano de q evaluado en z_j . La razón para emplear esta distancia es que el algoritmo FCQS no tiene solución para dimensiones elevadas.

C.7. Algoritmos robustos

Los métodos de agrupamiento deben ser robustos si se pretende que sean útiles en la práctica. Por *robustez* se entiende la capacidad de un algoritmo para no ser afectado, de forma significativa, por pequeñas desviaciones de un modelo y no verse drásticamente empeorado su funcionamiento por ruido o datos extraños o alejados de los habituales. En las aplicaciones reales los datos tienen ruido y elementos alejados, y las distribuciones gaussianas son solo una aproximación. Otro aspecto en el que convendría que un algoritmo de agrupamiento fuese robusto es con respecto al número de clases, ya que como se ha visto la mayoría de algoritmos son muy sensitivos a este parámetro. En (Imai *et al.*, 1998) se realiza un estudio de la influencia de las perturbaciones en datos sobre el algoritmo FCM, tanto para dirección de perturbación conocida como desconocida. Para ello se van tomando los datos uno a uno y se compara el resultado de minimizar (C.11) con el resultado obtenido antes de la modificación del punto en estudio. De este modo se identifican los datos que afectan gravemente al resultado así como la dirección de perturbación de máximo efecto en el resultado para cada punto.

A continuación se van a revisar tres algoritmos que se ocupan del problema de la robustez: el algoritmo de Ohashi, el dla clase de ruido y el de agrupamiento posibilitístico.

C.7.1. Algoritmo de Ohashi

Este algoritmo (presentado por primera vez por Ohashi en un artículo no publicado (Ohashi, 1984)) hace un intento de sobrellevar el problema de la sensibilidad al ruido de los algoritmos del tipo FCM. La idea es introducir una clase de elementos extraños y modificar la función objetivo (C.11):

$$J(Z; U, C) = \alpha \sum_{i=1}^c \sum_{k=1}^N (\mu_{ik})^m \|z_k - c_i\|_B^2 + (1 - \alpha) \sum_{k=1}^N (\mu_{*k})^m \quad (\text{C.52})$$

donde μ_{*k} es la pertenencia del punto z_k a la clase de elementos extraños y el parámetro α debe ser predefinido.

C.7.2. Algoritmo noise cluster (NC)

La idea en este caso es crear una clase de ruido (Davé, 1991) para tratar así los datos con ruido. Aquí el ruido es considerado como una clase diferente y es representado por un prototipo que tiene una distancia constante d a todos los datos. La función de pertenencia μ_{*k} de un punto z_k a la clase de ruido se define como:

$$\mu_{*k} = 1 - \sum_{i=1}^c \mu_{ik} \quad (\text{C.53})$$

Esto permite a los puntos de ruido tener una función de pertenencia arbitrariamente pequeña a las clases "buenos". La función objetivo queda en este caso:

$$J(Z; U, C) = \sum_{i=1}^c \sum_{k=1}^N (\mu_{ik})^m \|z_k - c_i\|_B^2 + \sum_{k=1}^N d^2 (1 - \sum_{i=1}^c \mu_{ik})^m \quad (\text{C.54})$$

El único problema en este método es la determinación del parámetro d , que deberá de hacerse, según se propone en (Davé, 1991) para cada caso basándose en métodos estadísticos.

C.7.3. Algoritmo possibilistic c-means (PCM)

En este método (Krishnapuram and Keller, 1993) proponen sustituir en FCM la condición partición borrosa (6.15) por una partición posibilista (6.17). La función objetivo que se utiliza con este tipo de particiones es:

$$J(Z, \eta; U, C) = \sum_{i=1}^c \sum_{k=1}^N (\mu_{ik})^m \|z_k - c_i\|_B^2 + \sum_{i=1}^c \eta_i \sum_{i=1}^c (1 - \mu_{ik})^m \quad (\text{C.55})$$

con el primer término idéntico al de FCM y donde η_i es una constante positiva. El segundo término tiene por objetivo hacer μ_{ik} tan grande como sea posible, evitando así la solución trivial $U = 0$ con la que podría darse solución a la minimización de (C.55) al sustituir (6.13) por (6.16).

La modificación de la condición (6.13) da lugar a que el algoritmo FCM se convierta en PCM, cuya estructura de iteración es, dado un conjunto de datos como (C.12), y seleccionados un número $c \in [1, N]$ de clases, un exponente de ponderación $m > 1$, una tolerancia $\varepsilon > 0$ (normalmente entre 0.01 y 0.001) y una norma B :

- Inicializar con una buena matriz de partición borrosa $U^{(0)}$, haciendo $l=1$ (índice de iteración). Estimar η_i con:

$$\eta_i = \frac{\sum_{k=1}^N (\mu_{ik})^m D_{ik B}}{\sum_{k=1}^N (\mu_{ik})^m}, \quad 1 \leq i \leq c$$

- Calcular los centros de las clases:

$$c_i^{(l)} = \frac{\sum_{k=1}^N (\mu_{ik}^{(l-1)})^m z_k}{\sum_{k=1}^N (\mu_{ik}^{(l-1)})^m}, \quad 1 \leq i \leq c$$

- Calcular las distancias:

$$D_{ik B}^2 = (z_k - c_i^{(l)})^T B (z_k - c_i^{(l)}), \quad 1 \leq i \leq c \quad 1 \leq k \leq N$$

- Recalcular la matriz de partición para todo $1 \leq i \leq c$, $1 \leq k \leq N$ con:

$$\mu_{ik}^{(l)} = \frac{1}{1 + (D_{ik B} / \eta_i)^{2/(m-1)}}$$

- Si $\|U^{(l)} - U^{(l-1)}\| < \varepsilon$, parar. Si no, volver al segundo paso del algoritmo.

Existen más métodos que tienen en cuenta el problema de la robustez en los algoritmos de agrupamiento, pero los comentados son los más empleados. Para una revisión más profunda y un establecimiento de conexiones entre las técnicas borrosas y las técnicas estadísticas robustas (Davé and Krishnapuram, 1997) es una referencia muy recomendable.

C.8. Determinación del número de clases

Excepto un número reducido de los algoritmos descritos hasta el momento (chain-map, max-min y agrupación por eliminación), todos los algoritmos de agrupamiento necesitan saber, con mayor o menor certeza, el número de clases con las que se va a encontrar para comenzar a iterar a la busca de clases.

Una solución obvia al problema del número de clases inicial es realizar un estudio previo de los datos con algún algoritmo capaz de detectarlos como chain-map, max-min o agrupación por eliminación. El valor obtenido en esta etapa de inicialización se introducirá en el algoritmo que se desee emplear. Existen otras soluciones más avanzadas a este problema de la inicialización, de las que se van a comentar tres en este apartado.

C.8.1. Algoritmo de unión de clases compatibles (CCM)

Comenzar con un número de clases suficientemente elevado y realizar uniones entre clases compatibles con el algoritmo de unión de clases compatibles (CCM) (Krishnapuram and Freg, 1992) es una posibilidad a tener en cuenta. Originalmente este algoritmo era una alternativa a FCM ó GK para mejorar la detección de clases lineales, pero la idea es aplicable a cualquier algoritmo que detecte clases lineales.

Este algoritmo usa las propiedades de cualquier algoritmo para la detección de líneas (por ejemplo, GK) y las mejora uniendo clases compatibles (dos clases son compatibles si ambas están en el mismo hiperplano y la distancia entre ellos es pequeña comparada con su tamaño). La idea del algoritmo es comenzar con un número de clases mayor que el máximo esperado para el problema en estudio. El algoritmo, dado un conjunto de datos como (C.12), quedaría de la siguiente manera:

1. Elegir un número c_{max} de clases.
2. Aplicar el método de agrupamiento borroso (por ejemplo, GK) hasta su convergencia.
3. Determinar los autovalores $\{\lambda_{i1}, \dots, \lambda_{in}\}$ y autovectores $\{\Phi_{i1}, \dots, \Phi_{in}\}$ de cada clase i .
4. Aplicar las condiciones de compatibilidad, por parejas i y j , a las clases obtenidos, con el fin de obtener conjuntos de clases compatibles. Dichas condiciones de compatibilidad son:

- las clases deben ser paralelos:

$$|\Phi_{in}^T \Phi_{jn}| \geq k_1, \quad k_1 \cong 1$$

- Las normales a los hiperplanos definidos por las clases deben ser ortogonales a la línea que une ambas clases:

$$\left| \frac{\Phi_{in}^T + \Phi_{jn}^T}{2} \frac{c_i - c_j}{\|v_i - v_j\|} \right| \leq k_2, \quad k_2 \cong 0$$

- las clases deben ser cercanos:

$$\frac{\|c_i - c_j\|}{\sqrt{\lambda_i} + \sqrt{\lambda_j}} \leq k_3, \quad k_3 \in [2, 4]$$

5. Unir cada uno de los grupos compatibles en una única clase.
6. Recalcular los centros y las matrices de covarianzas de cada clase.
7. Repetir el proceso desde el paso 2 hasta que la nueva matriz U de partición borrosa coincida con la del paso anterior.

Existen modificaciones al algoritmo básico de CCM presentado, y que consisten en la variación del criterio de compatibilidad de agrupamiento. Por ejemplo, en (Babuska, 1996), se relajan las condiciones de compatibilidad eliminando la segunda condición, con lo que se permite la unión de clases con datos contaminados por el ruido.

C.8.2. Medidas de validez

Realizar la agrupación de los datos para diferentes valores de c , y usar posteriormente *medidas de validez global* para determinar la bondad de las particiones obtenidas es la otra opción que se suele emplear.

La revisión de las técnicas existentes de validación podría dar lugar a una monografía sobre el tema. Un buen estudio sobre los índices de validez en agrupamiento clásico se puede encontrar en (Bezdek and Pal, 1998). De entre los índices existentes, los más conocidos son: el índice de Hubert, el índice de Davies-Bouldin y el índice de Dunn.

En cuanto a los índices para agrupamiento borroso, destacan el criterio de varianza (Sugeno and Yasukawa, 1993) y el de dispersión (Emami *et al.*, 1998). El criterio de varianza se define como:

$$S(C) = \sum_{i=1}^c \sum_{k=1}^N (\mu_{ik})^m (\|z_k - c_i\|^2 - \|c_i - \bar{z}\|^2) \quad (C.56)$$

y el índice de dispersión es:

$$\begin{aligned} S(C) &= \sum_{i=1}^c \sum_{k=1}^N (\mu_{ik})^m (\|z_k - c_i\|^2 - \|c_i - \bar{c}\|^2) \\ \bar{c} &= \frac{1}{\sum_{i=1}^c \sum_{k=1}^N (\mu_{ik})^m} \sum_{i=1}^c \sum_{k=1}^N (\mu_{ik})^m z_k \\ c_i &= \frac{1}{\sum_{k=1}^N (\mu_{ik})^m} \sum_{k=1}^N (\mu_{ik})^m z_k \end{aligned} \quad (C.57)$$

donde: N es el número de datos, c es el número de clases, z_k es el k -ésimo vector de datos, \bar{z} es la media de los datos y c_i el centro del i -ésima clase. En ambos casos, el número de clases se determina incrementando c hasta que $S(c)$ alcanza un mínimo local. La diferencia entre ambos índices es la extensión de vector de medias \bar{z} al vector borroso de medias \bar{c} . La diferencia entre los resultados de los dos índices se hace considerable para grandes valores de m (en al caso habitual de mantener $m=2$, los resultados no deberán diferir mucho).

Otras medida de validez para el caso de particiones borrosas puede ser, por ejemplo, el valor que va tomando la función objetivo a minimizar. Otra opción, la del índice de partición, consiste en estudiar el grado de borrosidad de las funciones de pertenencia obtenidas y a mayor certeza en la solución (menor borrosidad), más validez se le da al conjunto de clases. Otras medidas de validez están más relacionadas con la geometría de los datos, como el método del hipervolumen borroso, basado en el tamaño de las clases, y en el que una baja medida indica que se han creado pequeñas clases que tan solo encierran los datos, sin detectar clases. Con todos estos métodos de medidas de validez global puede que se llegue a que la mejor solución no tiene ninguna buena clase, por ser índices de la globalidad.

Esto se evita empleando *medidas de validez local* en cada clase, con lo que se pueden detectar buenas clases aunque la solución global no sea del todo correcta. Una medida de este tipo con muy buenos resultados, especialmente en conjuntos de datos bidimensionales, es la densidad superficial, que da la relación entre el número de datos asignados a una clase y el número de datos que tendría si la clase fuese perfecta.

C.8.3. Algoritmos de agrupamiento parcialmente supervisados

En algunos casos puede ser interesante utilizar un *algoritmo con agrupamiento parcialmente supervisado* para inicializar el algoritmo, cuando se tienen datos de alguna clase en concreto, en combinación con los métodos anteriores. En (Pedrycz and Waleczky, 1997) se añade esta información previa al algoritmo FCM definiendo:

- un vector b que distingue entre los datos que se sabe asignados a una clase y los que no, y

$$b_k = \begin{cases} 1, & \text{hay informaci3n del dato } z_k \\ 0, & \text{otro caso.} \end{cases} \quad (\text{C.58})$$

- una matriz F que asigna los valores de funciones de pertenencia conocidos de esos datos.

$$F = [f_{ik}], \quad i = 1, 2, \dots, c, \quad k = 1, 2, \dots, N \quad (\text{C.59})$$

La funci3n objetivo a minimizar ser3a la siguiente modificaci3n de (C.11):

$$J(Z; U, C; b, F) = \sum_{i=1}^c \sum_{k=1}^N (\mu_{ik})^m \|z_k - c_i\|_B^2 + \alpha \sum_{i=1}^c \sum_{k=1}^N (\mu_{ik} - f_{ik} b_k)^m \|z_k - c_i\|_B^2 \quad (\text{C.60})$$

donde α es un factor cuyo cometido es mantener el balance entre la componente supervisada y la no supervisada en el proceso de optimizaci3n. La minimizaci3n de la nueva funci3n objetivo da lugar al siguiente algoritmo para la asignaci3n de los datos a clases. Dado un conjunto de datos como (C.12), asignando la informaci3n conocida de los datos en el vector b y la matriz F , y seleccionados un n3mero $c \in [1, N]$ de clases, un exponente de ponderaci3n $m > 1$, una tolerancia $\varepsilon > 0$ (normalmente entre 0.01 y 0.001) y una norma B , son los siguientes:

- Inicializar la matriz de partici3n borrosa U con los valores conocidos en sus entradas correspondientes y con valores aleatorios $U^{(0)}$ en el resto, haciendo $l=1$ (3ndice de iteraci3n).
- Calcular los centros de las clases:

$$c_i^{(l)} = \frac{\sum_{k=1}^N (\mu_{ik}^{(l-1)})^m z_k}{\sum_{k=1}^N (\mu_{ik}^{(l-1)})^m}, \quad 1 \leq i \leq c$$

- Calcular las distancias:

$$D_{ikB}^2 = (z_k - c_i^{(l)})^T B (z_k - c_i^{(l)}), \quad 1 \leq i \leq c \quad 1 \leq k \leq N$$

- Recalcular la matriz de partición:

$$\mu_{ik}^{(l)} = \frac{1}{1 + \alpha} \left\{ \frac{1 + \alpha \left(1 - b_k \sum_{l=1}^c f_{ik} \right)}{\sum_{j=1}^c (D_{ikB} / D_{jkB})^{2/(m-1)}} + \alpha f_{ik} b_k \right\}$$

- Si $\|U^{(l)} - U^{(l-1)}\| < \varepsilon$, parar. Si no, volver al segundo paso del algoritmo.

Apéndice D

Identificación de modelos borrosos

En este anexo se hace una exposición estructurada y completa, mucho más extensa que en el capítulo 6, de los distintos pasos a seguir para la identificación del modelo de una sistema no lineal mediante modelos borrosos, identificando su estructura y parámetros con técnicas de agrupamiento.

Se finaliza con la presentación de una serie de funciones programadas en Matlab que permiten la identificación de sistemas no lineales con el método de identificación propuesto.

D.1. Proceso de identificación de modelos borrosos

El proceso de identificación de modelos borrosos con las técnicas vistas en el punto 2.5 es iterativo por naturaleza (Díez and Navarro, 1999) consistiendo en la repetición, para diferentes elecciones de diversos parámetros, de los pasos que se muestran en la figura D.1 con el fin de obtener todos los elementos del modelo descritos en el apartado 5.4.

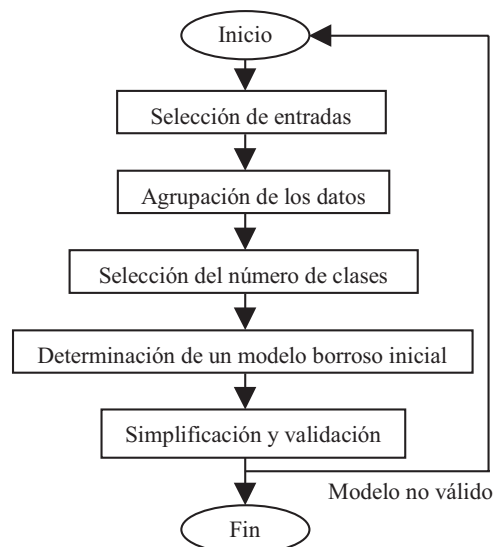


Figura D.1: Esquema sistematizado de identificación de modelos borrosos.

¿En qué partes del esquema D.1 pueden ser de ayuda las técnicas de agrupamiento? Obviamente, se podría aplicar en el paso de agrupación de datos con el fin de determinar diferentes comportamientos del sistema en estudio y dar una idea de cuántas reglas son necesarias. Sin embargo, la aplicación del agrupamiento borroso proporciona ventajas adicionales en la identificación borrosa ya que la pertenencia de los datos experimentales a las clases pueden servir para dar las funciones de pertenencia.

Con todo lo anterior puede parecer que la construcción de un modelo borroso (aunque sea aproximado) desde datos experimentales sea una tarea simple y que puede ser resuelta de forma automática. Sin embargo, tareas como la selección de entradas, la simplificación y validación de modelos no están resueltas. Además, una vez elegido el tipo de modelo borroso y el algoritmo de agrupamiento apropiados al sistema a identificar, se han de definir un buen número de parámetros como: el número de clases c , el parámetro de borrosidad m , la tolerancia ε , la norma B a emplear, etc.

El primer intento de sistematización de la identificación de modelos borrosos mediante técnicas de agrupamiento fue realizado en la primera referencia existente a este método de modelado (Sugeno and Yasukawa, 1993), y es el que ha sido utilizado posteriormente por la mayoría de autores. Sin embargo, en (Babuska, 1996) esta sistematización es ampliada en dos pasos adicionales, como son la simplificación del modelo y la validación del mismo, no explicitados en (Sugeno and Yasukawa, 1993) pero supuestos en él, ya que su fin era el de conseguir un modelo cualitativo y altamente interpretable, cuya lectura por el experto sería de por sí una forma de validación del modelo. A continuación se detallan cada uno de los pasos en que consiste la identificación del modelo tal y como proponen (Sugeno and Yasukawa, 1993), (Babuska, 1996) y (Emami *et al.*, 1998).

Diseño de experimentos Este paso inicial es importante en cualquier método de identificación. Los métodos de excitación mediante señales binarias pseudoaleatorias no son apropiados para los sistemas no lineales que se tratan en este trabajo, como los sistemas no lineales (Godfrey, 1990).

No es el objeto de esta Tesis estudiar las señales más apropiadas para la identificación de modelos borrosos de sistemas no lineales, ya que la elección de la señal de excitación puede depender del problema concreto y sería objeto de una amplia investigación adicional. Basándose en referencias (Ljung, 1999), parece que conviene excitar al sistema a identificar por un rango completo de señales tanto en amplitud como en frecuencia, por lo que las señales con dos valores de amplitud (binarias) no son apropiadas. Una señal de excitación con señales senoidales variables en amplitud y frecuencia o escalones de amplitud y anchura aleatoria pueden ser una buena solución. A la señal de excitación elegida se le suele añadir un ruido blanco de pequeña amplitud para garantizar la adecuada excitación de la dinámica del proceso. En el diseño de experimentos nunca se debe olvidar la elección de un periodo de muestreo adecuado así como del tiempo que debe durar el experimento, así como las posibles ventajas que ofrece la normalización de los datos (apartado 6.3.3).

Selección de la estructura Además de la forma de tratar la dinámica del proceso y la granularidad a emplear en el modelo, en este paso se pretende identificar cuales son

las variables de entrada y salida relevantes para el problema de modelado. Aunque el experto en el sistema puede dar las claves en este paso, se puede automatizar la elección de las variables comparando varias propuestas desde el punto de vista de algún índice. En el punto D.2 se verán con detalle algunos métodos en este sentido.

Agrupación de los datos Una vez se tienen identificadas cuales son las entradas y salidas de importancia para el sistema, se realiza una partición de los datos, mediante las técnicas de agrupamiento vistas en el capítulo 6, con el fin de identificar posibles regímenes de funcionamiento. Cada uno de esos regímenes de funcionamiento podría, por ejemplo, ser definido por un modelo lineal. El conjunto de esos submodelos lineales aproximará el modelo no lineal de la forma prevista en el anexo A. En el punto D.3 se tratará en profundidad este tema.

Selección del número de clases Para este paso son válidas las técnicas vistas en el apartado C.8. La selección del número correcto de clases pasa por una iteración de los pasos comentados en los apartados D.1 y D.1 hasta que se llegue a un valor apropiado.

Determinación de un modelo borroso inicial De algún modo, y basándose en la agrupación de los datos realizada en los pasos anteriores para el espacio de las variables elegidas, se han de derivar reglas para algún tipo de modelo borroso. En D.4 se verá con detalle este paso.

Simplificación del modelo inicial Los modelos borrosos determinados desde los datos experimentales suelen ser redundantes en el sentido de tener más funciones de pertenencia y/o reglas de las necesarias, con lo que alguna medida de similaridad deberá ser empleada para simplificar el modelo inicial. En D.5 se tratará este tema.

Validación del modelo Diferentes métodos de validación, incluida la simulación y la revisión de las reglas por parte del experto, pueden ser usados en esta etapa final, con la idea de aceptar el modelo de forma definitiva o comenzar de nuevo el proceso de modelado desde el principio. El punto D.6 define como tratar este problema.

D.2. Selección de la estructura

Tres problemas diferentes aparecen en la selección de la estructura dentro del marco del modelado borroso. Su resolución no es directa sino que requieren técnicas iterativas, lo que lleva a que sus valores definitivos no sean determinados hasta el último paso de la construcción del modelo borroso, con lo que todo el proceso de obtención del modelo se convierte en un gran problema de iteración, con sus múltiples partes relacionadas entre sí y sin ser posible la determinación aislada y precisa de los parámetros referentes a la estructura.

D.2.1. Elección de las variables de entrada

Aunque muchos de los métodos revisados en la bibliografía ((Nakamori and Ryo-ke, 1994), (Klawonn and Kruse, 1995b), (Hong and Lee, 1996), (Genther and Glesner, 1997), (Klawonn and Keller, 1997), (Klawonn and Kruse, 1997), (Delgado *et al.*, 1997) y (Kim *et al.*, 1997)) suponen, para la construcción del modelo borroso de un sistema, que las entradas y salidas son conocidas, generalmente no está claro en los sistemas multivariable qué variables han de ser elegidas como entradas al sistema (Sugeno and Yasukawa, 1993). El conocimiento de los expertos es fundamental en este tema, aunque los métodos estadísticos de correlación también pueden jugar un papel importante.

En las soluciones heurísticas a este problema se parte de un número N de posibles candidatos a ser entradas al sistema y se estudia el efecto de las distintas combinaciones de variables de entrada sobre la salida. En general, para un conjunto Z de posibles variables de entrada z_1, z_2, \dots, z_N , el número total de casos a estudiar es $2^N - 1$. Con el fin de no tener que probar todas las posibles combinaciones de entradas, se seleccionan algunas entradas de entre las candidatas y se van incrementando el número de entradas una a una, teniendo en cuenta algún criterio. En (Emami *et al.*, 1998) se emplea un *índice de no-significatividad* de las variables. Otra opción con la que resolver el problema de la selección de entradas, de mayor carga computacional pero más precisa, es el llamado *criterio de regularidad* que se da en (Sugeno and Yasukawa, 1993) se resume a continuación.

En primer lugar se divide el conjunto de datos en dos grupos, A y B, cada uno de ellos con un número similar de datos. El criterio de regularidad utilizado para elegir alguna entrada o conjunto de entradas entre los posibles es definido como:

$$RC = \frac{\left[\sum_{i=1}^{k_A} (y_i^A - y_i^{AB})^2 / k_A + \sum_{i=1}^{k_B} (y_i^B - y_i^{BA})^2 / k_B \right]}{2} \quad (D.1)$$

donde:

- k_A y k_B son el número de datos de los grupos A y B
- y_i^A e y_i^B son las salidas correspondientes a los datos de A y de B
- y^{AB} es la salida estimada para la salida de los datos de B por el modelo identificado con los datos de A
- y^{BA} es la salida estimada para la salida de los datos de A por el modelo identificado con los datos de B

La utilización de RC exige la construcción de dos modelos en cada paso de la elección de las variables. Es decir, se exige seguir con el proceso de construcción del modelo hasta al menos obtener un modelo borroso inicial en cada caso (para cada combinación de variables).

Ejemplo. Criterio de Regularidad. Supongamos que se dispone de un conjunto de datos con cuatro entradas diferentes a un sistema, y se pretende determinar cual es la mejor combinación de entradas, es decir, cual de todas las combinaciones explica mejor

la salida del sistema, obviando las entradas que no aportan información. En primer lugar, se comenzaría por la construcción de modelos borrosos de una sola entrada. Para ellos se calcularían cuatro diferentes valores de RC, uno para cada una de las entradas, lo que exige la construcción de ocho modelos borrosos (dos por entrada). La entrada que minimiza el RC se elige como la mejor y se fija para el siguiente paso. Las combinaciones de entradas a estudiar en esta segunda etapa, serán las parejas formadas por entrada fijada en el paso anterior y cada una de las restantes entradas (tres en total). Los modelos borrosos a construir tienen dos entradas en este paso. Al igual que en el paso anterior se elige la entrada (en este caso pareja de entradas) que minimizan el valor de RC. El proceso sigue hasta que el valor de RC aumente, lo que indicará que la nueva variable añadida ya no es necesaria.

En la figura D.2 se muestra una estructura en árbol que ayuda a comprender el algoritmo. Como se observa, para el ejemplo comentado de cuatro entradas, el árbol se compone de cuatro niveles. Cada nodo del árbol se corresponde a un subconjunto de los posibles candidatos. El número total de combinaciones es $2^n - 1$ para n candidatos. En cada etapa, sólo una combinación es elegida, por lo que las combinaciones unidas por líneas discontinuas no son evaluadas. Como resultado, el algoritmo evalúa, como mucho, $n(n+1)/2$ nodos en lugar de los $2^n - 1$ posibles. Para el ejemplo del árbol en la figura, tras la combinación x_1x_2 el RC aumenta, siendo la combinación x_1x_2 la elegida, tras la evaluación de las dos combinaciones posteriores.

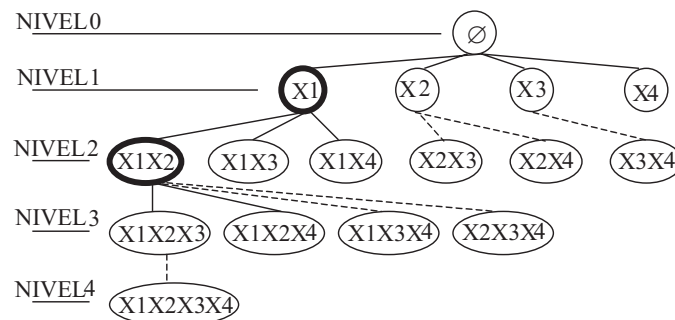


Figura D.2: Árbol de búsqueda de mejores entradas.

■

En cuanto a la segunda opción heurística para la selección de las entradas más significativas a un proceso (propuesta de índice realizada por Emami), en ella se aprovecha el hecho de que las relaciones entrada-salida se definen por particiones en el espacio entrada-salida por medio de muchas reglas SI-ENTONCES. Cada partición se representa por una función de pertenencia. Considérese, por ejemplo, el modelo borroso de un sistema MISO:

R_i : Si x_1 es A_{i1} y ... y x_s es A_{is} entonces y es B_i , $i = 1, 2, \dots, k$; $j = 1, 2, \dots, s$.

En cada regla i , las funciones de pertenencia A_{ij} de la entrada se agregan por una conectiva AND (operador de norma triangular), cuyo elemento neutro es el uno. Por

tanto, si una variable de entrada no tiene efecto sobre una salida, sus funciones de pertenencia A_{ij} deberán ser uno en todo el rango de x_j . En otras palabras, para cada variable de entrada x_j , el rango en el que su función de pertenencia A_{ij} es uno, puede ser un buen índice de la importancia de esa entrada en la i -ésima regla. Si el rango es igual a todo el rango de x_j , entonces x_j no tiene efecto en la i -ésima regla; si esto ocurre para las otras reglas del modelo borroso, entonces x_j no tiene efecto en el sistema y puede ser eliminada.

Como resumen de lo expuesto, se podría decir que en el modelo borroso de un sistema, la condición necesaria y suficiente para que una variable de entrada sea no-significativa, es que su función de pertenencia convexa sea igual a uno en todo su dominio y para todas las reglas. De forma gráfica, esto queda representado en la figura D.3.

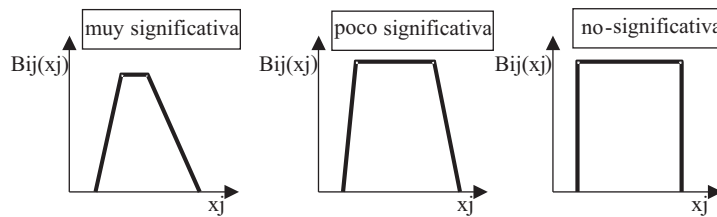


Figura D.3: Ilustración cualitativa del efecto de la variable de entrada x_j en la regla i -ésima.

Para cuantizar la idea expuesta, se define un índice π_j como medida de no-significatividad de la variable x_j en el sistema borroso de la forma:

$$\pi_j = \prod_{i=1}^k \frac{\Gamma_{ij}}{\Gamma_j} \quad (D.2)$$

donde:

- Γ_{ij} es el rango en el que A_{ij} es uno,
- Γ_j es todo el rango de la variable x_j .

En el dominio $[0,1]$, pequeños valores de π representan una variables más efectiva, y viceversa. Mientras π_j representa el efecto global de x_j en el sistema, cada Γ_{ij}/Γ_j da información sobre el efecto de x_j en la i -ésima regla.

Ambos métodos (regularidad y no significatividad) requieren la construcción de modelos borrosos en diversas etapas para cada una de las entradas o combinaciones de ellas, pero el segundo de los métodos presentados tiene un parámetro adicional que, a priori, no parece que sea precisamente fácil de determinar, y es el valor de umbral de π para el cual se considera una entrada significativa o no. La única ventaja que se destaca del método propuesto por Emami es que el tiempo global de cálculo es menor.

D.2.2. Representación de la dinámica del proceso

En este paso se transforma el problema de identificación de un sistema dinámico en un problema de regresión estática. La idea es realizar alguna transformación desde el dominio temporal al de las variables de estado, de manera que éstas determinen el estado del sistema sin necesidad del tiempo. Los métodos usualmente empleados en el modelado dinámico mediante sistemas borrosos fueron vistos en 5.3.

En el caso de tratar con sistemas discretos, una solución sencilla y que da buenos resultados consiste en tratar todas las muestras disponibles de una misma variable u como variables independientes ($u(t)$, $u(t-1)$, $u(t-2)$, ...), con lo que todo sistema se convierte en estático. A estas "nuevas" variables se les aplica el método que se desee de selección de variables.

D.2.3. Granularidad del modelo borroso

Este parámetro está relacionado tanto con el número de reglas como con el número de términos lingüísticos para cada variable. Es el paso que en la literatura se le llama normalmente *identificación de la estructura*. El valor que inicializa el proceso de determinación de estos valores, si se usan métodos de agrupamiento como los vistos en el tema 6, es el número de clases inicial (ver punto C.8). Con los métodos de simplificación de modelos borrosos que se verán en D.5 se disminuirá, si es posible, el número de conjuntos borrosos en las reglas número de clases final mediante medidas de similaridad y el número de reglas con técnicas de detección de redundancia.

D.3. Agrupación de los datos

Este paso es el que encaja perfectamente con los métodos de agrupamiento vistos en el tema C. La idea es utilizar algún algoritmo de agrupamiento de los vistos con el fin de localizar clases de elementos en el espacio de datos elegido (entradas como espacio N-dimensional, entrada a entrada independientemente, salidas, entradas y salidas, combinaciones de entradas de dos en dos, de tres en tres, ...). Cada uno de los grupos de datos detectados tiene, generalmente, un comportamiento similar y se podrá obtener en un paso posterior un modelo común a todos ellos, lineal o no. Sin embargo, este método tiene sus limitaciones (Klawonn and Kruse, 1995a) que se estudian a continuación.

D.3.1. Disposición de los datos

La primera limitación puede darse por la disposición de los datos con los que se va a trabajar, ya que su distribución suele ser continua y esto se adecua más a la regresión que al agrupamiento. Esto puede ser solucionado con un preprocesado adecuado de los datos antes de comenzar a realizar la agrupación por clases de los datos.

Conviene resaltar que, si bien preprocesar los datos puede ser absolutamente necesario en ciertos casos, puede ser interesante aplicarlo de forma generalizada previamente a la inicialización del algoritmo de agrupamiento, de manera que el algoritmo de agrupamiento empleado se encuentre con unos datos ya organizados. Así el algoritmo de

agrupamiento exige menos recursos y el resultado mejora. Ejemplo de esta técnica es el agrupamiento direccional (Hirota and Pedrycz, 1996), que busca una relación entre las variables de entrada y salida, mediante alguna técnica de agrupamiento sencilla y rápida, eligiendo los prototipos de cada clase para realizar un modelo más detallado mediante la unión de las clases de la primera etapa. El efecto es el de realizar una especie de filtrado de los datos, tomando para la realización del modelo final solo los mejores datos.

De todas formas hay que tener en cuenta que el preprocesado también puede tener efectos perniciosos en el agrupamiento (ver apartado D.4.1), como cambiar las variables del sistema por otras sin sentido e imposibles de interpretar.

Una última tendencia en cuanto a modificaciones en algoritmos de agrupamiento, es cambiar el algoritmo de forma que se facilite la siguiente tarea en el proceso de identificación de modelos borrosos (determinación de un modelo borroso inicial que se verá en D.4). En esta línea se pueden encontrar dos opciones.

Preprocesamiento avanzado de datos

Los datos son agrupados mediante FMLE y las clases obtenidos se separan entre ellos mediante el giro de un ángulo δ , dejándolos así paralelos a los ejes. Para el cálculo de δ se usa el siguiente algoritmo (Genther and Glesner, 1997):

- Calcular los autovalores y autovectores de la matriz borrosa de covarianzas de las clases c_a y c_b de dos entradas x e y .
- Calcular la dirección (ángulo con la horizontal) α_a de los autovectores dla clase c_a .
- Calcular la dirección α_b de los autovectores dla clase c_b .
- Calcular la dirección α_{ab} de la línea de separación de las clases c_a y c_b (perpendicular a la línea que une los centros de ambas clases).
- Calcular las relaciones entre autovalores de la clase $r_a = \text{av}_{amax} / \text{av}_{amin}$ y $r_b = \text{av}_{bmax} / \text{av}_{bmin}$.
- Tomar $r_{max} = \max \{r_a, r_b\}$.
- Calcular el ángulo β que minimice

$$\omega_a \min\{(\alpha_a - \beta)^2, (\pi/2 - (\alpha_a - \beta))^2\} + \omega_b \min\{(\alpha_b - \beta)^2, (\pi/2 - (\alpha_b - \beta))^2\} \quad (\text{D.3})$$

donde $\omega_a = r_a - 1$ y $\omega_b = r_b - 1$.

- Si las dos clases están asignados a clases diferentes (salidas diferentes), calcular el ángulo γ_{ab} que minimice

$$(r_{max} - 1) \min\{(\beta - \gamma_{ab})^2, (\frac{\pi}{2} - (\beta - \gamma_{ab}))^2\} + \min\{(\alpha_{ab} - \gamma_{ab})^2, (\frac{\pi}{2} - (\alpha_{ab} - \gamma_{ab}))^2\} \quad (\text{D.4})$$

si no tomar $\gamma_{ab} = \beta$.

- El cálculo definitivo del ángulo entre las entradas x e y se hace calculando los ángulos para todas las combinaciones a, b posibles y calculando el δ que minimice

$$\sum_{a=1}^{c-1} \sum_{b=a+1}^c \min\{(\gamma_{ab} - \delta)^2, (\frac{\pi}{2} - (\gamma_{ab} - \delta))^2\} \quad (D.5)$$

Al rotar los datos en el plano dado por las variables de entrada x e y elegidas puede influenciar los ángulos calculados para otras combinaciones de variables de entrada, el procedimiento expuesto hasta aquí debe ser aplicado a los datos de forma iterativa, resultando el algoritmo siguiente:

- Calcular el ángulo δ para cada combinación de variables de entrada con el algoritmo anterior.
- Elegir el espacio xy en el que δ sea máximo.
- Si el δ elegido es mayor que cierto límite dado rotar los datos en $xy - \delta$, si no acabar.
- Volver al primer paso.

Modificación de algoritmos existentes

Los algoritmos GK y GG se han de modificar de forma que las clases encontrados tengan siempre sus ejes paralelos a los ejes de coordenadas, en lugar de buscar clases de forma arbitraria ((Klawonn and Kruse, 1995b), (Klawonn and Keller, 1997) y (Klawonn and Kruse, 1997)). Esto significa que se debe restringir la matriz borrosa de covarianzas del algoritmo a la forma diagonal. Por tanto, es necesario obtener nuevas fórmulas para el cálculo de dichas matrices. Para el algoritmo GK visto se obtiene:

$$F_i^{(v)} = \frac{\left(\rho_i \prod_{j=1}^n \sum_{k=1}^N (\mu_{ik})^m (z_{kj} - c_{ij})^2 \right)^{1/n}}{\sum_{k=1}^N (\mu_{ik})^m (z_{kv} - c_{iv})^2} \quad (D.6)$$

para el v -ésimo elemento de la matriz diagonal borrosa de covarianzas $F_i^{(v)}$.

En el caso del algoritmo GG que también se ha visto que el cálculo se ha de realizar mediante:

$$F_i^{(v)} = \frac{\sum_{k=1}^N (\mu_{ik})^m}{\sum_{k=1}^N (\mu_{ik})^m (z_{kv} - c_{iv})^2} \quad (D.7)$$

para el v -ésimo elemento de la matriz diagonal borrosa de covarianzas $F_i^{(v)}$.

D.3.2. Espacio de datos

Otro problema surge con la decisión del espacio de datos sobre el que realizar el agrupamiento, ya que los datos tienen significado en un espacio que incluye todas las entradas y salidas al sistema. La partición de los datos en, por ejemplo, pares entrada-salida y la posterior proyección de las clases puede llevar a solapamientos y dificultad en la interpretación. De todas formas esto es más una dificultad inherente del proceso de identificación que un escollo insalvable. De hecho, en (Delgado *et al.*, 1997) se utiliza la elección del espacio de datos a los que aplicar el agrupamiento como una variable más, en este caso, para obtener modelos de mayor o menor carga computacional. En dicha referencia se plantean cinco formas diferentes de elegir los grupos de datos a los que aplicar el agrupamiento:

- Aplicar agrupamiento en Z^N (espacio de entrada) e Y (espacio de salida) con el fin de detectar los prototipos o centroides de cada clase. Las funciones de pertenencia de cada clase se definirán con alguna medida de distancia al prototipo, creando una regla, mediante proyección de la función borrosa en el espacio de la variable correspondiente, para cada par entrada-salida.
- Aplicar directamente agrupamiento en $Z^N \times Y$ y proyectar cada clase en los dominios que se desee, aproximando las funciones de pertenencia proyectadas por funciones, por ejemplo, trapezoidales. Este método es el que permite una mayor interpretabilidad de las reglas.
- Aplicar agrupamiento en Z^N y $Z^N \times Y$ (espacio conjunto de entradas-salidas) y crear reglas que relacionen la pertenencia a una determinada clase de entradas con una determinada clase de entrada-salida.
- Aplicar agrupamiento, de forma independiente, a cada una de las N entradas (Z_1, \dots, Z_N) y cada una de las P salidas (Y_1, \dots, Y_P). Posteriormente se crean $N \times P$ reglas que relacionen todas las clases de entradas creadas con todas las clases de salidas.
- En el caso de modelos de sistemas muy complejos se recomienda hacer directamente un modelo Takagi-Sugeno, ya que da directamente una regresión lineal en el consecuente. Se propone utilizar algún algoritmo de agrupamiento que busque hiperplanos en $Z^N \times Y$, con lo que ya se asume la linealidad del modelo desde el principio y las clases se ajustarán mejor al propósito final. A cada una de esas clases se le ajusta el consecuente, en principio lineal, por mínimos cuadrados.

La lista anterior está ordenada en orden creciente de carga computacional y la elección de los grupos de datos a los que aplicar el agrupamiento dependerá de la aplicación concreta.

D.3.3. Algoritmo de agrupamiento

Un último problema es la elección del algoritmo de agrupamiento. Las diferencias entre la mayoría de algoritmos de agrupamiento estudiados estriban en el tipo de norma elegida y la forma de la clase, con lo que el mismo grupo de datos puede dar lugar

diferentes agrupaciones si se elige uno u otro método. Ninguno de ellos es, en principio, mejor o peor que los otros, sino que cada uno será apropiado en ciertas circunstancias. La información que a priori se puede tener sobre la forma de las clases puede ser fundamental, ya que se han visto métodos de agrupamiento que buscan elipsoides de tamaños similares, elipsoides de tamaños variables, prototipos lineales, etc. El método más utilizado es el FCM ((Davé and Krishnapuram, 1997), (Hwang and Woo, 1995)), aunque también hay referencias a FMLE (Genther and Glesner, 1997), FCRM (Kim *et al.*, 1997), GK y GG ((Klawonn and Kruse, 1995b), (Klawonn and Keller, 1997), (Klawonn and Kruse, 1997), (Ronen *et al.*, 1998)). Algunos autores desarrollan su propio algoritmo de agrupamiento para la identificación de modelos borrosos en su aplicación concreta, como el agrupamiento hiperelipsoidal (Nakamori and Ryo, 1994), o modifican algoritmos ya existentes como las variaciones de GK y GG en (Klawonn and Kruse, 1995b), (Klawonn and Keller, 1997) y (Klawonn and Kruse, 1997). Concretamente, en el caso de modelado de sistemas no lineales para su posterior aplicación al control del sistema, los métodos de agrupamiento más apropiados si se quieren aplicar técnicas de control lineales a cada grupo de datos, serán aquellos que busquen consecuentes lineales (clase lineales o hiperplanos) como los vistos en el punto C.4. En la tabla 6.1 se hizo una primera selección de algoritmos enfocada a los problemas que trata de resolver esta Tesis, pero el capítulo 7 trata este tema con mayor profundidad desde puntos de vista cuantitativos y cualitativos.

D.4. Determinación de un modelo borroso inicial

La clasificación de los datos realizada con los métodos del apartado anterior nos podría permitir, por ejemplo, determinar un modelo lineal para cada uno de las clases detectados, con lo que ya tendríamos una buena aproximación del sistema no lineal complejo en estudio. Esta determinación de modelos borrosos basados en particiones ha sido muy empleada últimamente (Babuska, 1996). También existe la posibilidad de, por ejemplo, hacer un modelo basado en redes neuronales para cada clase (Ronen *et al.*, 1998) siendo la salida total del sistema una ponderación del resultado obtenido para cada red.

En cuanto a los modelos borrosos de reglas basados en esas particiones, en principio, cualquier tipo de modelo borroso es válido (Takagi-Sugeno, lingüístico, relacional). Se pueden encontrar referencias de todos ellos, aunque es más utilizado el modelo lingüístico ((Sugeno and Yasukawa, 1993), (Hong and Lee, 1996), (Genther and Glesner, 1997), (Klawonn and Keller, 1997), (Klawonn and Kruse, 1997), (Hwang and Woo, 1995)) que el de Takagi-Sugeno ((Nakamori and Ryo, 1994), (Babuska, 1996), (Kim *et al.*, 1997)).

Aunque la opción que se persigue en el presente trabajo es la de obtener modelos borrosos de Takagi-Sugeno cuyos consecuentes sean modelos locales afines, de forma que el modelo sea equivalente a los vistos en el capítulo 3, por su similitud y relación se va a ver la forma de obtener tanto los modelos de Takagi-Sugeno como los lingüísticos.

D.4.1. Modelos borrosos de Takagi-Sugeno

En este caso se busca un modelo basado en reglas del tipo (5.2). Sin pérdida de generalidad, se va a suponer que el consecuente de la regla es una función lineal, con lo que tan solo sería necesario estimar dos parámetros en el consecuente (5.3). Las reglas que se proponen son:

$$R_i: \text{Si } x \text{ es } A_i \text{ entonces } y_i = a_i^T x + b_i, \quad i = 1, 2, \dots, c;$$

donde a_i es un vector de parámetros y b_i un escalar. A este modelo se le llama *modelo afín de Takagi-Sugeno*. Por tanto, llegados a este punto, será necesario estimar funciones de pertenencia para las variables A_i del antecedente y los parámetros a_i y b_i del consecuente para cada regla de las c que se determinen con los métodos descritos en D.2.3.

Funciones de pertenencia del antecedente

Desde los resultados del proceso de agrupamiento de los datos, se tienen dos opciones para obtener funciones de pertenencia del antecedente:

- Proyectar los conjuntos borrosos definidos punto a punto en la matriz de partición sobre las variables del antecedente, como en (Sugeno and Yasukawa, 1993), (Genther and Glesner, 1997), (Klawonn and Keller, 1997), (Klawonn and Kruse, 1997), (Delgado *et al.*, 1997) y (Hwang and Woee, 1995).
- Calcular los grados de pertenencia directamente en el espacio de las variables de entrada (Delgado *et al.*, 1997), usando la norma empleada en el algoritmo de agrupamiento.

Generación de funciones de pertenencia del antecedente por proyección.

En este método se proyecta la matriz de partición U obtenida en el proceso de agrupamiento sobre las variables del antecedente de las reglas a construir, que serán las que componen el espacio sobre el que se ha realizado el agrupamiento. Si se proyecta sobre las variables del conjunto de datos inicial, nos encontramos ante una proyección ortogonal, como se puede observar en la figura D.4.

Sin embargo, con una proyección de este tipo se pueden obtener errores en el sentido de pérdida de información (zona sombreada en la figura D.4), si los ejes de la clase no son paralelos (o casi paralelos) a los ejes sobre los que se proyecta. Con la idea de evitar este tipo de problemas se modifican los métodos de agrupamiento, como se comentó en el apartado D.3.1, creando únicamente clases paralelas a los ejes y obteniendo matrices de partición U idénticas al caso anterior.

Otra opción es cambiar las variables del sistema (variables sobre las que se proyecta), intentando explicar el modelo en términos de las variables que definen las direcciones de los autovectores de la matriz que se proyecta (ejes de la clase). Este método tiene la ventaja de que no se pierde información en la proyección, pero puede encontrarse la desventaja de que las nuevas variables no tengan ningún sentido para nosotros o se pierda su interpretación física, haciendo el modelo incomprensible para el experto. La disminución del error de proyección en este método, con el cambio de variables

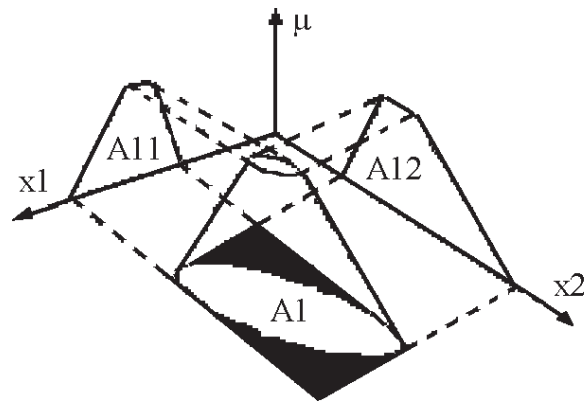


Figura D.4: Proyección ortogonal de una clase del espacio de entrada tomando como ejes las variables de entrada.

pertinente (el correspondiente a los autovectores de U), queda reflejado en la figura D.5. Desde el punto de vista de la identificación esta opción no tiene mucho sentido, ya que se obtendría un sistema explicado en función de una serie de variables sin significado y, lo más probable, que no se pudiesen medir.

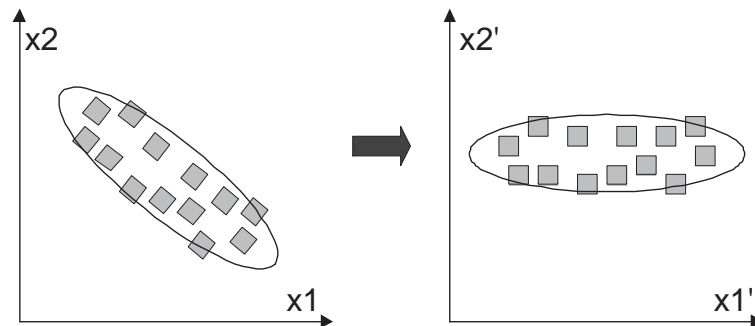


Figura D.5: Proyección de una clase del espacio de entrada tomando como ejes las direcciones de los ejes de la clase.

Se elija el método de generación de funciones de pertenencia que se elija, después del agrupamiento del conjunto de datos Z , para cada dato $z_k^T = [x_{1k}, \dots, x_{nk}, y_k]$ los grados de pertenencia son dados por la matriz de partición borrosa $U = [\mu_{ik}]$. Para z_k estará dado por la k -ésima columna de la matriz de partición $\mu_k^T = [\mu_{1k}, \dots, \mu_{ck}]$. Proyectando la fila i (grado de pertenencia a una clase i de cada dato j) sobre la variable x_j , se obtiene una proyección como la de la figura D.6.a. Evidentemente, esta no es la forma adecuada de una función de pertenencia. Por supuesto, se puede aproximar por el tipo de función convexa que se desee (como en D.5.b), aunque lo más habitual en la bibliografía es utilizar funciones de pertenencia triangulares o, a lo sumo, trapezoidales, como se muestra en la figura D.5.c. Evidentemente las proyecciones (o sus ajustes) no cumplen con la restricción de partición borrosa (suma de pertenencia 1) y este extremo también ha de ser tenido en cuenta.

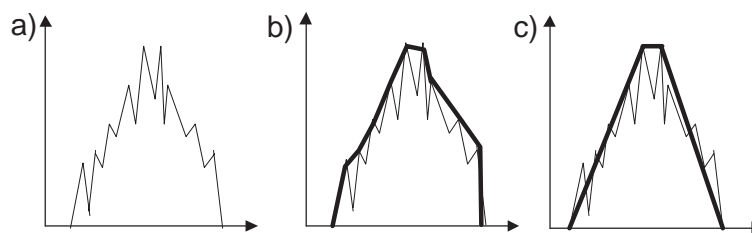


Figura D.6: Función de pertenencia proyectada y aproximaciones.

Generación de funciones de pertenencia del antecedente de forma directa en el espacio de entradas. En este caso, la idea es medir la distancia de cada punto al prototipo de cada clase y transformar la distancia en un grado de pertenencia aplicando alguna expresión del tipo (C.18) en FCM o expresiones similares en otros algoritmos de agrupamiento, con lo que se da un valor de pertenencia de cada punto a cada clase. Esto conlleva la desventaja de su alta carga computacional pues, para evaluar la pertenencia de nuevos datos, cada vez hay que buscar entre los valores de las funciones de pertenencia identificadas el punto más cercano al cual se pretende evaluar la pertenencia, asignando esa pertenencia o bien interpolando entre varias pertenencias si hay más de un punto a la misma distancia.

Una segunda opción consiste en intentar ajustar los parámetros de alguna superficie con forma similar a las funciones de pertenencia identificadas. De este modo, si la superficie a la que se decide ajustar no es la correcta se puede perder información y unos buenos modelos lineales pueden llegar a componer un mal modelo no lineal por su incorrecta interpolación.

En ambos casos las reglas serán proposiciones de conjuntos borrosos multidimensionales.

Parámetros del consecuente

Para la estimación de los parámetros del consecuente del modelo borroso de Takagi-Sugeno, la solución habitual (Babuska, 1996) es aplicar a los resultados del agrupamiento técnicas de mínimos cuadrados que minimicen el error de predicción del modelo global o de cada uno de los subespacios aproximados por cada clase. Otra opción bastante común es plantear problemas de optimización que sean resueltos por algoritmos genéticos, como en (Delgado *et al.*, 1997) y (Hwang and Wooc, 1995).

Cuando se usa para la agregación de los resultados de los consecuentes de las reglas borrosas la fórmula de la media borrosa (5.4), la estimación de los parámetros del consecuente se puede obtener resolviendo un problema lineal de mínimos cuadrados. El grado de cumplimiento de la i -ésima regla $\beta_i(\mathbf{x})$ se puede obtener mediante la matriz de partición borrosa U . Cada fila de U (6.15) contiene la definición, punto a punto, de la función de pertenencia de los datos en el espacio entradas-salidas. Para obtener la función de pertenencia A_i en el espacio del regresor (entradas), la i -ésima fila de U deberá de proyectarse en el espacio del regresor:

$$\beta_{ik} = \text{proj}_{reg}(\mu_{ik}), \quad k = 1, \dots, N, \quad (\text{D.8})$$

Para escribir (5.4) en forma matricial para todos los datos (x_k, y_k) , con $1 \leq k \leq N$, se define una matriz diagonal Γ_i en el espacio $N \times N$

$$\Gamma_i = \begin{bmatrix} \gamma_{i1} & 0 & \cdots & 0 \\ 0 & \gamma_{i2} & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & \gamma_{iN} \end{bmatrix} \quad (\text{D.9})$$

donde γ_{ik} es un grado de pertenencia normalizado según

$$\gamma_{ik} = \frac{\beta_{ik}}{\sum_{j=1}^N \beta_{jk}} \quad (\text{D.10})$$

Llamando X' a la matriz compuesta por las matrices Γ_i y X_e

$$X' = [(\Gamma_1 X_e); (\Gamma_2 X_e); \dots; (\Gamma_c X_e)] \quad (\text{D.11})$$

con X_e la matriz regresora (vectores de entradas) extendida por una columna unitaria

$$X_e = \begin{bmatrix} x_1^T & 1 \\ x_2^T & 1 \\ \vdots & \vdots \\ x_N^T & 1 \end{bmatrix} \quad (\text{D.12})$$

y denotando como θ' a

$$\theta' = [\theta_1^T; \theta_2^T; \dots; \theta_c^T]^T \quad (\text{D.13})$$

donde $\theta_i = [a_i^T; b_i]$ para $1 \leq i \leq c$. El problema resultante de mínimos cuadrados es

$$y = X' \theta' + \varepsilon \quad (\text{D.14})$$

y su solución la da:

$$\theta' = [(X')^T X']^{-1} (X')^T y \quad (\text{D.15})$$

Los parámetros a_i y b_i se obtienen de (D.13) mediante:

$$a_i = [\theta'_{q+1}; \theta'_{q+2}; \dots; \theta'_{q+p}]^T, \quad b_i = [\theta'_{q+p+1}] \quad \text{con } q = (i-1)(p+1) \quad (\text{D.16})$$

Llegados a este punto, se ha construido un modelo de Takagi-Sugeno cuya inferencia se reduce a evaluar (5.4) o alguna expresión alternativa de características similares 5.2.

D.4.2. Modelos borrosos lingüísticos

El interés de este tipo de modelos está debido en gran medida a su alta interpretabilidad, aunque la aplicación al modelado para control no tenga demasiado sentido. De todas formas, su construcción desde las particiones borrosas es sencilla.

Funciones de pertenencia del antecedente

En este caso se aplica el mismo método que el visto en D.4.1, ya que ambos antecedentes tienen la misma estructura en ambos modelos.

Funciones de pertenencia del consecuente

Con el modelo lingüístico, el consecuente tiene una estructura igual a la del antecedente, por lo cual los métodos válidos para la estimación de funciones de pertenencia del antecedente son válidos para las funciones de pertenencia del consecuente. En la figura D.7 podemos ver dos sencillos ejemplos.

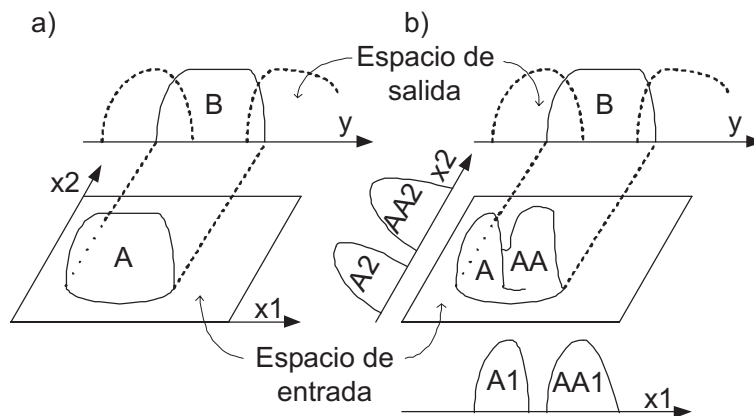


Figura D.7: Función de pertenencia del consecuente.

De la figura D.7.a se obtendría una regla como:

R: Si x_1 es A_1 y x_2 es A_2 entonces y es B

De la figura D.7.b se obtendrían un par de reglas:

R₁: Si x_1 es A_1 y x_2 es A_2 entonces y es B

R₂: Si x_1 es AA_2 y x_2 es AA_1 entonces y es B

La posterior elección de mecanismo de inferencia, operadores conectivos, método de desborrosificación es amplia y depende del usuario y la aplicación (Wang, 1997).

D.5. Simplificación del modelo borroso obtenido

El proceso de simplificación puede ser cedido a un experto, o bien puede ser automatizado. El empleo de CCM (punto C.8.1) puede simplificar el modelo enormemente. Pero si lo que se pretende es identificar conjuntos borrosos similares (que se solapan y describen prácticamente la misma región) en el modelo borroso inicial y reemplazarlos por un conjunto borroso común que represente a los originales, se pueden usar medidas

de similaridad (similaridad, matemáticamente hablando, se refiere a algún tipo de norma). De este modo se reduce el número de conjuntos borrosos en el modelo e incluso, si la redundancia es alta, puede dar lugar a eliminar reglas similares.

Un algoritmo de simplificación de una base de reglas sería (Babuska, 1996):

1. Elegir unos valores umbrales de similaridad $\mu, \mu_r \in (0,1)$ para unir conjuntos borrosos similares entre sí o similares al conjunto universal, respectivamente.
2. Selección de los dos conjuntos borrosos más similares en la base de reglas, usando una medida de similaridad $s_{ijk} = S(A_{ij}, A_{kj})$.
3. Unir los dos conjuntos borrosos más similares y actualizar la base de reglas. Para ello, si se da $S(A_{lq}, A_{mq}) \geq \mu$ se unen A_{lq} y A_{mq} , creando un nuevo conjunto borroso A y sustituyendo $A_{lq} = A$ y $A_{mq} = A$.
4. Si queda algún conjunto borroso que cumpla $s_{ijk} \geq \mu$, volver al paso 2. Si no, continuar.
5. Eliminar conjuntos borrosos similares al conjunto universal. Para esto se calcula la similaridad de cada conjunto borroso existente en este momento, con el conjunto U tal que $\mu_U = 1$. Si $S(A_{ij}, U) \geq \mu_r$, se elimina A_{ij} del antecedente de la i -ésima regla.

Aunque, en principio, cualquier medida de similaridad sería válida, en (Babuska, 1996) se referencia a (Setnes, 1995) como un extenso estudio sobre medidas de similaridad.

D.6. Validación del modelo

La validación final del modelo borroso obtenido, puede ser realizada de muchas formas, o por las combinaciones convenientes de aquellas aplicables al caso en estudio. La más habitual es una validación mediante simulaciones numéricas y comparación con datos reales del proceso, como se realiza en la mayoría de la bibliografía.

Un tema de especial importancia es la interpretabilidad del modelo borroso obtenido pues, como se ha comentado, esta es la característica más reseñable de los sistemas inteligentes y no su capacidad de aproximadores universales.

El análisis del espacio de entrada que cubre la base de reglas no debe ser olvidado. Si una base de reglas es incompleta, se pueden añadir reglas por conocimiento del sistema o mediante un modelo lineal del sistema en el entorno que no había sido cubierto por las reglas iniciales.

D.7. Software para la identificación sistematizada de modelos borrosos

En este apartado se muestra una herramienta software para realizar de forma sencilla la identificación de sistemas desde datos experimentales según la estructura propuesta en 7.1.

Con la idea de no tener que programar desde el principio cada uno de los pasos del algoritmo que implementase el método sistematizado, se realizó una revisión de los paquetes de software existentes que estén relacionados con el tema (Díez, 1999). La FMID (Fuzzy Modeling and Identification) Toolbox para Matlab es un conjunto de funciones para la construcción de modelos borrosos de Takagi-Sugeno desde datos experimentales. La toolbox proporciona herramientas para la generación automática de modelos en estructuras de Matlab desde datos entrada-salida, para la simulación de los modelos obtenidos, para la exportación de los modelos obtenidos a ficheros de Latex, así como una demostración del funcionamiento de la toolbox en la identificación de modelos estáticos y dinámicos. Además, también están disponibles numerosos ejercicios que fueron creados posteriormente a la toolbox.

El algoritmo (Babuska, 1996) en el que se basa la toolbox encaja con el método para la determinación de modelos borrosos definido: dada una serie de datos entrada-salida (la toolbox no determina cuales son importantes para el sistema y cuales no) y el número de clases con los que se quieren agrupar los datos mediante GK (este valor también debe de ser suministrado por el usuario), genera un modelo borroso del sistema en estudio bien por proyección de las clases borrosos generados, bien manteniendo una función de pertenencia borrosa en el espacio multidimensional de las entradas.

FMID se podría emplear como base para iniciar la implementación de una herramienta sistematizada de identificación de modelos borrosos mediante agrupamiento. La toolbox es gratuita para fines de investigación, funciona con Matlab 5 o versiones posteriores (de las cuales se dispone), los códigos pueden ser editados y modificados sin problema alguno, y se puede descargar.

Disponible la FMID toolbox, para obtener una herramienta que responda a la estructura algorítmica deseada, será necesario implementar nuevas funciones que encajen con aquellas de interés y ya existentes en la FMID. Estas funciones responderán al método de agrupamiento FCM, y a los algoritmos de selección de entradas y selección del número de clases.

D.7.1. Funciones de interés de FMID

No todas las funciones y utilidades de FMID son de interés para la construcción de la herramienta de identificación automática que se pretende crear. Conviene revisar la forma de actuar de las que si que son de interés, de forma que las nuevas funciones a crear no tengan ningún problema de compatibilidad con las ya existentes.

fmclust

Esta función construye un modelo borroso multivariable de Takagi-Sugeno desde datos entrada-salida mediante el algoritmo de agrupamiento de Gustafson-Kessel. La forma de la función es:

$$[FM, Mu, Z] = FMCLUST(DAT, PAR, DYN) \quad (D.17)$$

donde:

DAT es una estructura con los siguientes campos

U,Y matrices de datos de entrada y salida, respectivamente
 N números de datos en cada ensayo
 Ts periodo de muestreo (opcional, 1 por defecto)
 PAR es una estructura con los siguientes campos
 c número de clases por salida
 m parámetro de borrosidad (2 por defecto)
 tol tolerancia de finalización (0.01 por defecto)
 seed semilla para la inicialización (por defecto sum(100*clock))
 ante ... antecedentes:
 1.- fdp en el espacio de entradas (por defecto)
 2.- fdp proyectadas
 DYN es una estructura con los siguientes campos
 Ny retardo en y (0 por defecto)
 Nu retardo en u (1 por defecto)
 Nd retardo de transporte (0 por defecto)
 FM es una estructura que contiene el modelo borroso (ver FMSTRUCT)
 Mu es una matriz con las matrices de partición borrosas
 Z es una matriz que contiene los datos a los que aplicar el agrupamiento

fmsim

Esta función simula un modelo borroso MIMO. La forma de esta función es:

$$[Ym, q, DOF, Yl, Ylm] = FMSIM(U, Y, FM, Ymin, Ymax, show, H) \quad (D.18)$$

donde:

U,Y matrices de datos de entrada y salida, respectivamente (los datos de salida se necesitan para la inicialización y comparación)
 FM estructura con los parámetros del modelo borroso
 Ymin restricción del límite inferior de Y (opcional)
 Ymax restricción del límite superior de Y (opcional)
 Show 1 para gráficos on-line, 2 para sólo gráfico final y 0 para no dibujar gráficos (opcional, 1 por defecto)
 H horizonte de predicción (aún sin desarrollar)
 Ym salida simulada del proceso
 q índice de funcionamiento del modelo
 DOF grado de cumplimiento de las reglas
 Yl salida del consecuente de cada regla de forma independiente
 Ylm como Yl pero con datos con DOF ¡0.5 ocultos por NaN)

plotmfs

Esta función dibuja las funciones de pertenencia de un modelo borroso MIMO almacenado en una estructura del tipo FM. Las funciones de pertenencia proyectadas se

dibujan directamente. En cambio, ya que es imposible visualizar las funciones de pertenencia en un espacio N-dimensional, se dibujan sus proyecciones para hacerse una idea de su aspecto. La forma de esta función es:

$$PLOTMFS(FM) \quad (D.19)$$

D.7.2. Nuevas funciones

Dado que las funciones del apartado anterior no completan el algoritmo buscado, nuevas funciones, que responden a los métodos de agrupamiento FCM, FCRM, AFRCR y AFCRC, así como a los algoritmos de selección de entradas y selección del número de clases han de ser implementadas.

fcf

El algoritmo FCM ya está implementado en la Fuzzy Toolbox de Matlab, de la cual se dispone y, por tanto, no será implementada. Esta función aplica el método de agrupamiento FCM a un conjunto de datos dado. La forma de esta función es:

$$[CENTER, U, OBJ_FCN] = FCM(DATA, CLUSTER_N) \quad (D.20)$$

donde:

- DATA es el conjunto de datos a clasificar (cada fila es una muestra)
- CLUSTER_N número de clases (mayor que 1)
- CENTER centros finales de las clases (cada fila es un centro)
- U matriz borrosa de partición final
- OBJ_FCN valores de la función objetivo a lo largo de la iteración

afcr

Los algoritmo FCRM, AFRCR y el nuevo AFCRC han sido implementados con la siguiente estructura:

$$[U, BETA, CENTROIDES, J_{EVOL}] = FCM(DATOS, CLUSTERS, PARAM, M) \quad (D.21)$$

donde:

- DATOS es el conjunto de datos a clasificar (cada fila es una muestra, cada variable una columna, y la última de ellas la de salida)
- CLUSTERS número de clases (mayor que 1)
- PARAM vector de parámetros del algoritmo $[\nu, \sigma_1, \sigma_2, \gamma_1, \gamma_2]$ (con $\gamma_1 = \gamma_2 = 0$ se tiene el algoritmo AFRCR, y si además $\nu = 0$, se tiene FCRM)
- M parámetro de borrosidad (2 por defecto)
- U matriz borrosa de partición final
- BETA parámetros de los modelos locales lineales (por fila)
- CENTROIDES puntos prototipo de las clases (por fila)
- J_{EVOL} valores de la función objetivo a lo largo de las iteraciones

nclust_sy

Aplica el criterio de Sugeno (y Yasukawa) para determinar el número de clases óptimo conocidos el parámetro de borrosidad m y una norma B . La forma de esta función es:

$$[OUT] = NCLUST_SY(DATA, M, B) \quad (D.22)$$

$NCLUST_SY$ calcula el número óptimo de clases en el que dividir el conjunto de datos $DATA$ y lo devuelve en la variable OUT . La variable $DATA$ tendrá la estructura de una variable por columna y una muestra por fila. La variable M es el índice de borrosidad a emplear en el agrupamiento (2 por defecto). La matriz B determina la forma de las clases a buscar (matriz identidad por defecto, que induce clases hiperesféricas). OUT será un número entero representando el valor más adecuado con el que inicializar el proceso de agrupamiento de los datos.

Para la determinación del número de clases se usa la norma determinada por B (o la euclídea por defecto) y se realiza agrupamiento a los datos usando Fuzzy C-Means. Se da como valor óptimo el del número de clases que minimiza la función de (2.21).

nclust_em

Aplica el criterio de Emami para determinar el número de clases óptimo conocidos el parámetro de borrosidad m y una norma B . La forma de esta función es:

$$[OUT] = NCLUST_EM(DATA, M, B) \quad (D.23)$$

$NCLUST_EM$ calcula el número óptimo de clases en el que dividir el conjunto de datos $DATA$ y lo devuelve en la variable OUT . La variable $DATA$ tendrá la estructura de una variable por columna y una muestra por fila. La variable M es el índice de borrosidad a emplear en el agrupamiento (2 por defecto). La matriz B determina la forma de las clases a buscar (matriz identidad por defecto, que induce clases hiperesféricas). OUT será un número entero representando el valor más adecuado con el que inicializar el proceso de agrupamiento de los datos.

Para la determinación del número de clases se usa la norma determinada por B (o la euclídea por defecto) y se realiza agrupamiento a los datos usando Fuzzy C-Means. Se da como valor óptimo el del número de clases que minimiza la función de (2.26).

bestdata_1

Determina la mejor entrada de entre las disponibles. La forma de esta función es:

$$[OUT, BEST_1] = BESTDATA(DATA, M) \quad (D.24)$$

Gracias a esta función se puede determinar la entrada que es capaz de explicar mejor la salida de un sistema MISO. Para ello utiliza el criterio de regularidad de Sugeno (2.1). La variable $DATA$ tendrá la estructura de una variable por columna y una muestra por fila. La última columna de $DATA$ debe corresponderse con los datos de la variable de salida. La variable M es el índice de borrosidad a emplear en el agrupamiento (2 por

defecto). OUT será una matriz con los valores que da el criterio de regularidad (RC) para cada una de las variables de entrada. La entrada con menor valor de RC, almacenado en la variable de salida BEST_1 indicará que dicha entrada explica mejor que ninguna otra la salida.

Esta función es la que mayor carga computacional tiene en todo el proceso de identificación del modelo borroso. La mera observación de (2.1) lleva a la conclusión de que se han de construir dos modelos borrosos para cada una de las posibles combinaciones de entradas al sistema que se desee estudiar. Por tanto, dentro de la función BESTDATA_1, se hacen numerosas llamadas a las funciones fmclust (para construir los modelos), nclust_sy y nclust_em (para inicializar fmclust con un valor correcto de clases).

bestdata_n

Determina la mejor determina la mejor combinación de N entradas de entre las disponibles. La forma de esta función es:

$$[OUT, BEST_N] = BESTDATA_N(DATA, BEST_N_ANT, M) \quad (D.25)$$

Con esta función se puede determinar la combinación de N entradas que es capaz de representar mejor la salida de un sistema MISO. Para ello utiliza el llamado criterio de regularidad empleado por Sugeno (2.1). La variable DATA tendrá la estructura de una variable por columna y una muestra por fila. La última columna de DATA debe corresponderse con los datos de la variable de salida. La variable BEST_N_ANT es un vector fila con los mejores N-1 entradas de las disponibles. La variable BEST_N indicará las columnas en la que están almacenadas las N variables que explican mejor la salida. La variable M es el índice de borrosidad a emplear en el agrupamiento (2 por defecto). OUT será una matriz con los valores que da el criterio de regularidad (RC) para cada una de las combinaciones de variables de entrada. La combinación de entradas con menor valor de RC, almacenadas en la variable de salida BEST_N, indicará que dicha combinación de N entradas explica mejor que ninguna otra la salida.

Por la misma razón que BESTDATA_1, BESTDATA_N tiene una gran carga computacional.

D.7.3. Identificación de sistemas no lineales con I-FMID

Al conjunto de funciones de FMID y las nuevas funciones creadas se la ha llamado I-FMID (Improved – Fuzzy Modeling and Identification) Toolbox para Matlab.

Se deja al usuario la creación de un sencillo programa que, en cada caso, concatene las funciones necesarias e introduzca los datos del proceso en estudio. Para observar el comportamiento de la toolbox en ejemplos adicionales a los descritos en la Tesis, se puede consultar (Díez, 1999) o (Díez and Navarro, 1999).