UNIVERSIDAD DE LEÓN

Departamento de Ingeniería Eléctrica
y de Sistemas y Automática

TESIS DOCTORAL

Técnicas de extracción del conocimiento basadas en data mining visual para la supervisión de procesos industriales. Análisis de la dinámica basado en mapas auto-organizados.

Miguel Ángel Prada Medrano

Junio de 2009

A mis padres

Agradecimientos

En primer lugar, me gustaría mostrar mi más sincero agradecimiento a mis compañeros Manuel Domínguez, Juan José Fuertes y Perfecto Reguera. Sin su dirección, apoyo y consejo esto no habría sido posible. El buen ambiente, el trabajo en equipo y el dinamismo que se respira en el laboratorio son los que hacen posible avanzar. En estos cuatro años no sólo he aprendido más que en cualquier etapa anterior, sino que he madurado como persona y me he sentido muy a gusto.

También debo agradecer su ayuda, apoyo y compañía a los otros becarios del laboratorio, en especial a Antonio, Diego, Serafín y los Robertos. Sin olvidar, por supuesto, a todos aquellos compañeros presentes y pasados con los que también he colaborado en estos años: Juanma, David, Pablo, etc.

Estoy agradecido también a la gente del CNEL de la Universidad de Florida, en especial al Dr. Príncipe, Julie, Sudhir, Sohan, Aaron, Erion, Alex y Memming. Allí aprendí nuevas formas de trabajar, la importancia de los aspectos teóricos y que, en la investigación, el primer paso es siempre plantear las preguntas correctas.

También agradezco la colaboración de los compañeros de la Universidad de Oviedo, en especial de Nacho Díaz, fuente constante de ideas.

A mis padres, por su cariño y apoyo. Sin ellos nada habría sido posible. Y al resto de mi familia, en especial a los que ya no están.

Y a mis amigos, afortunadamente demasiados para citarlos aquí, estén en Puente Castro, León, Coyanza, Madrid o al otro lado del océano.

Resumen

La extracción de conocimiento a partir de grandes volúmenes de datos se presenta como una opción eficaz en el ámbito del análisis y supervisión de los procesos industriales. El mapa auto-organizado (SOM) es un método útil para este propósito, pues permite proyectar, de un modo no supervisado, la información más significativa que se encuentra disponible en un conjunto de datos sobre un espacio de baja dimensión ordenado. Su capacidad para resumir los datos de entrada y preservar su estructura es útil para la creación de modelos locales o la definición de mapas interpretables de un modo consistente. Sus aplicaciones en el área de los procesos industriales se han centrado, hasta el momento, fundamentalmente en características estáticas. No obstante, en el análisis y supervisión de procesos, el conocimiento del comportamiento dinámico resulta clave.

Puesto que el mapa auto-organizado es útil como infraestructura para el modelado dinámico mediante agrupación de modelos locales, en esta tesis se comparan diversas modificaciones del SOM, originalmente propuestas para mejorar su procesamiento temporal, con el objetivo de evaluar su potencial para este propósito. Se utilizan cinco algoritmos representativos de las diferentes estrategias para introducir contexto temporal en el aprendizaje de la red. Estos algoritmos son el SOM con aprendizaje dinámico, el SOM recurrente, SARDNET, Merge SOM y SOMTAD. También se compara el ajuste de los modelos por medio de los vectores prototipo del SOM con el ajuste por medio de los vectores de entrada asociados a cada región local, de acuerdo al espacio de salida de la red. La capacidad de reconstrucción de la dinámica se compara por medio del error de predicción n pasos adelante y de la preservación de los invariantes dinámicos.

La evaluación de estos algoritmos permite reconocer pautas para la correcta selección de los métodos a utilizar e indicios sobre los aspectos clave a considerar para definir nuevos algoritmos. Las modificaciones que incorporan difusión de la actividad, recursividad o regulación del aprendizaje en función del error de predicción proporcionan, en general, mejores resultados que el algoritmo clásico. Por otra parte, el ajuste por medio de los vectores de las regiones de Voronoi proporciona una mejor reconstrucción de la dinámica.

Con el objetivo, de nuevo, de extraer conocimiento acerca de la dinámica de procesos industriales, también se propone aprovechar la naturaleza eminentemente visual de la red SOM para definir mapas de visualización que permitan representar, en todo el rango de operación, características dinámicas relacionadas con dos tareas comúnmente realizadas por los ingenieros de control: el análisis de la respuesta temporal de sistemas monovariable y el análisis del acoplamiento entre lazos y las dificultades de control en sistemas multivariable. Por ello, en el primer caso, se definen mapas que permitan relacionar comportamientos dinámicos específicos expresados en términos habituales (ganancia, tiempo de establecimiento, coeficiente de amortiguamiento, etc.) con regiones o conjuntos de puntos de funcionamiento, de un modo global e intuitivo. En el segundo caso, mediante mapas de ganancias relativas, valores y vectores singulares, número de condición y ceros multivariable, se busca reconocer las dificultades en el control y el acoplamiento entre variables en este tipo de sistemas.

Los mapas permiten descubrir, ampliar o confirmar conocimientos acerca del sistema, abarcando todo el rango de operación. Debido a su consistencia, tanto entre sí como con respecto a otros mapas previamente definidos, resultan una herramienta útil para comparar visualmente el comportamiento de características dinámicas o relacionarlas visualmente con las variables de su punto de funcionamiento.

Abstract

Knowledge extraction from large amounts of data is an effective approach for analysis and monitoring of industrial processes. Self-organizing map (SOM) is a useful method for this purpose, because it makes possible an unsupervised projection of the most important information contained in the data set onto an ordered low-dimensional space. Its ability to summarize the input data set while preserving its structure is useful to build local models or to define interpretable maps in a consistent way. The previous applications in the industrial field have been focused on static features of the processes. Nevertheless, knowledge of dynamic behavior is a key factor in process analysis and monitoring.

Since the self-organizing map is useful as an infrastructure for dynamic modeling by means of multiple local models, several variations of SOM, originally proposed to improve temporal processing, are compared in this dissertation to test their suitability for that purpose. Five algorithms, representative of different approaches to include temporal context during the network learning, have been used for this evaluation: SOM with dynamic learning, Recurrent SOM, SARDNET, Merge SOM and SOMTAD. Two different approaches for model fitting, through the codebook vectors and through the data contained in the Voronoi regions, are also compared. The dynamic reconstruction ability is compared through the n-step ahead prediction error and the preservation of dynamic invariants.

The evaluation of those algorithms lets us discover principles for the correct selection of methods and clues about the essential features to be considered for future algorithms. Variations of SOM that include activity diffusion, recursion or self-regulated learning according to the prediction error provide, in general, better results than the traditional SOM algorithm. On the other hand, model fitting through the vectors contained in the Voronoi regions gives better results for dynamic modeling. With the aim, again, in the extraction of knowledge about the dynamics of industrial processes, it is proposed to exploit the visual nature of SOM to define visualization maps that represent, through the whole operation range, dynamic features related to two usual tasks in control engineering: analysis of the temporal response in SISO systems and analysis of coupling among variables

and difficulties in control in MIMO systems. For that reason, in the first case, I define maps oriented to establish, in a global and intuitive way, relationships between specific dynamic behaviors, expressed in usual terms such as gain, settling time or damping ratio, and regions or sets of operating points. In the second situation, maps of relative gain, singular values and vectors, condition numbers and multivariable zeros are defined to extract knowledge about the coupling of variables and the difficulties in MIMO system control.

Those maps make it possible to discover, increase or confirm knowledge about the system, spanned through the entire operation range. Because of the coherence among maps of dynamics and with respect to other previously defined visualizations, the maps proposed in this dissertation are useful tools to compare dynamic features and link them with the variables of their operating points by simple visual reasoning.

Índice general

Índice de Tablas

Índice de figuras

Capítulo 1

Introducción

1.1. Descripción general

1.1.1. Introducción

En el ámbito industrial, el conocimiento que se requiere de los procesos industriales para su control o supervisión es cada vez más preciso. Sin embargo, muchos procesos industriales presentan gran complejidad: poseen un gran número de variables que se relacionan entre sí de forma no lineal y, en ocasiones, su comportamiento no es estacionario en el tiempo. Esto causa que la obtención de modelos analíticos, basados en ecuaciones explícitas, sea una tarea muy difícil o inviable. Para este propósito, en algunos casos, el conocimiento previo que se posee sobre las leyes que gobiernan el proceso es incompleto (modelos parciales, relaciones entre variables, casos, etc.) mientras que en otros es prácticamente nulo.

No obstante, la instrumentación industrial actual permite disponer de grandes volúmenes de datos de multitud de variables, que implícitamente contienen información acerca del proceso. Por ello, aunque se desconozcan las leyes concretas que rigen el proceso, un análisis detallado de estos datos mediante **métodos basados en los datos del proceso** (Venkatasubramanian *et al.*, 2003b) puede conducir a modelos útiles que permitan comprender mejor el sistema.

Sin embargo, la gran disponibilidad de datos constituye en sí misma un problema asociado a este enfoque, ya que el operador o ingeniero no puede evaluar mediante la simple observación semejante cantidad de información, cuya complejidad es excesiva para los métodos tradicionales. Es necesario utilizar otro tipo de técnicas que se enmarcan dentro del área de ***data mining*** y **descubrimiento de conocimiento** (*Knowledge Discovery and Data Mining*, KDD), cuyo objetivo es la extracción no trivial de conocimiento implícito, potencialmente útil y previamen-

te desconocido que permanece oculto entre grandes cantidades de datos (Fayyad *et al.*, 1996). Al igual que en otros campos, la utilización de *data mining* en el área de los procesos industriales es creciente. Debido a los avances en este campo y a la mayor disponibilidad de recursos de procesamiento y almacenamiento, estos métodos tienden a ser más utilizados en la actualidad. En efecto, el proceso de descubrimiento de conocimiento ha sido bastante estudiado y existen metodologías que describen las tareas desde la preparación de los datos a la comprensión del negocio y la implantación, como el modelo CRISP-DM (Chapman *et al.*, 2000), u otros orientados a la preparación de datos y el modelado, como el proceso KDD (Fayyad *et al.*, 1996). También se han definido estándares para la definición e intercambio de datos (Grossman *et al.*, 1999) como el *Predictive Model Markup Language* (PMML) .

Las técnicas que se pueden utilizar en este marco son muy diversas e incluyen desde métodos basados en la estadística a redes neuronales. En general, los métodos aplicables a la extracción del conocimiento se pueden clasificar como algoritmos de ***Machine Learning*** o **aprendizaje máquina** (Alpaydin, 2004). Como su propio nombre indica, el objetivo de estas técnicas es optimizar un objetivo o adaptarse al mismo, aprendiendo a partir de los datos. Como característica general, se puede decir de estas técnicas que son genéricas y polivalentes, aplicables a diversos tipos de sistemas.

Comúnmente, de entre estos algoritmos se utilizan técnicas de **aprendizaje no supervisado**, que no requieren especificar salidas deseadas ni obtener refuerzos del entorno, puesto que su objetivo es obtener una representación fiel de la entrada que se adecúe a los propósitos. Esto encaja con un objetivo de la extracción del conocimiento, que es llevar a cabo una **reducción de la dimensión** (Carreira-Perpiñán, 1996), es decir, una proyección del espacio de entrada en un espacio de salida de baja dimensión que conserve la información más significativa, y una **compresión de los datos**.

1.1.2. La supervisión y el análisis de procesos industriales

La transformación que los métodos de reducción de la dimensión llevan a cabo en los datos permite facilitar la **supervisión** del proceso. Puesto que el objetivo de la supervisión es determinar la condición de un sistema físico, a partir del reconocimiento e indicación de anomalías en su comportamiento, un enfoque útil puede ser la transformación de los datos en representaciones visuales que permitan a un supervisor humano comprender más fácilmente el proceso y reconocer situaciones importantes. Esta estrategia de *data mining* basada en la exploración visual de datos se denomina ***Visual Data Mining*** (Keim, 2002; Ferreira de Oliveira y Levkowitz, 2003).

Las técnicas de *visual data mining* llevan a cabo una reducción de la dimensión sobre los datos de entrada que permite conservar la información más relevante en un conjunto de datos reducido y de baja dimensión. De ese modo, proporcionan una instantánea del conjunto de datos que permite a los analistas del proceso extraer conocimiento acerca de sus condiciones y su evolución a lo largo del tiempo. Su principal objetivo es, por tanto, integrar más a la persona en el proceso de exploración de datos y explotar sus capacidades de percepción visual y razonamiento con objetos visibles.

Una de las tareas más importantes de la supervisión de procesos es la detección e identificación de fallos que pongan en peligro su correcta ejecución. Los **fallos** se pueden definir como desviaciones de al menos una característica o variable del sistema respecto al comportamiento considerado como aceptable o nominal y pueden clasificarse, con respecto a sus características, como abruptos, incipientes o intermitentes, o, respecto al modelo del proceso, como aditivos o multiplicativos.

En el proceso de tratamiento de los fallos, se pueden diferenciar varias tareas (Isermann y Ballé, 1997). Una de ellas es la **detección**, que consiste en determinar su existencia a partir de los síntomas. La determinación de la localización del fallo recibe el nombre de **aislamiento**, mientras que la determinación de su importancia y comportamiento en el tiempo se denomina **identificación**. En conjunto, se puede hablar de un proceso de diagnóstico de fallos que consta de tres fases diferenciadas:

- En primer lugar, se generan los **residuos** o síntomas, que son indicadores de fallo obtenidos a partir de una desviación entre las medidas y los valores predichos. La generación de residuos puede basarse en el principio de redundancia analítica, cuando existen ecuaciones explícitas que lo permiten, pero también puede obtenerse a partir de modelos basados en datos. En este último caso, los residuos se obtienen a partir de la comparación entre los valores reales y los aprendidos o entrenados.

- En segundo lugar, se lleva a cabo una evaluación de los residuos o clasificación de fallos. Este proceso se vale de conocimiento previo acerca del sistema o de los propios fallos.

- En tercer lugar, se analizan los fallos. En general, esta tarea se afronta de un modo semiautomático en el que el analista se vale de herramientas computacionales para facilitar su trabajo.

En resumen, las dos últimas etapas requieren aplicar métodos de decisión. Existe un peligro potencial de falsas alarmas provocadas por desajustes entre el modelo y el sistema real, por lo que algunos autores (Patton y Chen, 1997) han estudiado la **robustez**. El sistema se considera robusto si genera residuos sensibles

a los fallos especificados pero insensibles a la incertidumbre e, incluso, al resto de los fallos. Las técnicas que se utilizan para garantizar esta cualidad pueden ser activas, cuando las mejoras se introducen en la creación de los residuos, o pasivas, cuando se trabaja en la etapa de toma de decisión. Como se indicó, otra opción es que se involucre al supervisor humano, en cuyo caso el uso de técnicas de visualización puede ser muy útil para facilitar su interpretación de los residuos (Díaz y Hollmén, 2002).

Otro objetivo de supervisión parcialmente relacionado con la generación y evaluación de residuos es la **detección de novedades** (*novelty detection*). En este caso, el objetivo no es detectar desviaciones no deseadas sino nuevas estados del proceso o puntos de funcionamiento desconocidos que no se presentaban en los datos disponibles previamente. Se trata de una tarea muy importante, pues un conocimiento más completo del proceso mejora su comprensión y es útil para adaptar el control a la situación actual.

Por otra parte, además de la supervisión, que tiene un carácter inmediato, los métodos basados en datos del proceso permiten un **análisis** del proceso a más largo plazo, que puede ser útil para el diseño del sistema o para refinar el conocimiento sobre el mismo. El propósito en este caso es descubrir dependencias entre variables, reconocer grupos de datos similares (pertenecientes a la misma condición de proceso) u obtener información acerca de su evolución dinámica. Es necesario resaltar de nuevo que muchas veces el conocimiento que se tiene del proceso es, cuanto menos, incompleto. Por ello, en muchas ocasiones, no sólo es útil, sino necesario contar con información alternativa acerca del sistema que permita diseñar una estrategia de control y de supervisión óptima para el mismo.

En la actualidad, debido a la existencia de nuevos requisitos de producción y a la aplicación en nuevos campos, puede ser necesario considerar no sólo las variables del proceso sino también otras relacionadas con el mismo, que potencialmente pueden estar localizadas en lugares remotos, gracias a las telecomunicaciones y, en concreto, a Internet. Su análisis y supervisión también puede ser útil en la toma de decisiones estratégicas orientadas a la calidad, eficiencia, etc.

En el ámbito del análisis y supervisión de procesos industriales, cobra especial importancia el conocimiento del **comportamiento dinámico del sistema**, pues es el aspecto clave para comprenderlos. Por lo tanto, los mayores esfuerzos en este campo deben ir encaminados a entender su evolución, distinguir las diferencias que puedan existir localmente en cada punto de funcionamiento y reconocer la influencia de las variables del proceso en su evolución dinámica.

1.1.3. Técnicas basadas en los datos del proceso

Las técnicas utilizadas tradicionalmente para la supervisión de procesos y la detección de fallos se basan en modelos analíticos, como es el caso de las **ecuaciones de paridad** (Gertler, 1997), la **estimación de parámetros** (Isermann, 2005) o los **observadores** (Patton y Chen, 1997). Estas técnicas utilizan un modelo obtenido a partir de ecuaciones explícitas, que describe el sistema mediante principios físicos. Esto supone una ventaja, pues permite la incorporación de conocimiento físico y, por consiguiente, la definición de modelos constructivos y sólidos. La efectividad de estas técnicas, no obstante, está limitada por la disponibilidad de conocimiento previo, más problemática cuanto más complejo es el sistema en cuestión.

El modelado analítico es una tarea difícil que requiere suficiente precisión. En procesos complejos, la simple aplicación de un modelo lineal frecuentemente no es posible y el modelado es costoso. De igual manera, la construcción de modelos para un sistema a gran escala requiere un gran esfuerzo. Por ello, es conveniente contar con alternativas, entre las que se encuentran las **técnicas basadas en el conocimiento** (Venkatasubramanian *et al.*, 2003a); es decir, técnicas cualitativas basadas en un conocimiento impreciso, incompleto e incluso intuitivo expresado de forma similar al lenguaje natural. No obstante, la obtención de ese conocimiento también constituye una tarea difícil y generalmente estas técnicas sólo se aplican a sistemas relativamente pequeños o al final del proceso.

Por otra parte, cuando la instrumentación permite recopilar gran cantidad de datos de historial de variables del proceso, una situación que es común actualmente, se pueden utilizar ciertas técnicas que permiten transformarlos en conocimiento manejable para poder generar hipótesis verosímiles o encontrar nuevos patrones. Estos datos son a menudo desaprovechados, ya que generalmente no se tienen en cuenta salvo en casos excepcionales y de forma parcial o restringida. El proceso de búsqueda de información, estadísticas y modelos a partir de los datos recibe el nombre de **análisis exploratorio de datos** (Tukey, 1977). Y generalmente, trata de evitar suposiciones acerca de la distribución o estructura de los datos.

En este marco, se han definido técnicas para afrontar las situaciones enumeradas en el apartado anterior. Por ejemplo, algunas técnicas pueden reproducir el comportamiento del sistema, con el objetivo de generar vectores de residuos[1]. Este es el caso de ciertas redes neuronales como los perceptrones multicapa o MLP (Terra y Tinós, 1998) o de las redes de función de base radial o RBF (Yu *et al.*, 1999). Asimismo, otras técnicas (Xu y Wunsch, 2005) permiten realizar un agrupamiento (***clustering***) del proceso en condiciones de proceso, lo que resulta útil

[1]En este caso, se suele trabajar con los mismos en forma vectorial, en la que cada variable representa una dimensión.

en supervisión. En la etapa de toma de decisiones o evaluación, también pueden utilizarse redes RBF o perceptrones multicapa como herramienta de clasificación, con el objetivo de determinar la existencia de novedades o fallos. No obstante, estas propuestas solamente representan una pequeña parte de la multitud de estrategias y algoritmos basados en datos desarrolladas en los últimos años.

1.1.4. Visualización

El objetivo de las técnicas de **exploración visual**, tal y como se introdujo en la sección 1.1.2, es integrar a las personas en el proceso de exploración de los datos. Son técnicas orientadas a la representación de los datos de un modo visual que permita aprovechar la flexibilidad, creatividad y el conocimiento del dominio que poseen los seres humanos (Keim, 2001).

La tarea principal de estos gráficos es, pues, presentar la gran cantidad de datos al analista de un modo que agilice su razonamiento y facilite el reconocimiento de estructuras, patrones, novedades, anomalías, tendencias o correlaciones. También pretenden facilitar la comparación entre valores y la búsqueda de datos, la comparación con modelos y el descubrimiento de errores o detalles inesperados. El razonamiento con conceptos visuales ha demostrado ser muy apropiado en este aspecto por su utilidad a la hora de percibir directamente patrones que, de otro modo, solamente podrían descubrirse mediante arduos procesos cognitivos. Especialmente, cuando se conoce poco acerca del proceso, los objetivos o umbrales no están claros o los resultados son complejos.

Se han propuesto **técnicas de visualización de conjuntos de datos multidimensionales**, que utilizan diversos métodos (Keim, 2001) para visualizar simultáneamente más de tres variables. Estas técnicas se basan en:

- Transformaciones geométricas, como las curvas de Andrews o la técnica de coordenadas paralelas (Keim, 2002), donde un punto de un espacio n-dimensional se representa por medio de una polilínea.
- Gráficas en las que los valores se codifican mediante colores, conocidas como *dense pixel displays* (Keim, 2002), y en las que las componentes del vector se agrupan en un eje.
- Múltiples gráficas simultáneas (Keim, 2002), asociadas por posición o de forma explícita, que permiten una inspección visual eficaz, ya que es posible interpretarlas de forma similar y compararlas entre sí (Vesanto, 2002). Un ejemplo de ellas son las matrices de gráficas de dispersión, unas tablas donde se muestran, dos a dos, las gráficas de dispersión[2] de todas las dimensiones.

[2]Colecciones de puntos cuyas coordenadas son los valores de las variables.

- Visualizaciones icónicas que hacen uso de marcadores visuales parametrizables. En este caso son las formas las encargadas de incluir más información.

Las capacidades de estas representaciones también se pueden ampliar permitiendo interacciones y distorsiones sobre ellas (Keim, 2001). No obstante, estas técnicas visuales multidimensionales presentan un problema evidente para el caso que nos ocupa: no reducen por sí mismas ni la dimensión ni la cantidad de datos (Kaski, 1997). Por ello, no siempre son suficientes para ayudar al analista a razonar con conjuntos de datos de alta dimensión y gran tamaño, cuya representación directa continuaría siendo prácticamente incomprensible. En cualquier caso, eso no implica que estos métodos no sean útiles como herramientas auxiliares. Es más, puesto que son completamente separables de las transformaciones que se realicen sobre los datos, pueden proporcionar diferentes vistas de los resultados.

Debido a la incapacidad de las técnicas visuales para resolver el problema, es necesario utilizar técnicas de **reducción de la dimensión** que proyecten de forma efectiva los datos en una dimensión más baja. No obstante, la mayor parte de ellas no han sido específicamente diseñadas para la visualización. De igual manera, es necesario simplificar el conjunto de datos realizando una compresión de los mismos. Para ello, se hace uso de algoritmos de **cuantización de vectores** (VQ). Estos algoritmos proporcionan una representación eficiente y compacta de los datos, proporcionando un conjunto de vectores prototipo, que minimiza alguna medida de distorsión. Esta simplificación del conjunto de datos a información estrictamente relevante hace posible la abstracción que se requiere para su tratamiento.

1.1.5. Extracción del conocimiento

El enfoque de la transformación de los datos en información visualizable abre un amplio horizonte más allá de la simple monitorización mediante gráficas o espectros de señales.

Existen multitud de algoritmos que permiten reducir la dimensión de los datos o comprimirlos. En muchos de los casos, el objetivo de los mismos no es explícitamente la visualización. De entre las herramientas disponibles cabe destacar el **mapa auto-organizado** (Kohonen, 2001) o **SOM**—*self-organizing map*—, que permite llevar a cabo ambos procesos al mismo tiempo. La proyección que realiza de los datos de entrada en una retícula discreta y ordenada de baja dimensión le hace una herramienta muy apropiada para objetivos de visualización. De su potencial dan fe gran variedad de estudios, aplicaciones y extensiones (Oja *et al.*, 2003).

Su capacidad para preservar la topología de los datos en su representación simplificada ha sido clave para sus aplicaciones en el área de los procesos industriales

(Kohonen *et al.*, 1996). Estas aplicaciones se centran en características estáticas de procesos. No obstante, su capacidad para trabajar con comportamiento dinámico, tanto desde un punto de vista de modelado (Principe *et al.*, 1998) como de visualización (Díaz *et al.*, 2008), ha sido puesta de relieve.

Esto abre un interesante campo de estudio centrado en el uso de técnicas basadas en datos, concretamente el SOM, para modelar y/o visualizar la dinámica de los procesos, cuyo conocimiento es de suma importancia para el análisis y supervisión de procesos.

1.2. Objetivos de la tesis

La hipótesis de partida de esta tesis es doble. En primer lugar, se parte del supuesto de que el uso del algoritmo SOM para la definición de mapas de visualización a partir de datos de historial puede resultar útil para la comprensión de la dinámica de procesos industriales mono- y multivariables. En segundo lugar, se considera que el algoritmo SOM puede verse beneficiado del uso de modificaciones que introducen contexto temporal para objetivos de modelado dinámico basado en la agrupación de modelos locales.

Por eso, de forma general, y a la vista de todo lo comentado en este capítulo, se puede expresar como objetivo marco de esta tesis la extracción de conocimiento relativo a la dinámica de procesos industriales a partir de los datos de historial, con aplicaciones en el análisis y la supervisión de los mismos. Para ello, se utilizan mapas auto-organizados.

De forma particularizada, se pueden expresar los siguientes objetivos:

1. Definir mapas de visualización de apoyo al analista, que le permitan extraer conocimiento acerca de las características dinámicas más habituales en el análisis y supervisión tanto de sistemas monovariables como de sistemas multivariables. Estos mapas deben ser comparables entre sí y mostrar, de un modo sencillo y coherente, información acerca de la dinámica del proceso.

2. Evaluar la aplicabilidad, utilidad y rendimiento de las principales modificaciones para procesamiento temporal del mapa auto-organizado en el modelado de la dinámica mediante el método de múltiples modelos locales. Evaluar también los métodos de ajuste de los modelos locales.

1.3. Estructura de la tesis

La tesis se estructura de la siguiente manera:

- En el Capítulo 1, se describe de forma general el problema de partida en que se basa esta tesis y los objetivos que pretende alcanzar.
- En el Capítulo 2, se presenta el estado actual de la investigación en los ámbitos de los que se ocupa esta tesis, con el objetivo de encuadrar este trabajo en el contexto adecuado.
- En el Capítulo 3, se define en detalle el problema de extracción de conocimiento acerca de la dinámica de los procesos industriales, descompuesto en dos subproblemas, uno relacionado con la selección de métodos apropiados para el modelado y el otro con la necesidad de contar con mapas de visualización de apoyo a tareas comunes en ingeniería de control. También se explican los métodos propuestos para afrontar el problema.
- En el Capítulo 4, se definen el entorno de pruebas, las condiciones de las mismas y los experimentos propuestos para evaluar la validez de la metodología propuesta en el capítulo anterior.
- En el Capítulo 5, se analizan y discuten los resultados experimentales obtenidos.
- Finalmente, en el capítulo 6, se presentan las conclusiones extraídas de los resultados experimentales, se enumeran las aportaciones y se sugieren las líneas futuras de investigación que se derivan de esta tesis.

Capítulo 2

Antecedentes

2.1. Introducción

En el capítulo anterior se introdujeron las técnicas basadas en datos del proceso y el *visual data mining* como recursos útiles para extraer conocimiento acerca de los procesos industriales para su análisis y supervisión. También se sugirió el uso del SOM como método versátil para el tratamiento y visualización del comportamiento dinámico.

A lo largo de este capítulo, con el objetivo de encuadrar el trabajo dentro de las investigaciones en este ámbito, se analizan las técnicas orientadas a la reducción de la dimensión, especialmente las que realizan una proyección de los datos en un espacio de baja dimensión, los algoritmos de cuantización de vectores y, más en detalle, el SOM, con especial énfasis en su uso para visualización y modelado dinámico.

2.2. Reducción de la dimensión

Como se afirmó en el capítulo anterior, la reducción de la dimensión es un proceso crucial para la visualización de conjuntos de datos complejos. No obstante, no solamente es útil para obtener información visualizable con el objetivo de extraer conocimiento. En algunos casos, por ejemplo, su uso es indispensable para que los datos sean manejables por otros algoritmos, ya que el problema conocido como **maldición de la dimensionalidad** implica que, en muchos algoritmos, la complejidad y el número necesario de datos de entrenamiento dependa directamente de la dimensión de entrada.

Por otra parte, el concepto de reducción de la dimensión se puede enfocar desde

diversos puntos de vista. En el ámbito del reconocimiento de patrones se suele considerar un proceso de **extracción de características**, ya que transforma los datos de entrada a una representación más útil para la comprensión y utilización posterior. En términos estadísticos, se puede entender como un proceso de inferencia de las variables latentes (Carreira-Perpiñán, 1996). Y con un criterio geométrico, se trata de una representación en un sistema de coordenadas diferente.

La reducción de la dimensión es posible porque los conjuntos de datos de alta dimensionalidad suelen ser más simples que lo que su dimensionalidad indica, ya que contienen información redundante, correlaciones y/o variables con una variación menor que el ruido. Por eso, el número de variables independientes que pueden describirlo satisfactoriamente (**dimensión intrínseca**) es menor que su dimensión nominal (Carreira-Perpiñán, 2001). Esta justificación se puede formalizar utilizando un enfoque topológico:

Sean los datos de entrada vectores D-dimensionales, se supone que estos vectores se encuentran, al menos aproximadamente, en una variedad contenida en el espacio D-dimensional que tiene una dimensión intrínseca $n < D$.

Una **variedad** n-dimensional se define como un espacio topológico de Hausdorff que verifica el segundo axioma de numerabilidad y que es localmente euclídeo de dimensión n. Intuitivamente, se puede describir como un espacio topológico que localmente se asemeja a $\mathbb{R}^n$. Es decir, que independientemente del espacio del que es subconjunto, solamente se necesita una tupla de n valores para especificar un punto en su vecindad.

El objetivo de la reducción de la dimensión es, por tanto, encontrar una representación en la variedad (un sistema de coordenadas) que permita una proyección de los datos a una forma más compacta y que, a su vez, permita una reconstrucción no singular y derivable con errores pequeños (Carreira-Perpiñán, 2001).

El cálculo de la dimensión intrínseca es un problema central de la reducción de la dimensión, pues su conocimiento sería útil para afinar el entrenamiento. En cualquier caso, esta dimensión puede depender de los criterios que se consideren, así que podría requerirse información a priori. No obstante, para los casos de visualización, el objetivo no es reducir el conjunto de datos a su dimensión intrínseca, sino a una visualizable: 2 ó 3. Esta transformación debe preservar la información en la medida de lo posible. Asimismo, es interesante que la proyección conserve la topología, es decir, que los puntos adyacentes en el espacio de entrada lo sean en el espacio de salida. Un ejemplo esquemático de reducción de la dimensión se muestra en la Fig. 2.1. A continuación, se describen brevemente algunos de los métodos más utilizados para reducir la dimensión de los datos.

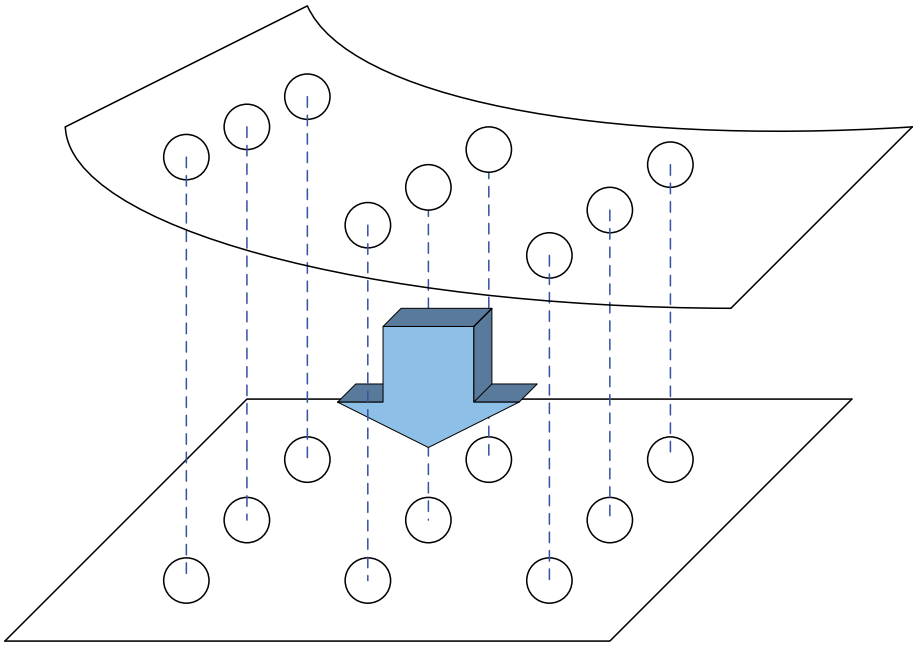


Figura 2.1. Reducción de la dimensión.

Selección de características. El objetivo es encontrar el subconjunto de variables del conjunto de datos de entrada que minimice un determinado criterio de error (Alpaydin, 2004). Este criterio se puede basar en conocimiento previo, en criterios estadísticos o en teoría de la información.

Análisis de componentes principales (PCA). También denominado **transformación de Karhunen-Loéve**, es un método no supervisado de reducción de la dimensión que preserva las direcciones que conservan en mayor medida la varianza de los datos (Alpaydin, 2004). Para ello, realiza una proyección lineal de los vectores $\mathbf{x}_i$ del espacio de entrada a un espacio de menor dimensión que tiene como base la matriz $\mathbf{A}$ de autovectores correspondientes a los mayores autovalores de la matriz de covarianza $\mathbf{\Sigma}$

$$\mathbf{y}_i = \mathbf{A}\mathbf{x}_i. \tag{2.1}$$

Es decir, realiza una descomposición ortogonal de la matriz de covarianza en las direcciones con mayor variación en los datos. Estos autovectores reciben el nombre de **componentes principales**.

Este método es probablemente el más utilizado y produce imágenes fáciles de interpretar. En multitud de ocasiones, se utiliza como paso previo antes de otros algoritmos, y ha sido ampliamente aplicado en todos los ámbitos, incluyendo la supervisión de procesos. Es la mejor proyección lineal en cuanto a la minimización del error cuadrático medio en los datos reconstruidos y la maximización de la información mutua entre los vectores de entrada y los proyectados, suponiendo que los vectores de entrada tengan una distribución normal (Carreira-Perpiñán, 2001). Por otra parte, su principal defecto es que no es útil cuando los datos son no lineales, puesto que solamente utiliza información estadística de segundo orden. No obstante, se han propuesto diversas generalizaciones que solucionan este problema.

Generalizaciones no lineales del PCA. Existen varios algoritmos que permiten aplicar el análisis de componentes principales al caso no lineal.

- Los **perceptrones multicapa** (*MLP*) permiten realizar una proyección equivalente a un análisis de componentes principales cuando se utilizan de forma **autoasociativa** (Kramer, 1991). Para ello, se define una red cuya capa de entrada y de salida tienen un número de neuronas igual a la dimensión de los datos de entrada, d. Después, se introduce una capa oculta lineal con m neuronas, tal que $m < d$, y la red se entrena con el algoritmo tradicional de *backpropagation*, pero presentando cada patrón como la entrada y salida deseadas. De esa manera, la capa oculta actúa como cuello de botella y elimina información redundante. Se puede conseguir una versión no lineal introduciendo capas ocultas con funciones de activación no lineales antes y después de la capa oculta lineal. No obstante, este tipo de redes presentan ciertos problemas como un entrenamiento lento o tendencia a acabar en mínimos locales.

- Las **curvas principales** (Hastie y Stuetzle, 1989) son una generalización natural al caso no lineal de las componentes principales, que se estima de forma no paramétrica, a partir de los datos. Cada punto que forma una curva principal es el promedio de los datos para los cuales dicho punto es el más cercano en la curva, es decir, la curva es autoconsistente. Por tanto, las curvas principales son aquellas que pasan por el centro de un conjunto de datos p-dimensional, proporcionando un resumen no lineal de ese conjunto. El concepto se puede extender a dimensiones o variedades más altas y entonces reciben el nombre de *superficies o variedades principales*. Se ha demostrado que las curvas principales guardan gran similitud con el SOM (Mulier y Cherkassky, 1995), principal objeto de estudio de esta tesis.

- Otra opción es recurrir al método de kernel para aplicar el PCA en un espacio de características de alta dimensión, relacionado con el espacio de entrada por medio de una transformación no lineal. Para ello fue propuesto el algoritmo **Kernel PCA** (Schölkopf *et al.*, 1998), cuya principal ventaja es que evita la necesidad de realizar una optimización no lineal, pues el problema de autovalores se expresa en forma de productos escalares y los kernels permiten realizar el cálculo sin necesidad de realizar explícitamente la transformación al espacio de alta dimensión. El método evita mínimos locales, pero a cambio es muy dependiente del *kernel* utilizado. En el caso de que el conjunto de datos sea muy grande, el algoritmo requiere muchos cómputos.

Projection Pursuit. El objetivo de este conjunto de métodos es encontrar la proyección que optimiza un **índice**, que generalmente mide alguna desviación respecto a la distribución normal (Carreira-Perpiñán, 2001). Diversos métodos, entre los que se encuentra el PCA, son casos especiales del mismo, pero los índices no se limitan a información estadística de segundo orden. Estas técnicas suelen requerir un cómputo intensivo y tienden a preservar *outliers*, debido a su apariencia no gaussiana, que pueden enturbiar las estructuras interesantes.

Análisis de Componentes Independientes (ICA). Es un método que busca proyecciones lineales que sean lo más estadísticamente independientes posibles (Hyvärinen *et al.*, 2001). La condición de independencia es más fuerte que la de correlación e implica momentos de alto orden. Las componentes independientes deben ser no gaussianas. ICA puede encontrar una recodificación interesante de las variables que revele ciertos fenómenos.

Escalado multidimensional (MDS). El objetivo de esta familia de algoritmos (Webb, 1999) es conseguir, mediante la minimización de una función de coste, que las distancias en el espacio de baja dimensión de salida representen las disimilitudes del espacio de entrada. De este modo, la configuración geométrica de puntos (vectores de datos) puede reflejar su estructura oculta y facilitar su comprensión. En algunos de estos algoritmos no se requiere que las disimilitudes sean distancias en un sentido matemático estricto.

Una de las variantes es el **MDS lineal** o **escalado clásico**, que se resuelve como un problema de autovalores y está relacionado con el PCA. Otro algoritmo, conocido como **Escalado multidimensional métrico**, utiliza la siguiente función de coste:

$$E = \sum_i \sum_{j \neq i} (X_{ij} - Y_{ij})^2 \tag{2.2}$$

donde $\mathbf{X} = (X_{ij})$ e $\mathbf{Y} = (Y_{ij})$ son las matrices de distancias mutuas de los puntos de entrada $\mathbf{x}_k \in \mathbb{R}^n$ y los de salida $\mathbf{y}_k \in \mathbb{R}^p$, respectivamente. Este algoritmo se centra en preservar largas distancias, lo que resulta en una buena conservación de la estructura global.

También se ha definido un algoritmo de **Escalado multidimensional no métrico**, motivado por la necesidad de tratar datos ordinales. En este caso, modificada mediante una función monótona creciente que actúa sobre las distancias originales, la proyección preserva el orden de distancias entre vectores pero no su valor absoluto.

Todos estos métodos son intuitivos pero presentan algunas desventajas como su alta carga computacional, tendencia a converger a mínimos locales o la necesidad de que los datos estén distribuidos de forma uniforme sobre su variedad.

Proyección de Sammon. Este método de escalado multidimensional (Sammon, Jr., 1969) utiliza una función de coste normalizada con la distancia en el espacio de entrada

$$E = \frac{1}{\sum_i \sum_{j<i} X_{ij}} \sum_i \sum_{j<i} \frac{(X_{ij} - Y_{ij})^2}{X_{ij}}, \tag{2.3}$$

que da más importancia a las distancias pequeñas. Se basa en la idea de que, puesto que la proximidad normalmente refleja similitud o pertenencia a un grupo, es razonable dar más énfasis a la información acerca de la vecindad para así preservar la topología local.

Isomap. El uso de distancia euclídea para el escalado multidimensional ignora la existencia de variedades, ya que dos puntos cercanos en el espacio podrían estar lejos dentro de una variedad de baja dimensión definida por los datos. El camino mínimo, contenido en la variedad, que une dos puntos de la misma, recibe el nombre de **distancia geodésica** y puede ser más útil que la distancia (Carreira-Perpiñán, 2001).

El Isomap (Tenenbaum *et al.*, 2000) es una variante del escalado multidimensional lineal que utiliza distancias geodésicas, que se aproximan por medio de caminos mínimos de grafos en los que las distancias euclídeas son los pesos de las aristas entre vecinos. El resultado, que busca una optimización global, se puede obtener aplicando MDS lineal a la matriz de distancias mínimas. Este método también puede considerarse un método de aprendizaje de la variedad (*manifold learning*), pues proporciona una estimación de la dimensión de la misma por medio del número de autovalores distintos de cero encontrados por el algoritmo. Bajo ciertas condiciones, garantiza que el Isomap converge a una parametrización correcta de la

variedad. No obstante, en la práctica, la escasez de los datos de entrenamiento y la complejidad del algoritmo que, además, es muy dependiente de la vecindad, lo hacen difícil.

Locally Linear Embedding (LLE). El algoritmo se basa en la suposición de que es posible realizar una aproximación localmente lineal de la variedad de los datos (Roweis y Saul, 2000), ya que un vector y sus vecinos se encuentran aproximadamente en un subespacio lineal de la variedad topológica. LLE preserva la topología mediante relaciones lineales en la vecindad en lugar de hacerlo por medio de distancias entre puntos, como en los algoritmos MDS previamente presentados. Para ello, cada dato se reconstruye en el espacio de salida como una combinación lineal de sus K vecinos más cercanos, por medio de mínimos cuadrados. En primer lugar, se eligen los pesos w_{ij} que minimicen el error

$$E(\mathbf{W}) = \sum_i \left\| \mathbf{x}_i - \sum_{j \epsilon \mathcal{N}(\mathbf{x}_i)} w_{ij} \mathbf{x}_j \right\|^2 , \tag{2.4}$$

donde $\mathcal{N}(\mathbf{x}_i)$ es el conjunto de vecinos de $\mathbf{x}_i$ y $\sum_j w_{ij} = 1$. Con estos pesos y tras fijar la dimensión de salida, se calculan los puntos $\mathbf{y}_i$ de salida resolviendo otro problema de minimización

$$E(\mathbf{Y}) = \sum_i \left\| \mathbf{y}_i - \sum_{j \epsilon \mathcal{N}(\mathbf{y}_i)} w^*_{ij} \mathbf{y}_j \right\|^2 . \tag{2.5}$$

El problema se puede resolver mediante un cálculo de autovalores. Se introduce la restricción de que $\mathbf{Y}$ tenga media cero y covarianza unitaria.

Eigenmaps. El método **Laplacian Eigenmap** (Belkin y Niyogi, 2003) es similar al algoritmo LLE, pues también considera solamente los K vecinos más cercanos. De hecho, se ha demostrado su equivalencia en ciertas situaciones. Sin embargo, la justificación geométrica de este algoritmo se basa en la analogía entre el laplaciano de un grafo y el operador Laplace-Beltrami de una variedad. El conjunto de puntos se obtiene resolviendo

$$\mathbf{L}\mathbf{y} = \lambda \mathbf{D}\mathbf{y}, \tag{2.6}$$

donde $\mathbf{L} = \mathbf{D} - \mathbf{W}$ es la matriz laplaciana, $\mathbf{D}$ es una matriz diagonal con elementos, W_{ij} son los pesos de las aristas y $D_{ii} = \sum_j W_{ij}$. La solución son los autovectores correspondientes a los autovalores más pequeños distintos de cero.

Por otra parte, el algoritmo **Hessian Locally Linear Embedding** (Donoho y Grimes, 2003) es una modificación del LLE cuyo marco teórico deriva del *laplacian eigenmap*, pero que utiliza un estimador hessiano en lugar del laplaciano del grafo.

Stochastic Neighbor Embedding (SNE). Este algoritmo (Hinton y Roweis, 2002) no trata de preservar distancias entre puntos, como los métodos de escalado multidimensional, sino las probabilidades de que esos puntos sean vecinos. Para ello, se centran funciones gaussianas en cada punto i del espacio de entrada y se determina la distribución de probabilidad sobre todos los potenciales vecinos. El objetivo es encontrar una configuración de puntos que aproxime esas distribuciones en el espacio de salida. Para ello, se minimiza la suma de las divergencias de Kullback-Leibler[1] entre las distribuciones de entrada y de salida. La función de coste de este algoritmo no sólo mantiene cerca la imagen proyectada de objetos cercanos, sino que también aleja la imagen proyectada de objetos lejanos en el conjunto de datos original, al contrario de la mayor parte de métodos, en los que puntos muy separados pueden ser vecinos en el espacio de salida.

Mapa auto-organizado (SOM). El mapa auto-organizado o *self-organizing map* (Kohonen, 1990) es una red neuronal no supervisada que realiza una proyección de un espacio de alta dimensión en una retícula discreta de baja dimensión visualizable (generalmente bidimensional), que captura sus características y preserva su topología. Asimismo, reduce el conjunto de datos, es decir, lleva a cabo una cuantización de vectores (ver apartado 2.3). Una explicación más completa de las características, capacidades y aplicaciones de este algoritmo se introduce en el apartado 2.4 y siguientes.

Análisis de componentes curvilíneas (CCA). Este algoritmo es otra variante de MDS (Demartines y Hérault, 1997), pues minimiza una función de coste basada en distancias entre puntos, tanto en el espacio de entrada como en el de salida. Esa función de coste se define como

$$E = \frac{1}{2} \sum_i \sum_{j \neq i} (X_{ij} - Y_{ij})^2 F(Y_{ij}, \lambda_Y), \tag{2.8}$$

[1]Medida de diferencia no conmutativa y no simétrica entre dos distribuciones de probabilidad que se define como

$$D_{KL} = \int_{-\infty}^{\infty} p(x) \log \frac{p(x)}{q(x)} dx. \tag{2.7}$$

donde F es generalmente una función acotada y monótona decreciente, como por ejemplo

$$F(Y_{ij}, \lambda_Y) = \begin{cases} 1 & si \quad Y_{ij} \leq \lambda_Y \\ 0 & si \quad Y_{ij} > \lambda_Y \end{cases} \tag{2.9}$$

El parámetro λ_Y decrece a lo largo del entrenamiento y permite controlar la escala a la que trabaja el algoritmo.

El CCA es un algoritmo de menor complejidad computacional que otros métodos MDS, pues en cada etapa fija un vector $\mathbf{y}_i$ y adapta el resto, $\mathbf{y}_j$. De este modo, en cada ciclo de adaptación no es necesario calcular $N(N-1)/2$ distancias sino solamente las distancias desde el dato i a los demás. Con una $F(Y_{ij}, \lambda_y)$ discreta, se obtiene la siguiente regla de actualización de los vectores $\mathbf{y}_j$

$$\triangle \mathbf{y}_j = \alpha(t) F(Y_{ij}, \lambda_y)(X_{ij} - Y_{ij}) \frac{\mathbf{y}_j - \mathbf{y}_i}{\mathbf{Y}_{ij}} \qquad \forall j \neq i \tag{2.10}$$

El CCA entrenado permite la interpolación y extrapolación de nuevos datos. En este caso, aunque se utiliza la misma función de coste, solamente se calcula con respecto al punto $\mathbf{y}_0$ correspondiente al dato $\mathbf{x}_0$. Esto da lugar a la adaptación de $\mathbf{y}_0$ mediante un descenso de gradiente estocástico, mientras el resto de datos se mantienen fijos.

Este algoritmo se ha propuesto como una mejora del SOM en la que la salida no es una malla fija sino un espacio continuo que puede tomar la forma de la variedad. Al igual que en el SOM, permite realizar una proyección no lineal y una cuantización de vectores y posee buenas características para la visualización (Venna y Kaski, 2006).

Se han propuesto diversas extensiones del algoritmo como el **análisis de distancias curvilíneas** o CDA, que utiliza distancias geodésicas en lugar de euclídeas, o el **escalado multidimensional local** (Venna y Kaski, 2006), que permite parametrizar la fiabilidad y continuidad de la visualización.

Generative Topographic Mapping (GTM). En este caso, se parte de la premisa de que los datos observados de alta dimensión son generados por un proceso de baja dimensión subyacente (Carreira-Perpiñán, 2001). Es decir, que la distribución $p(\mathbf{t})$ en el espacio observado $\mathcal{T}$, de D dimensiones, se debe a un pequeño número $L < D$ de variables latentes. Las causas de esta alta dimensionalidad serían múltiples, como la variación estocástica y el proceso de medida.

El GTM (Bishop *et al.*, 1998) es un modelo no lineal de variables latentes que se puede considerar una alternativa con base estadística al SOM. El objetivo del modelo es obtener una representación de la distribución $p(\mathbf{t})$ del espacio

D-dimensional en función de las variables latentes $\mathbf{x} = (x_1, \ldots, x_L)$. La transformación que proyecta los puntos del espacio latente en los puntos correspondientes del espacio de datos (concretamente en una variedad L-dimensional contenida en ese espacio) depende de la matriz de parámetros $\mathbf{W}$, de tal manera que la distribución $p(\mathbf{x})$ en el espacio de variables latentes induce otra distribución $p(\mathbf{y}|\mathbf{W})$ en el espacio de datos. Puesto que $p(\mathbf{y}|\mathbf{W})$ está confinada en la variedad de baja dimensión, se introduce un modelo de ruido para obtener el vector $\mathbf{t}$, concretamente una distribución gaussiana simétrica centrada en $\mathbf{y}(\mathbf{x}; \mathbf{W})$ y con varianza β^{-1}. Para determinar los parámetros $\mathbf{W}$ y β se pueden utilizar los algoritmos de *Expectation-Maximization* y de Robins-Monro. El algoritmo garantiza la convergencia a un máximo local de la función objetivo. La preservación de la vecindad del algoritmo procede directamente del uso de la función continua $\mathbf{y}(\mathbf{x}; \mathbf{W})$ y el coste computacional del algoritmo es similar al del SOM.

Para objetivos de visualización se utiliza una dimensión $L = 2$. Puesto que el modelo GTM se define como una transformación del espacio latente al de datos, es necesario invertir la proyección mediante el teorema de Bayes, lo que da lugar a la distribución a posteriori en el espacio latente, de la cual suele representarse su media.

2.3. Cuantización de vectores

El objetivo de la cuantización de vectores (VQ) es aproximar los vectores de entrada utilizando un número pequeño de vectores prototipo (Gray y Neuhoff, 1998; Cherkassky y Mulier, 2007). Por una parte, es útil como herramienta de compresión de datos en un amplio rango de aplicaciones que toleran cierta distorsión. Por otra parte, la reducción del conjunto de datos también permite una mejor comprensión del mismo para el propósito del que se ocupa esta tesis.

Una cuantización de vectores Q de dimensión k y tamaño N es una transformación del espacio euclídeo k-dimensional $\mathbb{R}^k$ en un conjunto finito de **vectores de reproducción** o **vectores prototipo** (ver Fig. 2.2), que recibe el nombre de ***codebook*** $Y = (\mathbf{y}_1, \ldots, \mathbf{y}_N)$. Generalmente se utiliza un N mucho menor que el número de datos del conjunto de entrada. La cuantización es óptima si minimiza una función de distorsión, que representa la penalización o coste asociado a la sustitución del vector de entrada $\mathbf{x}$ por su vector prototipo correspondiente, $\mathbf{y}_i$. Generalmente se trabaja con el error cuadrático medio o norma L_2.

Los vectores prototipo particionan $\mathbb{R}^k$ en regiones, que se definen como

$$R_i = \{\mathbf{x} \epsilon \mathbb{R}^k : Q(\mathbf{x}) = \mathbf{y}_i\}. \tag{2.11}$$

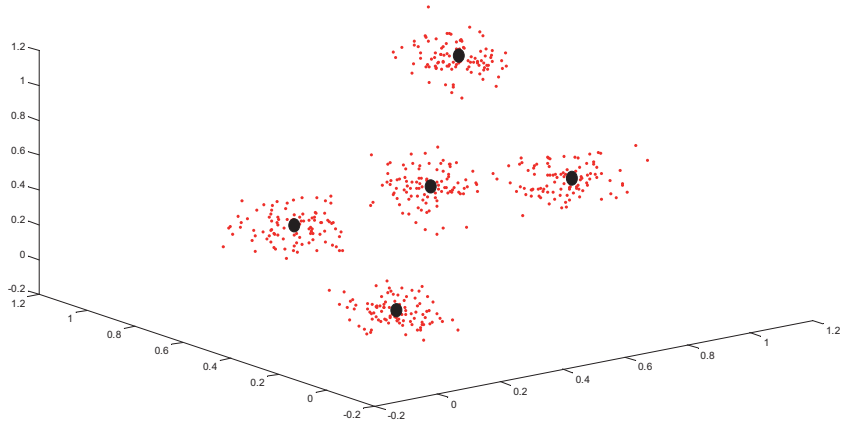


Figura 2.2. Cuantización de vectores.

Un vector reproduce de forma óptima una región R_i cuando es el centroide

$$\mathbf{y}_i = \frac{\sum_{j \epsilon R_i} \mathbf{x}_j}{|R_i|} \tag{2.12}$$

y la partición es de **Voronoi**, es decir, que para un conjunto fijo de vectores de reproducción, la partición óptima se construye de la siguiente manera:

$$Q(\mathbf{x}) = \mathbf{y}_i \Leftrightarrow \|\mathbf{x} - \mathbf{y}_i\| \leq \|\mathbf{x} - \mathbf{y}_r\|, \forall r \neq i. \tag{2.13}$$

Estos requisitos reciben el nombre de **condiciones de Lloyd-Max** (Cherkassky y Mulier, 2007).

Existe gran similitud entre la idea de cuantización de vectores y el concepto de ***clustering*** (Xu y Wunsch, 2005). Lo que les diferencia es el objetivo, pues los métodos de *clustering* buscan agrupaciones interesantes de los datos de entrada. Es decir, mientras que el objetivo de la cuantización es la representación de los datos (tras el entrenamiento, los datos se sustituyen por los vectores prototipo más cercanos), el del *clustering* es su interpretación. También existen similitudes con el *mixture modeling*[2], cuando se utilizan métodos con *annealing* (Heskes, 2001).

[2]Modelado de una distribución de probabilidad mediante la combinación de otras distribuciones.

El algoritmo de Linde *et al.* (1980), conocido como **LBG**, es el más utilizado para este propósito. Este algoritmo minimiza la norma L_2 y, aunque no asegura un mínimo global, sí garantiza alcanzar mínimos locales, pues cada vector se mueve en la dirección del centro de gravedad. La calidad de la solución depende en gran medida de la inicialización de los vectores de reproducción, que generalmente se realiza con valores aleatorios del mismo rango que los datos de entrada o con muestras aleatorias de ese conjunto de datos. Por ello, para encontrar una buena solución, el algoritmo se itera varias veces, con diferente inicialización, o se utilizan técnicas de *annealing*. Dado el conjunto de datos de entrada e inicializados los vectores prototipo, el algoritmo LBG sigue los siguientes pasos:

1. Se obtienen los vectores prototipo para todos los datos de entrada.
2. Se calculan los centroides a partir de los datos de entrada contenidos en cada región. Los antiguos vectores prototipo son reemplazados por estos centroides.
3. Se reiteran los pasos anteriores hasta que el error alcance un determinado umbral.

El algoritmo LBG es similar al **K-Means** (MacQueen, 1967), que es el método de *clustering* basado en error cuadrático más conocido.

También se pueden utilizar criterios de teoría de la información para conseguir cuantización de vectores. El algoritmo **ITVQ** presentado en (Rao *et al.*, 2007) minimiza la divergencia de Cauchy-Schwartz[3] entre las distribuciones de probabilidad de los datos de entrada y del *codebook*, que es estimada de forma no paramétrica mediante kernels gaussianos. Al utilizar información estadística de alto orden, ajusta mejor los vectores prototipo a las propiedades estructurales del conjunto de entrada, utilizando más vectores para representar regiones complejas. El algoritmo utiliza una actualización de punto fijo, derivada directamente a partir de la minimización de la divergencia, y permite una simple interpretación física en base a potenciales y fuerzas, si se consideran los vectores como partículas de información.

Los dos algoritmos presentados previamente necesitan tener acceso a todo el conjunto de datos de entrada durante el entrenamiento, es decir, utilizan un entrenamiento *batch* o por lotes. En ciertos casos, pueden ser más apropiados algoritmos VQ adaptativos o incrementales en los que cada vector se compara con los vectores de reproducción, se elige el que mejor le representa y el vector de reproducción se modifica para reflejar la inclusión del nuevo dato en su partición

$$\mathbf{y}_{Q(\mathbf{x})}(t+1) = \mathbf{y}_{Q(\mathbf{x})}(t) + \eta[\mathbf{x}(t) - \mathbf{y}_{Q(\mathbf{x})}(t)]. \tag{2.14}$$

[3]Definición de divergencia basada en la desigualdad de Cauchy-Schwartz.

Esta expresión se conoce como regla competitiva *winner-take-all* en el ámbito de redes neuronales. Se han propuesto diversas redes neuronales para llevar a cabo cuantización que permiten un entrenamiento secuencial mediante aprendizaje competitivo.

El mapa auto-organizado también produce una cuantización de vectores, pero sujeta a más restricciones. De hecho, la ecuación (2.14) es un caso especial del SOM sin interacciones laterales entre neuronas. La introducción del concepto de vecindad (regla *soft-max*) modifica el objetivo para preservar la topología, introduciendo una regularización cuyo objetivo es que las distancias entre neuronas vecinas sean pequeñas (Heskes, 2001).

2.4. Mapas auto-organizados

2.4.1. Descripción

El **mapa auto-organizado** (Kohonen, 1982, 2001) es una red neuronal no supervisada que realiza una proyección de un espacio de dimensión arbitraria en una retícula discreta de baja dimensión visualizable, por medio de un proceso de aprendizaje competitivo y cooperativo. La transformación se realiza de forma adaptativa y de un modo topológicamente ordenado (Haykin, 1994).

Los mapas auto-organizados fueron inspirados por una característica distintiva del cerebro humano, ya que lo las señales percibidas por sentidos como el tacto, la vista o el oído se almacenan en el córtex de un modo topológicamente ordenado. No obstante, el modelo de Kohonen no trata de simular detalles neurobiológicos, sino capturar las características esenciales de los datos mediante un algoritmo tratable computacionalmente (Kohonen, 2001).

La red está compuesta por una capa de entrada y una capa de salida, generalmente dispuesta en forma de malla bidimensional (aunque también podrían utilizarse capas de salida de una o tres dimensiones). Cada neurona de la malla está conectada a todos los nodos origen de la capa de entrada y se describe por medio de su **vector de pesos** $\mathbf{m}_i$, que es un vector prototipo $d-$dimensional del espacio de entrada (Hollmén, 1996), y su **posición** $\mathbf{g}_i$, en la malla de baja dimensión del espacio de salida. Los vectores prototipo cuantizan la entrada, dividiendo el espacio en una colección finita de regiones de Voronoi, mientras que las coordenadas de las neuronas preservan la topología de la distribución de entrada, es decir, los vectores cercanos en el espacio de entrada mantienen la vecindad en el espacio de salida. La estructura básica de la red se representa en la Fig. 2.3.

Las neuronas se relacionan con sus adyacentes por medio de una **función de vecindad** h_{ij}, que es el método utilizado en esta red para implementar la

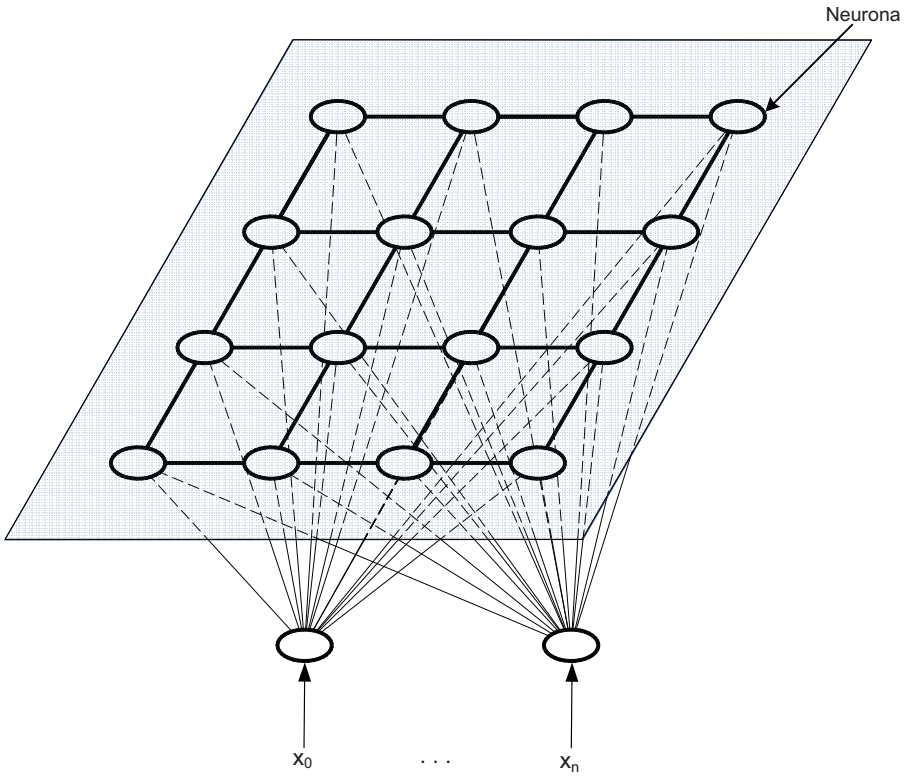


Figura 2.3. Esquema del mapa auto-organizado.

interacción lateral entre neuronas.

Sea $\mathbf{x} = [x_1, x_2, \ldots, x_d]^T \in \Re^d$ la señal de entrada, el algoritmo SOM lleva a cabo su proyección en dos etapas. En primer lugar, se elige la neurona $\mathbf{m}_c$ que mejor representa el vector de entrada, que recibe el nombre de **neurona ganadora** o *best matching unit* (**BMU**). La **selección** se realiza por medio de un proceso competitivo donde se elige a aquella cuya distancia al vector de entrada es menor

$$c = \arg \min_i \|\mathbf{x} - \mathbf{m}_i(t)\| \; i = 1, 2, \ldots, N, \tag{2.15}$$

donde $\|^\circ\|$ representa la norma euclídea.

Posteriormente, se lleva a cabo una **etapa cooperativa** y adaptativa, donde los pesos de las neuronas ganadoras y sus vecinas se adaptan al nuevo vector de entrada

$$\mathbf{m}_i(t+1) = \mathbf{m}_i(t) + \alpha(t) h_{ci}(t)[\mathbf{x}(t) - \mathbf{m}_i(t)] \tag{2.16}$$

donde $\alpha(t)$ es la **velocidad de aprendizaje** o *step size*, la **función de vecindad** $h_{ci}(t)$ es, generalmente, una función gaussiana de la distancia entre las coordenadas del espacio de salida

$$h_{ci}(t) = exp\left(-\frac{\|\mathbf{g}_i - \mathbf{g}_c\|^2}{2\sigma(t)^2}\right) \tag{2.17}$$

y $\sigma(t)$ es un parámetro que controla la cobertura efectiva de la función de vecindad. La relación de vecindad dicta la **topología del mapa**, que generalmente es rectangular o hexagonal. Estos parámetros que rigen la adaptación son críticos para el éxito o fracaso de la proyección y, en general, su valor se va disminuyendo con el tiempo (*annealing*).

El proceso de entrenamiento crea una especialización de las neuronas a ciertas áreas del espacio de entrada. Para conseguirlo, las neuronas de la red deben exponerse a un número suficiente de patrones que garantice que el proceso de autoorganización se desarrolla adecuadamente (Haykin, 1994). La regla de aprendizaje arrastra a la neurona ganadora y a sus neuronas vecinas hacia cada nuevo vector de entrada, como una red flexible que se pliega sobre el conjunto de datos de entrada. Por esa razón emerge un mapa que preserva la topología del espacio de entrada mediante su ordenación espacial que proporciona una representación fiel de las características importantes de la entrada (Kohonen, 1990; Haykin, 1994). Los vectores d-dimensionales de las neuronas de la red se modifican de tal manera que en las zonas donde hay una densidad mayor de datos se localiza una cantidad mayor de neuronas. Es decir, las regiones inducidas en el espacio de visualización son tanto mayores cuanto más intenso es el estímulo al que representan, por lo que se obtiene una cierta preservación de la distribución de probabilidad de entrada.

El SOM puede considerarse una generalización no lineal del análisis de componentes principales (Ritter *et al.*, 1992). De hecho, se trata de una aproximación a la discretización de las curvas o superficies principales (Mulier y Cherkassky, 1995), la generalización natural de las componentes principales que se presentó en la sección 2.2.

Debido a su naturaleza heurística, el algoritmo de Kohonen es simple e intuitivo. Sin embargo, por esa misma causa, resulta muy difícil de analizar matemáticamente y los resultados producidos son de una aplicabilidad limitada (Cottrell *et al.*, 1998). Una de las mayores limitaciones, que constituye un problema a la hora de analizar o tratar de optimizar el algoritmo, es la ausencia de una función de energía para una distribución continua de entradas (Erwin *et al.*, 1992) sobre la que pueda realizarse un descenso de gradiente. La razón por la que no es posible obtener una función válida es que la definición de BMU, que determina la probabilidad de asignación de una muestra a una neurona, no es una parte intrínseca de la función de error. Esta

dependencia no considerada en la definición del error provoca problemas en las fronteras de las regiones de Voronoi, ya que las entradas podrían encontrarse a la misma distancia de dos neuronas diferentes (Heskes, 1999) o, en cada actualización, la frontera de la región de Voronoi podría deslizarse sobre el vector de entrada, que pasaría a ser representado por un vector prototipo diferente (Claussen, 2005).

Puesto que es inviable para la definición original del SOM, solamente es posible contar con una función de energía válida si se realizan cambios en la definición de la neurona ganadora. La solución propuesta por (Heskes, 1999; Graepel *et al.*, 1998) es relajar la definición de neurona ganadora (ecuación 2.15) de la siguiente manera:

$$c = \arg\min_i \sum_j h_{ij} \|\mathbf{x} - \mathbf{m}_j\|^2, \tag{2.18}$$

e incluir un término de entropía que actúa como parámetro de regularización para evitar mínimos locales. Así obtienen una función de energía libre, que puede minimizarse directamente por medio de un algoritmo EM. Este algoritmo (Graepel *et al.*, 1998) recibe el nombre de Soft Topographic Vector Quantization (STVQ).

Por otra parte, la convergencia del SOM a un estado estacionario solamente esta probada para el caso de una dimensión (Cottrell *et al.*, 1998) y la complejidad computacional del algoritmo es $\mathcal{O}(nmd)$, donde d es la dimensión de los datos de entrada, m el número de neuronas y n el número de datos (Vesanto, 2000).

2.4.2. Entrenamiento

Durante el entrenamiento, los datos de entrada se introducen varias veces (varias épocas) para facilitar la convergencia del mapa.

Existen dos algoritmos de entrenamiento: el **entrenamiento secuencial**, que se utilizó para presentar la red en el apartado anterior, y el **entrenamiento *batch***. Mientras que en el algoritmo secuencial de entrenamiento los datos se presentan de uno en uno y, en cada paso, se calculan las distancias entre el vector de entrenamiento y todas las neuronas de la retícula (Kohonen, 2001), el entrenamiento *batch* opera con los datos de entrenamiento de forma conjunta. El algoritmo consiste en una iteración de punto fijo donde:

1. Se calculan las BMUs $\mathbf{b}_i$ para todos los datos de entrada $\mathbf{x}_i$.

2. Se actualizan las neuronas mediante la media ponderada de las entradas

$$\mathbf{m}_j = \frac{\sum_{i=1}^{k} h_{b_i j}\mathbf{x}_i}{\sum_{i=1}^{k} h_{b_i j}} \tag{2.19}$$

El paso 2 se puede implementar de un modo más eficiente con la siguiente media ponderada

$$\mathbf{m}_j = \frac{\sum_{i=1}^{M} N_i h_{ij}\bar{\mathbf{x}}_i}{\sum_{i=1}^{M} N_i h_{ij}} \tag{2.20}$$

donde

$$\bar{\mathbf{x}}_i = \frac{1}{N_i}\sum_{\mathbf{x}_k \epsilon \mathbf{V}_j} \mathbf{x}_k \tag{2.21}$$

son los centroides de las regiones de Voronoi y N_i el número de muestras de entrada contenidas en el conjunto de Voronoi i.

El entrenamiento *batch* es más rápido, pues posibilita el uso eficiente de operaciones matriciales, y presenta menos problemas para converger que el algoritmo secuencial, que se puede considerar su aproximación estocástica. Sin embargo, tiene como inconvenientes que no puede adaptar los vectores de codificación ante la llegada de nuevos datos de entrenamiento y que puede tener efectos negativos en la representación de los clusters.

En cualquier caso, el éxito del mapa podría depender de una elección correcta de los parámetros de la red, especialmente del parámetro de aprendizaje, en el entrenamiento secuencial, y del tamaño de la vecindad. No obstante, para el ajuste de los mismos solamente se cuenta con criterios heurísticos. Asimismo, el número de neuronas de la red o escala del modelo afecta a la capacidad de precisión y generalización, que son objetivos contrapuestos. Un SOM con gran número de neuronas permite una regresión más precisa mientras que un SOM con pocas neuronas logra una mayor generalización (Alhoniemi *et al.*, 1999).

A continuación se describen algunas consideraciones importantes acerca de las etapas del algoritmo.

Preprocesado y normalización de los datos. El preprocesamiento es un proceso importante (Kohonen, 2001) y existen varias tareas que se pueden realizar para optimizar el conjunto de datos para su posterior procesamiento:

- Centrarse en un subconjunto de datos.
- Eliminar datos erróneos utilizando conocimiento a priori del dominio del problema y sentido común.
- Realzar datos importantes que aparezcan con baja frecuencia en los datos de entrada (Kohonen, 2001).
- Normalizar o estandarizar. Los datos del sistema pueden ser de magnitudes muy dispares, por lo que es necesario realizar un proceso de normalización o de estandarización para que las componentes con valores altos o alta varianza no resten importancia a los demás. En el primer caso, los datos se escalan para que todos los elementos se encuentren entre 0 y 1, mientras que en el segundo caso se resta la media y se divide por la desviación estándar. Otros métodos más complejos como el *whitening*, que tiene en cuenta las correlaciones entre variables, también son aplicables.
- Repetir datos, cuando estén disponibles menos de los necesarios, de forma cíclica o en un orden permutado aleatoriamente (Kohonen, 2001).

Inicialización. Se pueden utilizar diferentes métodos de inicialización de los pesos de las neuronas:

- **Inicialización aleatoria**, en la que los pesos iniciales deberían ser diferentes para cada neurona y con valores pequeños (Haykin, 1994).
- **Inicialización con muestras iniciales**. Tiene la ventaja de que los pesos iniciales se encuentran en la misma zona del espacio que los datos de entrada (Hollmén, 1996).
- **Inicialización lineal**. Los vectores de pesos se inicializan de forma ordenada (Kohonen, 2001).

Competición. El espacio continuo de patrones de entrada se proyecta en un espacio discreto de salida de neuronas, mediante un proceso de competición entre las neuronas de la red. Se utiliza la distancia Euclídea,

$$d_E(\mathbf{x}, \mathbf{y}) = \|\mathbf{x}\| = \sqrt{\sum_{i=1}^{n} x_i^2}, \tag{2.22}$$

a no ser que exista otra métrica mejor para la naturaleza del problema (Kohonen, 2001).

Cooperación. Sea $h_{c,i}$ la **vecindad topológica** con centro en la neurona ganadora c. La función de vecindad depende del instante de tiempo y de la distancia $d_{c,i}$ en la malla, $d_{c,i} = \|\mathbf{g}_c - \mathbf{g}_i\|$. La **función de vecindad** debe satisfacer dos requisitos:

- Su mayor valor se encuentra en la neurona ganadora c, para la que la distancia $d_{c,i}$ es cero.
- Decae a cero cuando la distancia tiende a infinito.

Aunque se pueden utilizar varios tipos de funciones de vecindad, la más típica es la función Gaussiana, que no depende de la localización de la neurona ganadora. Otras funciones que se puede utilizar como función de vecindad son la función burbuja, la función *cut-Gauss* o la función Epanechikov. La vecindad gaussiana se define como:

$$h_{ci}(t) = exp\left(-\frac{{d_{c,i}}^2}{2\sigma^2(t)}\right) \tag{2.23}$$

donde $\sigma(t)$ es la **anchura de la vecindad**, que varía en el tiempo. Para hacer depender σ del tiempo se puede usar una función exponencial (Ritter *et al.*, 1992):

$$\sigma(n) = \sigma_0 \cdot exp\left(-\frac{n}{\tau_1}\right) \qquad n = 0, 1, 2, \ldots \tag{2.24}$$

donde σ_0 es el valor inicial de σ y τ_1 es una constante de tiempo o bien hacer

$$\sigma(n) = \frac{a}{1 + d\tau} \tag{2.25}$$

Es conveniente que las dimensiones de la red se correspondan aproximadamente a la distribución de los datos, para lo cual se podría realizar una previsualización de los mismos por medio de alguna herramienta de escalado multidimensional como la proyección de Sammon (Kohonen, 2001). Si el tamaño inicial con el que se comienza es demasiado pequeño, el mapa no se ordena globalmente.

La topología es generalmente rectangular o hexagonal, que algunos autores consideran preferible por su mayor isotropía. La forma de la malla puede influir en la facilidad de la red elástica formada por los vectores de referencia m_i para orientarse y estabilizarse de acuerdo a la función de densidad de probabilidad $p(x)$. La vecindad de las neuronas que se encuentran en los extremos del espacio del mapa contiene menos neuronas que la vecindad de las neuronas del centro del mapa. Esto provoca el indeseable efecto borde del algoritmo, que provoca que las neuronas de los extremos sean atraídas hacia los nodos del interior de la malla.

Se han propuesto soluciones para evitarlo, una de las cuales es la utilización de un mapa toroidal (Ultsch, 2003a), de modo que no existan extremos y todas las neuronas tengan el mismo número de vecinas.

Adaptación Sináptica. La neurona ganadora y sus vecinas se actualizan moviéndose ligeramente hacia la muestra de entrada. La adaptación de los pesos sinápticos de la red se puede descomponer en dos fases (Kohonen, 2001):

- **Fase de auto-organización u ordenación**. Es la primera fase del proceso y debe ser relativamente corta. El parámetro α debería empezar con unos valores razonablemente altos (cercanos a la unidad) y disminuir gradualmente. Se podría utilizar una regla de actualización similar a la de la ecuación (2.25). La vecindad debería incluir inicialmente casi todas las neuronas y disminuir lentamente (Kangas, 1994).
- **Fase de convergencia.** En esta segunda fase se afina el mapa. Es cuando se emplea la mayor parte del tiempo de computación. El parámetro de aprendizaje debe mantener un valor pequeño, del orden de 0,01, pero sin llegar a cero.

2.4.3. Propiedades útiles

En este apartado, se pretenden explicar las propiedades que hacen del mapa auto-organizado un algoritmo idóneo para la visualización y exploración de grandes volúmenes de datos; en particular, para el análisis o supervisión de procesos industriales.

En primer lugar, el SOM proporciona una buena aproximación a la **cuantización de vectores**, dividiendo el espacio de entrada en una colección finita de regiones de Voronoi. De hecho, sus ecuaciones presentan cierto parecido con las de los algoritmos LBG y *K-means*. No obstante, debido a la incorporación del concepto de vecindad, el SOM no consigue una transferencia óptima de información en cuanto al error de cuantización, ya que, al mismo tiempo que trata de aproximar los vectores prototipo a los datos impone una restricción, una estructura que impide operaciones que violen la ordenación topológica. Sin embargo, esto no es una desventaja, pues hace posible la preservación de la topología que provoca que la retícula de salida sea una representación de dimensión reducida de los datos de entrada. Es decir, al mismo tiempo que reduce el número de vectores, realiza una proyección de los mismos al plano 2D. Esta es la ventaja del SOM frente a otros algoritmos de *clustering* o VQ. De hecho, para propósitos en los que la información topológica fuese irrelevante, el mapa auto-organizado no sería el cuantizador de vectores más adecuado.

La preservación de la topología refleja, de forma intuitiva, la similitud en la estructura de vecindad de los conjuntos de datos de entrada y de salida. Se puede definir de un modo más formal (Villmann *et al.*, 1997) como la condición de que las proyecciones que transforman los datos en las coordenadas de sus neuronas BMU y las coordenadas en los datos sean continuas con respecto a cada topología. El concepto de topografía, también muy presente en la bibliografía, tiene un significado generalmente equivalente.

El modo de preservar la topología en el SOM es diferente al de los métodos de escalado multidimensional clásicos que tratan de optimizar la proyección globalmente (Kaski, 1997). El SOM trata de formar una proyección correcta localmente, ya que solamente las distancias locales contribuyen al error y el SOM no conserva las distancias de la distribución de entrada de forma óptima.

Se han propuesto varias métricas (Bauer *et al.*, 1999) para estimar el grado de preservación, como el producto topográfico, la ρ de Spearman, la Z de Zrehen o la función topográfica. En general, estas métricas coinciden en sus resultados, pero sus cálculos son más complejos que el propio algoritmo en sí (Heskes, 1999).

Por otra parte, el mapa **refleja variaciones en la distribución de probabilidad** de entrada, ya que las regiones con muchos datos son representadas mejor en el SOM, ocupando mayor parte del espacio de salida, lo que permite representar estas regiones con mejor resolución. Esta propiedad se puede cuantificar por medio del **factor de magnificación**, que describe la relación entre los datos y la densidad de vectores de pesos en la salida cuantizada y se expresa como una relación de potencias entre la densidad de vectores prototipo y la distribución de probabilidad P de los datos. Está relacionado con otras propiedades del mapa como su error de reconstrucción o el concepto de información mutua (Hammer y Villmann, 2003; Villmann y Claussen, 2006). Los autores coinciden en señalar que el factor de magnificación del SOM es menor que 1 y depende de la función de vecindad. Eso indica que, aunque el SOM sigue la densidad de datos subyacente, enfatiza ciertas regiones de la distribución de datos. Concretamente, el SOM tiende a sobrerrepresentar regiones de baja densidad y subrepresentar regiones de alta densidad (Haykin, 1994).

Las capacidades de **visualización** del SOM son diferentes a las de otros métodos de reducción de la dimensión, ya que los vectores se proyectan en posiciones fijas de la rejilla de salida. Esta naturaleza **compacta** y **ordenada** de la visualización facilita su comprensión (Kaski, 1997) pues la proyección se realiza en un espacio delimitado. Además, el uso de una representación homogénea para múltiples aplicaciones (visualización de distancias, componentes del vector, ...) permite que el analista se familiarice y pueda interpretar la información más fácil y rápidamente.

Una de las posibilidades visuales es la visualización de las distancias entre vectores prototipo sobre esa representación ordenada. Esta representación permite **analizar visualmente la estructura de clusters**. Aunque dicho análisis visual es subjetivo e implica riesgos (Flexer, 2001), no es necesario suponer de antemano que los clusters se ajustarán a una determinada forma, una restricción común a muchos algoritmos de *clustering*.

Esta serie de cualidades que sugieren que el SOM podría ofrecer una visualización fiable. Sin embargo, no es fácil cuantificar en qué medida la visualización representa los datos. Dos métricas útiles para este propósito son la *fiabilidad* (*trustworthiness*), definida como la propiedad de que los vecinos de una neurona en el espacio de salida se encuentren también cercanos en el espacio de entrada, y la *continuidad*, que los puntos cercanos en el espacio de entrada sean proyectados cerca en el espacio de salida (Venna y Kaski, 2006). La fiabilidad es la cualidad más importante pues garantiza que al menos una parte de las similitudes se percibe correctamente. En todas las pruebas empíricas realizadas por Venna (2007) frente a otros algoritmos de reducción de la dimensión, el SOM presenta, en general, los mejores resultados en cuanto a fiabilidad y mantiene una buena continuidad. Solamente el CCA en cuanto a fiabilidad y el SNE en cuanto a continuidad presentan unos resultados comparables o mejores.

En resumen, el SOM presenta varias ventajas para el propósito de esta tesis, aunque también presenta algunos inconvenientes que es necesario conocer. Algunos de ellos, como la ausencia de una función de energía a minimizar o que no se pueda garantizar la convergencia, ya han sido expuestos previamente. Para paliarlos, se han propuesto algoritmos alternativos que, en ciertas condiciones pueden ser más adecuados que el SOM o complementar sus resultados.

Algunos autores (Flexer, 2001) sostienen que la ejecución secuencial de un algoritmo de cuantización de vectores seguido de uno de escalado multidimensional proporciona mejores resultados: un mejor error de cuantización y una mejor preservación de las distancias. No obstante, otros autores consideran que el uso de métricas de preservación de distancias para evaluar el SOM frente a métodos como la proyección de Sammon es discutible (Kaski, 1997), puesto que el escalado que produce el SOM no es métrico (Yin, 2008). Otros autores (Bauer *et al.*, 1999) sostienen que es difícil revelar qué método es más apropiado en general con un pequeño conjunto de casos, ya que cada método para la obtención de mapas topográficos tiene sus méritos y defectos.

Por otra parte, el algoritmo GTM facilita el análisis matemático porque se basa en principios estadísticos, al contrario que el SOM, que es un algoritmo heurístico. GTM define una densidad de probabilidad explícita y, como consecuencia, existe una función de coste bien definida, se garantiza la convergencia a un máximo local y existe un medio de comparar diferentes elecciones de parámetros. No obstante, si

se asocia un modelo generativo de probabilidad al SOM (Lampinen y Kostiainen, 2002) y se utiliza la definición de neurona ganadora de (2.18), se solventan la mayor parte de las problemas y el algoritmo es más fácil de interpretar y visualizar. Tanto el GTM como este nuevo modelo SOM tienen mayor coste computacional que el SOM tradicional.

No obstante, las visualizaciones obtenidas a partir de otros algoritmos se pueden combinar con las proporcionadas por el SOM utilizando alguna técnica de enlace (Himberg, 1998).

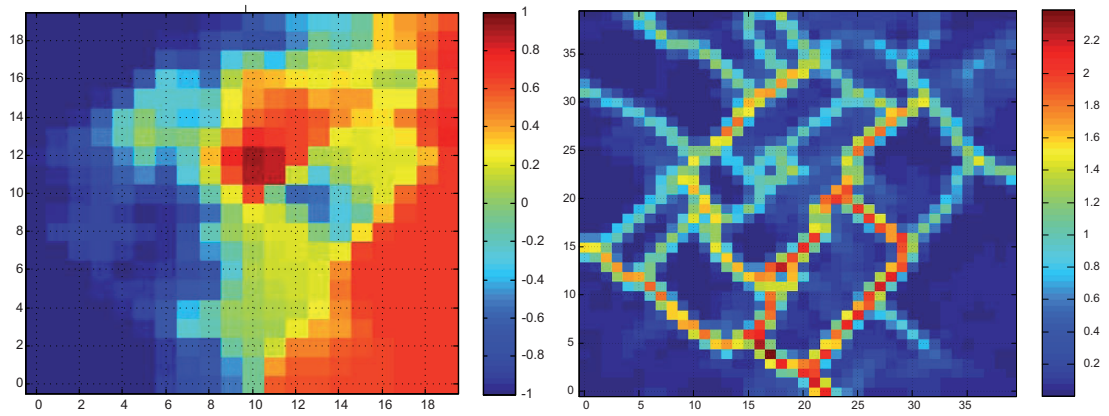
2.4.4. Herramientas de visualización

Como se expuso en el apartado anterior, el espacio reticular topológicamente ordenado del SOM es un marco magnífico para la visualización. Sobre él se pueden mostrar propiedades escalares o variables del espacio de entrada, distancias entre neuronas, etc. Las coordenadas de las neuronas permiten fijar la posición y el uso de un código de color permite mostrar de forma intuitiva y homogénea los valores.

Debido a sus cualidades, se han propuesto y aplicado en la bibliografía multitud de visualizaciones. A lo largo de esta sección se presentan brevemente algunas de las técnicas de visualización más destacadas asociadas al SOM.

Matriz de distancias Unificada. También conocida como ***u-matrix*** o **mapa de distancias** (Ultsch y Siemon, 1990), es un mapa que permite describir la estructura de clusters de los datos, representando como propiedad escalar, por medio de colores, la distancia promedio de cada neurona a sus vecinas más próximas, de acuerdo con una función de vecindad (ver Fig. 2.4). Esta distancia promedio interneuronal guarda una estrecha relación con la densidad de neuronas en una región determinada del espacio de entrada. Densidades elevadas (o distancias pequeñas) representan regiones densamente pobladas de datos del proceso, es decir, *clusters*. Al contrario, las zonas con baja densidad (distancias grandes) pueden verse como separadores de *clusters*. Por ello, la inspección visual del mapa a menudo permite realizar un *clustering* manual. Sin embargo, esta tarea puede ser ardua e inconsistente, por lo que se han propuesto métodos automáticos de etiquetado (Domínguez, 2003).

P-Matrix* y *u*-Matrix*.** En el *u-Matrix*, las distancias dentro del *cluster* se tratan de la misma manera que las distancias entre *clusters*. Esto puede impedir la detección correcta de *clusters* en algunos conjuntos de datos. Para solucionar este problema, en (Ultsch, 2003b) se define la ***p-Matrix de un SOM de forma análoga a la *u-Matrix*, pero asociando a la posición de cada neurona una estimación de la



(a) Plano de componentes **(b)** Matriz de distancias

Figura 2.4. Herramientas de visualización I.

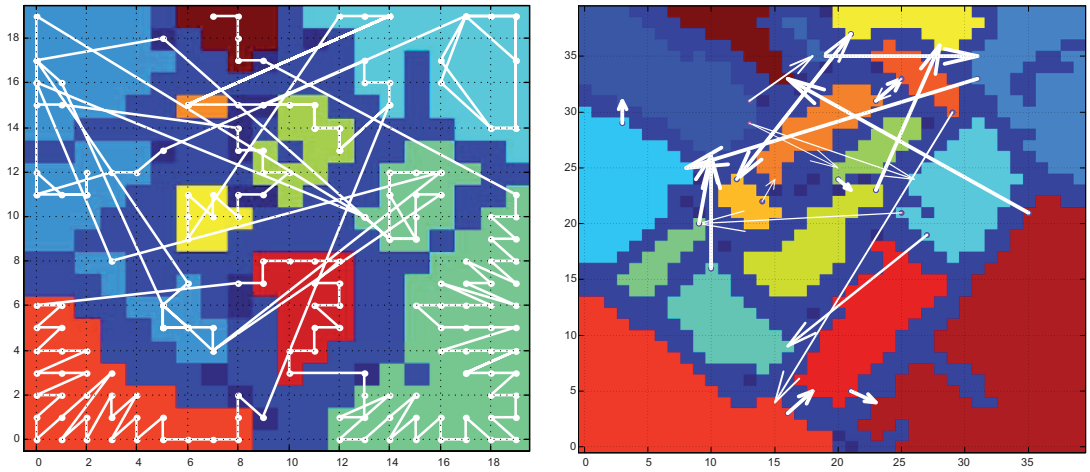
densidad (p-altura) en lugar de la distancia. Por otra parte, la ***u*-Matrix** (Ultsch, 2005) combina la *u-matrix* y la *p-matrix* de tal modo que, cuando la densidad de datos en una neurona es igual a la densidad de datos media, los valores de la *u*-matrix* son las de su *u-matrix*, cuando la densidad es mayor los valores son menores y cuando es menor, son mayores.

Campos vectoriales. Esta visualización de la estructura de clusters utiliza flechas que apuntan al centro de cluster más cercano (Pölzlbauer *et al.*, 2006).

Plano de componentes. También denominados **mapas de características**, permiten describir variables del proceso mediante los valores que toman sus vectores prototipo (Tryba *et al.*, 1989). Cada plano representa el valor escalar de una variable en cada nodo del SOM y, por ello, al igual que en el mapa de distancias, se utilizan colores para representar los valores asignados a cada neurona (ver Fig. 2.4). Comparando planos de componentes se puede detectar si dos componentes están correlacionados. No obstante, los planos de componentes podrían sugerir dependencias estadísticas inexistentes (Lampinen y Kostiainen, 2000).

Hit map. Se puede generar un **histograma** sobre el propio mapa que represente la frecuencia de aparición de las BMU (Vesanto, 1999) o proporción de datos que representa cada neurona.

Mapa de modelos. Se trata de una generalización del concepto de plano de componentes en el que, dada una ecuación explícita que sea función de las varia-



(a) Trayectoria **(b)** Mapa dinámico de transiciones

Figura 2.5. Herramientas de visualización II.

bles del espacio de entrada, los mapas de modelos permiten evaluar el grado de cumplimiento de dicho modelo analítico para las neuronas del espacio de visualización (Díaz *et al.*, 2005). Además de ecuaciones explícitas, se pueden utilizar reglas borrosas (Cuadrado Vega, 2002).

Mapa de etiquetas. Puede ser interesante mostrar etiquetas identificativas de los datos de entrada en el lugar en que se proyecta su BMU asociada. Si las etiquetas son suficientemente significativas pueden resultar útiles para examinar el mapa (Rauber y Merkl, 1999).

Mapa de estados. Se puede definir un mapa que etiquete o coloree las neuronas de acuerdo al cluster al que pertenecen (Fuertes *et al.*, 2005). En el ámbito de la supervisión de procesos, cada cluster puede considerarse una condición de proceso (Ahola *et al.*, 1999).

Proyección de la trayectoria. La BMU del dato actual, proyectada sobre la malla reticular, se puede interpretar como el punto de operación del proceso. Conectando la secuencia de puntos de operación a lo largo del tiempo (ver Fig. 2.5), se genera una trayectoria de la evolución del proceso en la malla reticular (Kasslin *et al.*, 1992; Tryba y Goser, 1991), a partir de la cual es posible determinar qué clusters o condiciones de proceso son accesibles desde una dada y cuál es la probabilidad de transición entre ellas (Fuertes *et al.*, 2007).

Mapa dinámico de transiciones. Partiendo del mapa de estados y tras un cálculo de *probabilidades de transición* entre condiciones de proceso, se puede definir una representación que muestre las condiciones del proceso sobre la malla neuronal, como en el mapa de estados, y las probabilidades de transición entre condiciones, con líneas de grosor proporcional a dicha probabilidad (Fuertes *et al.*, 2007). Las probabilidades de transición se calculan de la siguiente manera:

$$p_{C_iC_j} = \frac{n_{C_iC_j}}{\sum_{k=1}^{n_t} n_{C_iC_k}} \tag{2.26}$$

donde $n_{C_iC_j}$ es el número de transiciones de la condición de proceso C_i a C_j, obtenidas a partir del análisis de la trayectoria, y n_t es el número total de condiciones de proceso.

La visualización simultánea del mapa dinámico de transiciones y del estado actual facilita su supervisión en línea, la predicción de comportamientos futuros y la detección de fallos, reflejados por accesos del sistema a condiciones inaccesibles (con probabilidad nula en el modelo) o por transiciones entre condiciones no consistentes con las rutas de transición. En la Fig. 2.5 se muestra un mapa dinámico de transiciones.

Representación visual de los residuos. En el apartado 1.1.2 se definía el concepto de residuo, que se puede relacionar con el **error de cuantificación** (Kohonen *et al.*, 1996) entre un vector de entrada y un SOM entrenado[4],

$$\mathbf{r}(t) = \mathbf{x}(t) - \mathbf{m}_c(t). \tag{2.27}$$

El residuo $\mathbf{r}(t)$ tiene n componentes que representan las desviaciones individuales de cada variable respecto a sus valores esperados, tanto por defecto como por exceso. Estas componentes son fácilmente visualizables cuando la dimensión del espacio de entrada es baja. Sin embargo, no lo es cuando esta dimensión aumenta, por lo que es necesario utilizar una representación visual de apoyo (Díaz y Hollmén, 2002). Se propone una representación gráfica en la que el eje vertical corresponde a las componentes del vector, el eje horizontal representa el tiempo y se emplea un código de color para mostrar las desviaciones de los valores de las componentes, como se puede observar en la Fig. 2.6.

[4]No obstante, la aplicación directa de esa idea puede dar lugar a falsos negativos, ya que la desviación podría hacer que la muestra de entrada se aproxime a otra neurona del proceso con la que tenga menor error de cuantización, aunque sea incoherente con su evolución dinámica. La utilización de información sobre la evolución dinámica del proceso puede ayudar a mejorar los resultados de detección e identificación de fallos.

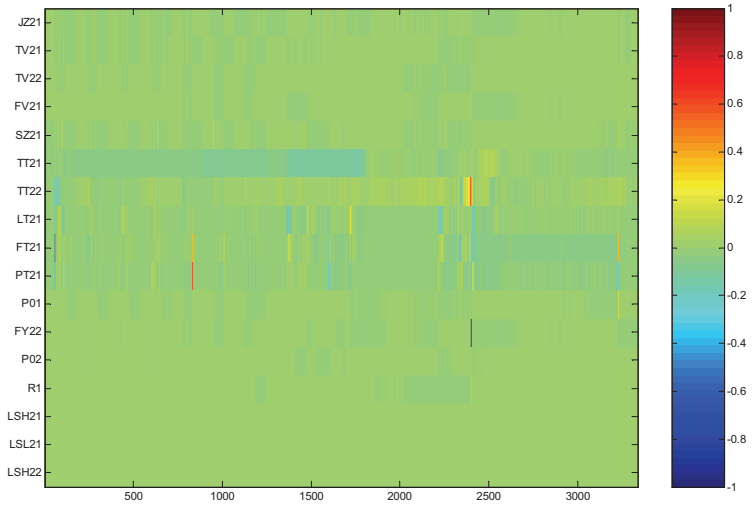


Figura 2.6. Representación gráfica de los residuos.

Modelo dinámico de trayectorias. Para este propósito, se utiliza el KR-SOM, una modificación del algoritmo SOM en la que la secuencia de activaciones de la BMU forma una trayectoria 2D continua más fácil de analizar. Esta trayectoria, a su vez, se utiliza como entrada para otro SOM (Fuertes *et al.*, 2007) con un espacio de entrada de 6 dimensiones que incluye las coordenadas, el vector unitario de velocidad, la inversa del módulo de la velocidad y el módulo de la velocidad del sistema en el espacio de entrada.

El aprendizaje de los vectores de flujo de trayectorias permite definir herramientas de visualización más sensibles a cambios leves (Fuertes *et al.*, 2007) que, por ejemplo, permiten representar las discrepancias entre la trayectoria modelada por el SOM y la generada por otra ejecución del sistema. Así, se pueden detectar desviaciones en la dirección o cambios de velocidad o aceleración, que indiquen un comportamiento anómalo.

Mapa de diferencias. Estos mapas están orientados a la comparación de procesos que se rigen por el mismo patrón de funcionamiento (Fuertes, 2006) para conocer las desviaciones existentes. El punto de partida del análisis son dos instancias entrenadas del SOM, sobre las que se define una función *diferencia* que asigna a cada neurona del modelo B la diferencia entre su vector prototipo y el de la BMU de esta neurona en el modelo A. Cuando el criterio de selección de la BMU es el de la más próxima según una cierta función de medida, la diferencia entre modelos se considera estática. Aunque los *mapas estáticos* pueden conducir a resultados, la carencia de conocimiento sobre la evolución dinámica induce, en

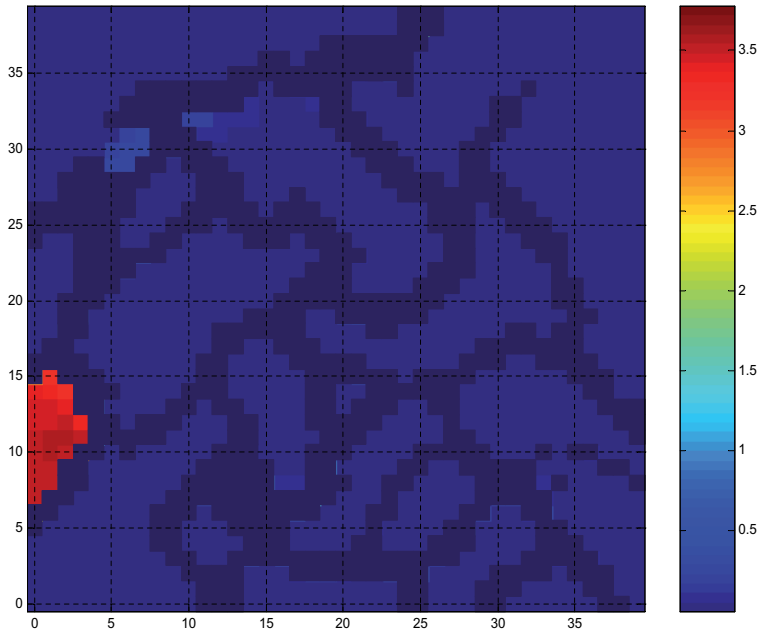


Figura 2.7. Mapa de diferencias.

ocasiones, interpretaciones erróneas, ya que la BMU elegida puede no corresponderse con el estado más similar del modelo A. Este problema se puede minimizar incorporando información sobre las transiciones, lo que da lugar a los *mapas de diferencias dinámicos*. Para facilitar la interpretación, se pueden visualizar tanto el *mapa de módulos de diferencias* como el *mapa de componentes de diferencias*. Se muestra un ejemplo de estos datos en la Fig. 2.7.

2.4.5. Extensiones

Debido a la potencia y versatilidad del algoritmo, numerosos autores han tratado de ampliar la funcionalidad del SOM, mejorar sus prestaciones o cubrir sus carencias. Se pueden utilizar métricas diferentes, aplicar a otro tipo de entradas, optimizar o acelerar el algoritmo, introducir procesamiento o estructura jerárquica, permitir que crezca, tener en cuenta el tiempo y la dinámica, etc. (Kohonen, 2001).

Por ejemplo, desde un punto de vista de **teoría de la información**, una mejora del SOM consistiría en la definición de un mapa topográfico óptimo que maximizase la información mutua. Este mapa óptimo reproduciría la distribución de probabilidad de entrada exactamente, es decir, tendría un factor de magnificación $\rho = 1$. Linsker (1989) propone un mapa que utiliza este criterio, conocido

como *infomax* (preservación máxima de la información mutua), pero sus cálculos requieren integrales costosas computacionalmente. Una modificación del SOM tradicional parametrizable, el Winner Relaxing SOM, propuesto por Claussen (2005) permite obtener también un factor de magnificación igual a 1 modificando la definición de neurona ganadora para dotarla de mayor importancia.

En el apartado 2.4.1 se presentó un algoritmo de origen probabilístico, STVQ, para optimizar una función de coste válida similar al SOM. Partiendo de ese punto de partida, los mismos autores definen una generalización que utiliza funciones kernel, el Kernel-Based Soft Topographic Mapping o STMK (Graepel *et al.*, 1998). Se han propuesto más algoritmos (Van Hulle, 2002), que partiendo de un marco probabilístico o de teoría de la información aplican la **teoría de kernels** (Schölkopf y Smola, 2001) al SOM. Una justificación teórica para su uso, es decir, para aplicar el SOM en un espacio de características de alta dimensión, es la mayor potencia que proporciona a la red para tratar con superficies complicadas del espacio de entrada. Desde un punto de vista práctico, también se puede justificar por la flexibilidad que introduce para controlar la anchura de las regiones de Voronoi por medio del parámetro de anchura σ. No obstante, los algoritmos propuestos por el momento requieren mayor esfuerzo computacional que el SOM, presentan más parámetros libres a ajustar y permiten, en el caso mejor, solamente una mejora de rendimiento marginal (Lau *et al.*, 2006).

Otras propuestas de extensión del SOM van a orientadas a permitir su **crecimiento**. El objetivo es paliar una de las desventajas más serias del SOM: la necesidad de especificar previamente y con poca información la arquitectura de la red, la cual tiene una influencia importante en el rendimiento de la misma. Existen diferentes propuestas como mapas 2D con topología arbitraria (Alahakoon *et al.*, 2000), mapas jerárquicos, (Dittenbach *et al.*, 2000), algoritmos basados en grafos (Furao y Hasegawa, 2006) o SOMs k-dimensionales (Fritzke, 1994). Nuevos problemas como el modo de crear o eliminar nuevas neuronas y el modo de visualizar datos aparecen asociados a este tipo de redes.

Como ejemplo de algoritmos parcialmente basados en el SOM podemos destacar el algoritmo **Neural Gas** (Martinetz *et al.*, 1993). Esta técnica es similar al SOM pero sin la imposición de una estructura de vecindad predefinida, lo que permite trabajar mejor en espacios de entrada complicados. La actualización no se aplica en función de la distancia al vector ganador sino en función del orden de distancia. No obstante, la falta de un espacio de salida limita su uso para visualización.

También existen modificaciones del SOM que le dotan de un carácter **continuo**. El *Kernel Regression SOM* (KR-SOM) es una versión interpolada del SOM por medio de GRNN que intenta paliar las desventajas que genera el carácter discreto de éste (Díaz *et al.*, 1999). El *PSOM* (SOM Parametrizado) (Walter *et al.*,

2000), por su parte, es una generalización continua del SOM que muestra excelentes capacidades de generalización a partir de pequeños conjuntos de datos de entrenamiento.

Otras variaciones del SOM pretenden mejorar su **visualización**. Por ejemplo, el algoritmo *ViSOM* (Yin, 2002) restringe la contracción lateral del SOM, regularizando sus distancias interneuronales, con el objetivo de preservarlas lo más posible. La malla que produce permite una medida cuantitativa, directa y visual de las distancias en el mapa. Por otra parte, un problema derivado de la utilización del plano euclídeo como espacio de visualización es que la vecindad que se encuentra alrededor de un punto está muy restringida. Los espacios hiperbólicos, que tienen una curvatura negativa uniforme, permiten que la vecindad alrededor de un punto se incremente de forma exponencial con su radio. Por eso se ha definido un *SOM hiperbólico* (Ritter, 1999) que permite la inspección visual de grandes estructuras jerárquicas.

Finalmente, cabe destacar las propuestas que extienden el SOM para procesamiento de secuencias temporales y modelado de la dinámica. Puesto que son importantes para los objetivos de esta tesis, se explican de forma más detallada en la sección 2.6.

2.4.6. Aplicaciones

La bibliografía sobre el SOM es muy extensa. Al menos 7718 artículos sobre el tema habían sido catalogados en 2003 por la Univ. Tecnológica de Helsinki (Kaski *et al.* (1998); Oja *et al.* (2003); http://www.cis.hut.fi/research/refs/). En (Kohonen, 2001), se enumeran aplicaciones del SOM en áreas tan diversas como tratamiento de imágenes, reconocimiento del habla, telecomunicaciones, control y supervisión de procesos, biomedicina, robótica, neurofisiología o lingüística.

En cuanto al uso, las principales aplicaciones del SOM son la visualización, el *clustering* y el modelado local. Ya se han descrito en este documento las buenas cualidades del SOM para visualizar información (sección 2.4.3), así como varias herramientas concretas que utilizan estas propiedades (sección 2.4.4). En cuanto al *clustering*, Vesanto y Alhoniemi (2000) han realizado experimentos que comparan la aplicación de métodos aglomerativos y partitivos sobre los datos frente a su aplicación sobre los vectores prototipo del SOM y los resultados han sido comparables con los métodos directos, siendo el *clustering* de las neuronas del SOM considerablemente más rápido, lo que proporciona escalabilidad a conjuntos de datos muy grandes. Por otra parte, el SOM en sí mismo también puede partir el conjunto de datos por medio de la matriz de distancias, ya sea de forma manual o automática, y representar los *clusters* de forma muy directa (Fuertes *et al.*, 2007). Finalmente, esa capacidad de particionar el espacio en regiones es la razón por la

que el SOM se ha utilizado para el modelado local (Vesanto, 1997). Esta aplicación se describe más en detalle en los siguientes apartados.

2.5. El SOM en la supervisión de procesos industriales

El análisis offline de procesos industriales y la supervisión de los mismos han sido uno de los campos de aplicación más prolíficos de los mapas auto-organizados. Se pueden encontrar numerosas referencias de sus aplicaciones en Simula *et al.* (1999), Kohonen *et al.* (1996) y Kohonen (2001). Tiene varios usos en este área (Alhoniemi *et al.*, 1999; Jamsa-Jounela *et al.*, 2003; Frey, 2008), incluyendo detección de fallos, visualización de la condición o estado del proceso, estimación y predicción. Se pueden encontrar ejemplos en diferentes industrias como la metalúrgica (Laine, 1998; Díaz *et al.*, 2003), papelera (Alhoniemi, 2000), química (Ultsch, 1993; Abonyi *et al.*, 2003; Machón y López, 2006); así como en otras aplicaciones industriales relacionadas (De y Chatterjeee, 2002; Postolache *et al.*, 2005). También se ha demostrado su utilización en la supervisión remota vía internet (Domínguez *et al.*, 2007).

El SOM se puede utilizar para identificar el estado del proceso (Kasslin *et al.*, 1992) y visualizar su evolución (Tryba y Goser, 1991; Ahola *et al.*, 1999; Fuertes *et al.*, 2007). También puede predecir ciertos parámetros de un proceso partiendo de sus datos (Hollmén y Simula, 1996) o estimar variables. Asimismo, se puede utilizar para detectar fallos por medio del error de cuantización, si el SOM se entrena utilizando datos de su operación normal (Ypma y Duin, 1997; Díaz y Hollmén, 2002). La utilización del conocimiento sobre la dinámica del proceso puede ayudar a mejorar los resultados de detección e identificación de errores (Fuertes, 2006), pues evita comparaciones incoherentes con la evolución del proceso. También se pueden utilizar tanto las situaciones normales como las de fallo en los datos de entrenamiento y que el SOM clasifique los estados (Vapola *et al.*, 1994).

Esta tesis se centra en la aplicación de los mapas auto-organizados en el modelado y supervisión de procesos industriales, concretamente centrándose en la dinámica. Como se describe en las secciones siguientes, existen aún problemas sin resolver y aplicaciones poco exploradas del SOM para la preservación y visualización de la dinámica de procesos.

2.6. Modelado de la dinámica mediante el SOM

2.6.1. Memoria a corto plazo

El SOM se ha utilizado para el procesamiento de secuencias temporales y extracción de características dinámicas en diferentes áreas, como reconocimiento del habla, control o predicción de series temporales. Para este propósito, tanto el SOM como otras redes neuronales no supervisadas deben poseer algún mecanismo de memoria a corto plazo (Barreto y Araujo, 2002), que permita conservar información del contexto. Existen diversos modelos, como las líneas de retardo, integradores *leaky*, recurrencia o mecanismos de reacción-difusión, que se pueden incorporar al SOM para aproximar la dinámica.

Las **líneas de retardo** implementan de forma directa la idea de reconstruir la dinámica de estados original mediante el uso de un vector cuyos componentes son las muestras de entrada pasadas. Es la memoria con mayor resolución (Principe *et al.*, 1993) pero carece de versatilidad, pues la longitud de las líneas de retardo debe ser elegida a priori e impone una estructura rígida de retardos. Provoca que patrones similares en el tiempo (por ejemplo, con un retardo de diferencia) sean muy diferentes espacialmente y multiplica el número de entradas, tantas veces como retardos haya, por lo que las redes resultantes tienen tantos grados de libertad que podrían no generalizar bien y ser computacionalmente mucho más costosas (Principe *et al.*, 2002). Un buen uso de la metodología de *embedding* requiere un trabajo significativo para calcular una serie de parámetros como el número de muestras, el retardo, etc.

Otra estructura muy utilizada son los **integradores leaky**, cuyo funcionamiento se basa en el concepto de realimentación local y decaimiento exponencial. Una neurona *leaky* tiene un potencial P_i que decae a lo largo del tiempo

$$P_i(t) = \lambda P_i(t-1) + I_i(t) \tag{2.28}$$

donde λ es el parámetro que lo define y $I_i(t)$ la actividad de la neurona en el momento t. Se pueden añadir a las neuronas de la red para recordar secuencias largas a costa de una menor resolución. No obstante, la adaptación de sus parámetros es complicada.

La **recurrencia** en las redes neuronales crea una memoria más flexible y ajustable en la que los pesos de la red no sólo almacenan las entradas, sino el estado de la red. En teoría, se trata de una arquitectura lo suficientemente potente para manejar problemas temporales de una complejidad arbitraria. En la práctica, no obstante, las redes recurrentes son más difíciles de entrenar, pues crean superficies de error complicadas. Puesto que estas estructuras de memoria son recursivas, su entrenamiento implica gradientes que dependen del tiempo (Principe *et al.*, 1993),

cuyo cálculo y consiguiente actualización de los pesos es una tarea difícil y costosa.

Una opción más novedosa (Principe *et al.*, 2002) es explotar mecanismos de **reacción-difusión** para almacenar el historial pasado de la señal de entrada en ondas de actividad. De este modo, la red se auto-organiza de acuerdo a sus correlaciones temporales.

2.6.2. Extensiones del SOM para procesamiento temporal

Recientemente, algunos autores (Barreto y Araújo, 2001; Barreto, 2007; Guimarães *et al.*, 2003) han revisado diversas ampliaciones del algoritmo dirigidas a dotarlo de capacidad para procesar series temporales. Estas ampliaciones se basan, generalmente, en las diversas implementaciones de memoria a corto plazo descritas en el apartado anterior: línea de retardos (Principe *et al.*, 1998; Cho *et al.*, 2006; Vesanto, 1997; Lendasse *et al.*, 2002), integradores *leaky* (Chappell y Taylor, 1993; Koskela *et al.*, 1998), recurrencia (Voegtlin, 2002; Hynna y Kaipainen, 2006; Hammer *et al.*, 2004) o activación-difusión (Principe *et al.*, 2002; Wiemer, 2003). A continuación se describen algunos de los algoritmos más significativos.

Operator Map. Esta modificación, propuesta por Kohonen (1993), hace corresponder a cada neurona del mapa con un operador que analiza el vector $\mathbf{X}(t) = \{\mathbf{x}(t), \mathbf{x}(t-1), ..., \mathbf{x}(t-n+1)\}$ y estima $\mathbf{x}(t+1)$. El error de predicción es el que define la neurona ganadora.

Temporal Kohonen Map. Este modelo (Chappell y Taylor, 1993) define activaciones variables para cada neurona, por medio de integradores *leaky*. El potencial de activación de una neurona se define como

$$V_i(t) = dV_i(t-1) - \frac{1}{2}\|\mathbf{x}(t) - \mathbf{w}_i(t)\|^2, \tag{2.29}$$

donde d define la profundidad de la memoria, es decir, cuántas muestras de tiempo es capaz de recordar. La resolución de la red, en cambio, viene determinada por el número de neuronas. Se elige como neurona ganadora aquella con mayor potencial de activación frente a una entrada. La desventaja de la red TKM es que pierde el contexto con secuencias largas, ya que, debido a la definición del potencial, los últimos valores de la secuencia tienen mayor importancia.

Recurrent SOM. Este algoritmo (Koskela *et al.*, 1998) utiliza los integradores *leaky* a la entrada de la neurona, en lugar de la salida. Puesto que la actividad es vectorial, permite capturar la dirección del error, al contrario que TKM, que solamente mantiene la magnitud de la salida. Por eso, proporciona mejores resultados.

SARDNET(Sequential Activation Retention and Decay Network). Esta red (James y Miikkulainen, 1995) permite almacenar la información de secuencias de entrada, sin las limitaciones de resolución de TKM y RSOM, porque utiliza el orden en el que las neuronas se activan para clasificar secuencias. Para ello, excluye a las BMUs recién activadas de la competición.

Vector Quantized Temporal Associative Memory (VQTAM). Esta red (Barreto y Araujo, 2002) utiliza una línea de retardos y separa los vectores de entrada en dos partes: las muestras anteriores y su predicción. Es decir, en el caso de un espacio de entrada de una dimensión, se obtienen $\mathbf{x}^{in}(t) = [x(t), x(t-1), \ldots, x(t-p+1)]$ y $\mathbf{x}^{out} = x(t+1)$. La neurona ganadora se determina únicamente con la información de $\mathbf{x}^{in}(t)$. La regla de actualización tiene la misma forma que en el algoritmo tradicional, pero se calcula por separado para los pesos de entrada y de salida. De este modo, el SOM asocia los vectores prototipo de entrada con los correspondientes de salida.

SOMs de modelos autorregresivos. Esta familia de algoritmos hace uso de una línea de retardos a la entrada e identifica el modelo local más apropiado para cada región, a partir de los vectores prototipo.

En Vesanto (1997) se propone un algoritmo cuyo entrenamiento es similar al del VQTAM. Una vez entrenado, se realiza un particionado de la red y se estima un modelo local autorregresivo para cada neurona, por mínimos cuadrados, a partir de los vectores en su región de Voronoi, ampliada si su cardinalidad no es suficiente.

El modelo LLM, Local Linear Map (Walter *et al.*, 1990), utiliza la misma idea pero los cálculos de los modelos se realizan simultáneamente al entrenamiento.

Principe *et al.* (1998) proponen otro método en el que, tras entrenar el algoritmo como en el VQTAM, se ejecuta una fase de modelado donde cada predictor lineal $[\mathbf{a}_i^T, b_i]$ es estimado a partir de un conjunto de neuronas en la vecindad de la neurona u_i, incluyéndola a ella misma. Para esta red, asimismo, se ha propuesto una regla de aprendizaje modificada (aprendizaje dinámico), en la que se utiliza el error de predicción para determinar el parámetro de aprendizaje

$$\eta_\varepsilon = \frac{1 - \exp(-\mu(\eta + \overline{\varepsilon}))}{1 + \exp(-\mu(\eta + \overline{\varepsilon}))}. \tag{2.30}$$

Este planteamiento de múltiples modelos lineales se ha aplicado también a sistemas no autónomos (Cho *et al.*, 2006). En este caso, el SOM se entrena solamente con los datos de salida, para evitar un crecimiento excesivo del número de variables, que dificulta un modelado preciso, y la normalización de las variables

de entrada. Las entradas sí que se tienen en cuenta para el cálculo de los modelos lineales ARX (autorregresivos con entradas exógenas).

El método KSOM (Barreto *et al.*, 2004) recalcula un modelo lineal AR, cada instante de tiempo, a partir de los K vectores prototipo que mejor representan la entrada. Está orientado a los casos en que la preservación de la topología no sea muy buena.

SOM recursivos. La red **Recursive SOM** (Voegtlin, 2002) incrementa la dimensionalidad de cada neurona al añadir un vector de pesos de salida que se compara con las salidas reales del instante anterior. El error se calcula como

$$E_i(t) = \exp(-\alpha\|\mathbf{x}(t) - \mathbf{w}_i^{entrada}(t)\|^2 - \beta\|V(t-1) - \mathbf{w}_i^{salida}(t)\|^2) \tag{2.31}$$

donde el vector de contexto $V(t-1)$ se define del siguiente modo

$$V(t-1) = (\exp(-E_1(t-1)), \ldots, \exp(-E_n(t-1))). \tag{2.32}$$

El contexto preserva información sobre la activación en la red. Es superior al Recurrent SOM, especialmente cuando se necesita una visión global del espacio de entrada. Pero presenta un coste computacional alto. Existen otros modelos menos exigentes computacionalmente como el **SOMSD**, **SOM from structured data** (Hagenbuchner *et al.*, 2003), o el **Merge SOM** (Hammer *et al.*, 2004), que utilizan una representación del contexto más compacta. Este último presenta gran parecido en su planteamiento con el Recurrent SOM, pero su parametrización es más flexible.

Por otra parte, la red **Activation-based Recursive SOM** tiene en cuenta la activación de todo el mapa para procesar las entradas (Hynna y Kaipainen, 2006). Para ello, cada neurona se asocia con dos vectores de pesos que determinan la respuesta para cada entrada (contenido) y el estado de activación de la red (contexto). El vector de contexto tiene como dimensión el número de neuronas de la red y la activación se obtiene mediante un producto de las dos respuestas transformadas al rango $[0, 1]$.

SOM with Temporal Activity Diffusion. La arquitectura SOMTAD (Principe *et al.*, 2002) crea vecindades correlacionadas temporalmente en el espacio de salida, sin realizar cambios bruscos en su operación subyacente. Para ello, se generan ondas de actividad en las neuronas ganadoras, que decaen con el tiempo y que difunden información a través del espacio de salida.

Se determina cuál es la neurona ganadora combinando la regla estándar con la actividad de la neurona, de acuerdo a dos parámetros: el de realimentación, μ, y

el de acoplamiento espacio-temporal, β. Y se crean dos ondas, una que se mueve a través del espacio (μ^k) y otra a través del tiempo ($(1-\mu)^\tau$). El concepto de actividad dota al SOMTAD de capacidad anticipatoria, pues los umbrales de las neuronas vecinas bajan, aumentando su probabilidad de selección para la siguiente muestra.

2.6.3. Modelos locales

El **modelado de la dinámica** supone capturar el comportamiento a largo plazo del sistema, es decir, ajustar un modelo o conjunto de modelos para reconstruir su trayectoria en el espacio de fase, de tal modo que la salida autónoma de dicha estructura presente un comportamiento dinámico similar al original. Para llevar a cabo esta tarea, la mayoría de los modelos propuestos son globales (Gershenfeld y Weigend, 1994), cuyo resultado depende en gran medida de la representación funcional elegida. No obstante, el planteamiento de **agrupaciones de modelos locales** que modelen las regiones del espacio de estados de forma separada es apropiado para sistemas dinámicos complejos, pues ciertos sistemas dinámicos no lineales, especialmente aquellos de tipo caótico, presentan una estructura complicada y no uniforme que es difícil de aproximar mediante un modelo global. Si se utiliza la estrategia de múltiples modelos locales, cada submodelo describe separadamente cada comportamiento local mediante estructuras sencillas (generalmente lineales) y es la agrupación de estos submodelos la que da lugar a un modelo global no lineal con validez en un espacio de trabajo amplio (Narendra y Parthasarathy, 1990; Arahal *et al.*, 2009b). Es evidente que cada modelo local requiere menos suposiciones geométricas y estadísticas sobre los datos que las aproximaciones globales y, por ello, menores requisitos en términos de número de datos. De hecho, el grado de no linealidad del modelo conjunto puede controlarse por medio del número de submodelos que se utilizan para construirlo y, por tanto, se puede adaptar a la distribución de los datos (McNames, 1999). El diseño de un sistema de control a partir de este modelado también es más sencillo, pues se reduce a diseñar múltiples controladores locales más simples que se intercambian o cooperan (Erdogmus *et al.*, 2005). Esta estrategia es escalable, pues su complejidad se incrementa linealmente con el número de submodelos.

Bajo esta estrategia de descomposición, el comportamiento de los sistemas dinámicos no lineales puede ser descrito por medio de un conjunto de procesos locales autorregresivos (AR), autorregresivos de media móvil (ARMA) o con entradas exógenas en caso de no ser sistemas autónomos (ARX o ARMAX), cuyos coeficientes dependen del estado. Aunque se podrían utilizar polinomios de alto orden, los resultados no son mucho mejores (Farmer y Sidorowich, 1987) y se corre un riesgo mayor de *overfitting* (Vesanto, 2002). La descripción global de la

dinámica se compone mediante la unión de esos modelos locales.

Aunque las variables de estado no están accesibles normalmente, se pueden generar vectores de estado representativos mediante los valores pasados de las señales de entrada y salida para poder identificar el comportamiento del sistema. Este método de reconstrucción del vector de estado recibe el nombre de ***embedding* de retardos temporales**. En el caso de sistemas no lineales autónomos, existe una clara justificación teórica a este procedimiento, así como una orientación acerca de la longitud de la línea de retardos, por medio del teorema de *embedding* de Takens (1981). Supongamos un sistema dinámico estacionario y determinista del que podemos medir una cantidad escalar en cualquier momento. Es decir, existe una función de observación que proporciona una medida del estado actual del sistema $x_n = g(z_n)$. De acuerdo al teorema de Takens, en una situación libre de ruido, la evolución del estado actual en el tiempo

$$\mathbf{x}(t) = [x(t), x(t-\tau), \ldots, x(t-(d_e-1)\tau)], \tag{2.33}$$

donde τ es un retardo y d_e la dimensión de reconstrucción del espacio de estados, es igual a la descripción completa del sistema para una dimensión de reconstrucción $d_e \geq 2d_A$, donde d_A es la dimensión del atractor[5]. La reconstrucción da lugar a la trayectoria en el espacio de estados que, si se cumplen las condiciones anteriores, preserva los invariantes dinámicos y garantiza que no haya cruces entre trayectorias en el espacio de reconstrucción. Se han propuesto varios métodos para estimar d_e, como el análisis de falsos vecinos más cercanos o el índice de Lipschitz (Erdogmus *et al.*, 2005).

Por otra parte, aunque cualquier valor del retardo τ es aceptable, su elección es crítica para el resultado de la reconstrucción. Por eso es apropiado seleccionar un valor τ que separe los datos lo más posible. Existen varios criterios para estimarlo, uno de los cuales es asignarle un valor de la cuarta parte del período aproximado de la serie. Otros criterios se basan en información mutua y autocorrelación.

Para el caso de señales no autónomas, el proceso es prácticamente el mismo, salvo porque hay que tener en consideración las variables independientes de entrada. No obstante, si la inclusión de las variables de entrada aumenta mucho los grados de libertad, puede provocar problemas en el modelado.

En resumen, el procedimiento de modelado lineal local incluye la cuantización/*clustering* de los vectores de estado y la optimización de los coeficientes de los modelos locales correspondientes a cada vector, utilizando generalmente un criterio de mínimos cuadrados. La selección del vector más cercano de acuerdo a

[5]Se define atractor como el conjunto hacia el que un sistema dinámico evoluciona en el espacio de fase. Geométricamente puede ser, por ejemplo, un punto, un ciclo límite o un objeto geométricamente complicado (un atractor extraño).

alguna medida de distancia y la aplicación de su modelo asociado proporcionan la aproximación lineal por tramos a la dinámica no lineal global.

2.6.4. SOM y modelos locales

El mayor problema del método descrito en el apartado anterior emerge en la creación de la aproximación global del sistema por medio de la unión de los modelos locales. Por ello, las fronteras entre los modelos locales tienen que ser suaves, con cierto grado de solapamiento, pues, en caso contrario, el modelado de la dinámica fallará (Crutchfield y McNamara, 1987), ya que la mayor parte de los procesos físicos no exhiben cambios arbitrariamente rápidos en sus primeras derivadas.

Ese problema sugiere la creación de mapas auto-organizados del espacio de reconstrucción (Principe *et al.*, 1998), que presentan características muy interesantes como infraestructura para el modelado. En primer lugar, la cualidad de preservación de la topología garantiza la similitud de vectores vecinos y, por tanto, mitiga las discontinuidades. Además, al particionar el espacio, los mapas auto-organizados proporcionan una simplificación ordenada y discreta del espacio de entrada que permite aumentar la eficiencia y que sistematiza la selección de regiones de operación, que en este caso son las propias regiones de Voronoi de las neuronas del SOM (ver Fig. 2.8). Puesto que las regiones adyacentes en el espacio de entrada se proyectan como vecinos en el espacio de salida, también se reducen los errores causados por el ruido, ya que en caso de que existan desviaciones, se activarán neuronas similares (Principe *et al.*, 1998). Por otra parte, permite comprobar si el modelo está extrapolando (Bakker *et al.*, 2000), pues es importante permanecer dentro del contorno del atractor. Por todas esas razones, resulta muy útil en la reconstrucción de la dinámica global.

Para la construcción de modelos locales de señales autónomas haciendo uso del SOM, es necesario realizar un *embedding* de la serie temporal en un espacio de reconstrucción, aplicar el SOM sobre los pares $(y(n+1), \mathbf{x}(n))$, tal que $\mathbf{x}(n) = [x(n), x(n-1), \ldots, x(n-d)]$ y $y(n+1) = f_i(\mathbf{x}(n))$, y ajustar los coeficientes de los modelos locales lineales (Principe *et al.*, 1998) mediante mínimos cuadrados:

$$\mathbf{\Theta}_i = [\mathbf{a}_i^T, b_i]^T = (\mathbf{H}_i^T \mathbf{H}_i)^{-1} \mathbf{H}_i^T \mathbf{y}_i, \tag{2.34}$$

donde el vector $\mathbf{y}_i$ y la matriz $\mathbf{H}_i$ están formados, respectivamente, por los valores escalares $y(n+1)$ y los vectores $[\mathbf{x}(n), 1]$ de las neuronas vecinas a i (o de los datos en su región de Voronoi y adyacentes). Puede ser necesaria una regularización si $\mathbf{H}_i^T \mathbf{H}_i$ produce una matriz casi singular.

El proceso es el mismo para señales no autónomas pero, en lugar de trabajar con $\mathbf{x}(n)$, se utiliza $\psi(n) = [\psi_y(n), \psi_u(n)] = [y(n), \ldots, y(n-d_y), u(n), \ldots, u(n-d_u)]$.

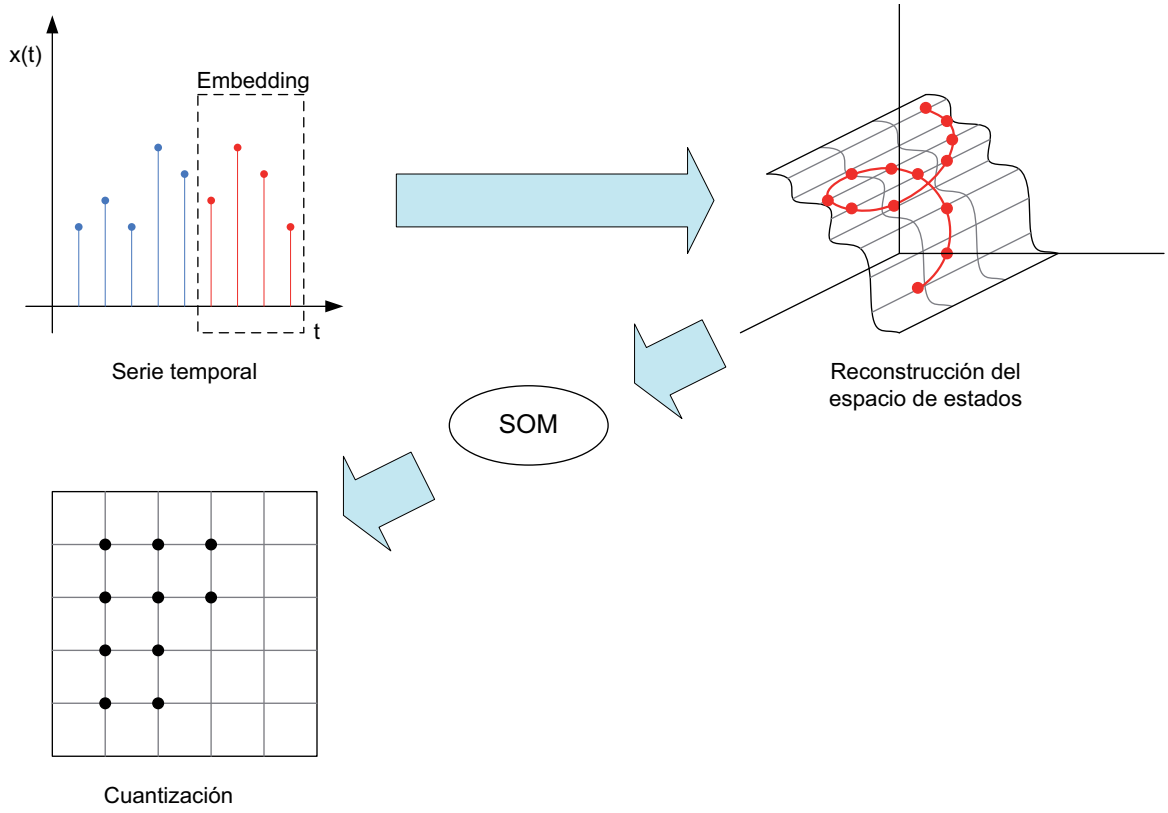


Figura 2.8. Proceso del modelado local con el SOM.

Finalmente, la unión de las predicciones locales lineales proporciona el modelado global de la dinámica en el espacio de estados

$$f \approx \bigcup_{i=1,\ldots,N} f_i. \tag{2.35}$$

2.7. Visualización de la dinámica

El comportamiento dinámico del sistema es un objetivo importante del análisis, para el cual sería útil contar con herramientas visuales como las presentadas en la sección 2.5. Sin embargo, la visualización de los aspectos dinámicos de los procesos industriales haciendo uso del SOM no ha sido objeto de estudio hasta muy recientemente (Díaz *et al.*, 2008), si exceptuamos las propuestas relacionadas con el tratamiento de la trayectoria (Simula y Kangas, 1995; Fuertes *et al.*, 2007) como la obtención y visualización de las condiciones de proceso, de las transiciones entre las mismas o la proyección del punto de operación.

No obstante, el SOM puede ser útil no solamente para su modelado con el objetivo de predicción o control, sino también para visualizar ciertos aspectos del comportamiento del proceso, aunque apenas haya sido aplicado. Es por ello que Díaz *et al.* (2008) propuso el análisis exploratorio de las características dinámicas de un modo exhaustivo, incluyendo relaciones entre variables y parámetros dinámicos.

Para este propósito se definen los **mapas de dinámica**, que permiten visualizar **características asociadas a modelos paramétricos** como, por ejemplo, las funciones de transferencia u otros modelos dinámicos no lineales. Estos mapas se construyen mediante una proyección entre un **espacio de entrada** que incluye los **parámetros dinámicos**, previamente conocidos o identificados, y las variables del proceso útiles para describir y diferenciar la dinámica local (**selectores de dinámica**) y el espacio de salida de baja dimensión. También se puede introducir el tiempo, si se pretenden analizar comportamientos no estacionarios. Así, una vez entrenado, el SOM asocia a cada coordenada de salida un vector prototipo con un conjunto de parámetros $\mathbf{p}_i$ de un modelo dinámico $\mathbf{y}(k) = f(\varphi(k), \mathbf{p}_i)$, que refleja el comportamiento local del proceso en ese estado, y un conjunto de variables, que distingue el punto de funcionamiento. Los mapas de dinámica presentan gran semejanza con los planos de componentes, pues se basan en asociar a cada neurona el valor de una característica, que en este caso es función del conjunto de parámetros $\mathbf{p}_i$. Los comportamientos similares se agrupan en regiones, lo que permite una representación ordenada de la dinámica del proceso.

Si se utilizan características o descriptores dinámicos con un sentido físico, los mapas de dinámica permiten informar cualitativamente de diversos aspectos de la dinámica del proceso y se establece una sinergia entre herramientas de visualización y conceptos de ingeniería de control que abre un amplio horizonte en la supervisión y el análisis exploratorio de la dinámica, pues facilita el análisis de diferentes modos dinámicos locales en procesos no lineales o no estacionarios, así como de los factores que influyen en los cambios.

Estos mapas puede ser visualizados e interpretados de forma consistente con las herramientas de visualización clásicas del SOM, pues todas ellas se definen a partir de los valores de los vectores prototipo y se visualizan sobre un malla 2-dimensional finita y fija. Es decir, las características estáticas y dinámicas del proceso se pueden combinar en una representación unificada que permite encontrar correlaciones entre las mismas.

Algunas visualizaciones propuestas hasta la fecha en este marco son los planos de componentes de los parámetros del proceso, los mapas de respuesta en frecuencia o los mapas de ganancia (Díaz *et al.*, 2008). No obstante, su estudio y aplicación han sido todavía muy limitados y se pueden diseñar nuevos mapas de visualización que sean útiles en el análisis y supervisión de procesos industriales.

Sería especialmente útil aplicar este procedimiento en tareas habituales de un ingeniero de control, como la obtención de la respuesta en el tiempo de un sistema o la comprensión del acoplamiento entre lazos presente en procesos multivariable.

El método también admite su uso conjunto con un algoritmo de modelado de la dinámica mediante el SOM (Díaz *et al.*, 2007), como los presentados en el apartado anterior. De esta manera se puede obtener la visualización de la dinámica a partir de los datos de entrada, sin tener que realizar una identificación previa de los parámetros dinámicos.

Capítulo 3

Definición del problema y metodología propuesta

3.1. Introducción

El problema de partida de esta tesis es la extracción y visualización de conocimiento acerca de la dinámica de procesos industriales por medio de mapas auto-organizados. Para ello, se realiza un proceso de reducción de la dimensión y cuantización de la dinámica del sistema mediante el que se asocia un modelo local a cada neurona del SOM. La naturaleza del espacio de salida, una malla discreta, permite que las propiedades relacionadas con esos modelos puedan ser visualizadas. Por eso, una de las vertientes del problema es aprovechar esa cualidad para definir mapas de visualización de características dinámicas que permitan al usuario extraer, de forma efectiva, conocimiento acerca del proceso. Este problema y la metodología propuesta para resolverlo se exponen de un modo más amplio en la sección 3.3.

Los modelos locales asociados a las neuronas pueden conocerse de antemano u obtenerse por medio de una identificación previa con cualquier algoritmo conocido. No obstante, si se aplica la técnica de modelado local al SOM, en la que los vectores construidos mediante *embedding* de retardos de la señal constituyen la entrada, es posible estimar los modelos a partir del mapa entrenado. Si este algoritmo se utiliza conjuntamente con los mapas de visualización, el SOM lleva a cabo de una forma integral el proceso que transforma los datos de entrada en información visualizable de la dinámica, lo que resulta muy conveniente. Aunque la efectividad del SOM como técnica de modelado ha sido comprobada ampliamente en la bibliografía (Principe *et al.*, 1998), existen ciertas modificaciones del SOM para dotarle de naturaleza espacio-temporal que podrían ser útiles para mejorar el

proceso de reconstrucción de la dinámica. Por eso, es interesante conocer que variaciones del algoritmo proporcionan mejores resultados de modelado o identificación de los procesos. Puesto que la evaluación de la reconstrucción de la dinámica no es un proceso trivial, se recurre a métodos previamente propuestos en la bibliografía pero que están limitados al modelado de señales autónomas. No obstante, se asume que las conclusiones que se extraigan de esta evaluación pueden extenderse a procesos con entradas y salidas, pues la existencia de entradas simplemente amplía la dimensión de entrada. El problema y la metodología se introducen en la sección 3.2.

3.2. Comparación de métodos de modelado local de la dinámica basados en SOM

3.2.1. Motivación

En el capítulo anterior, se presentó el interés de utilizar el mapa auto-organizado como infraestructura para el modelado de la dinámica mediante múltiples modelos locales. Como opciones para su implementación, nos encontramos con múltiples variantes orientadas a mejorar tanto el aprendizaje del espacio de estados de entrada como el cálculo de los modelos asociados a cada vector prototipo. Es necesario evaluar si alguna de ellas permite conseguir mejores modelados que el algoritmo clásico.

Puesto que las modificaciones espacio-temporales del SOM han sido definidas para mejorar su capacidad de procesar secuencias, la utilización de las mismas puede ser útil para mejorar la reconstrucción de la dinámica mediante modelos locales basada en SOM. De hecho, además de con el algoritmo clásico, se tiene constancia de la aplicación de esta misma estrategia con dos de esas variantes, DL-SOM (Principe *et al.*, 1998) y RSOM (Koskela *et al.*, 1998), si bien es cierto que el algoritmo equivalente al SOMTAD basado en Neural Gas también ha sido aplicado a la predicción de series temporales (Euliano, 1998) y los algoritmos recursivos han sido aplicados a su aprendizaje (Voegtlin, 2002).

Las comparaciones entre métodos son escasas en la bibliografía y, de hecho, prácticamente se limitan a las variantes recurrentes/recursivas (Hammer *et al.*, 2004; Strickert y Hammer, 2005). Sin embargo, la comparación que se propone permitiría alcanzar conclusiones globales que permitan contrastar el rendimiento de estos métodos, más ampliamente estudiados, con otros como los de reacción-difusión.

Se podría argumentar que el uso de mapas recursivos o de reacción-difusión

para el modelado local es algo innecesario, pues el *embedding* ya introduce, de forma externa, la memoria a corto plazo para trabajar con información espacio-temporal. No obstante, cabe recordar que el *embedding* produce una **trayectoria en el espacio de reconstrucción** que, a su vez, puede presentar cierta complejidad. El algoritmo SOM tradicional determina la región correspondiente de acuerdo a la posición estática del punto de la trayectoria. Sin embargo, las variantes espacio-temporales pueden utilizar información de contexto (posiciones pasadas en la trayectoria) para determinar esta región, lo que puede ser útil para mitigar las desviaciones en la trayectoria debidas al ruido, que ante sistemas caóticos provocan una divergencia muy rápida. Por esa razón, **se sostiene que la información sobre el contexto implícita en estos mapas puede mejorar el modelado de la dinámica**.

Por otra parte, se presenta la posibilidad de estimar los coeficientes de los modelos a partir de los datos o directamente a partir de los pesos y es necesario conocer si, en el segundo caso, se sacrifica eficacia por rapidez de ejecución.

El interés de contar con una evaluación de estos métodos, en términos de su capacidad de modelado, es múltiple: permite determinar que estrategias son las más adecuadas, optimizar los resultados en su aplicación y orientar la investigación hacia aquellos más adecuados.

Una aplicación directa de la técnica de modelado es la predicción. Pero la predicción, además de ser una aplicación del modelado, también constituye un método apropiado para validarlo, no sólo por medio del error, sino también mediante el uso de la predicción iterada como un método para reconstruir la señal y, posteriormente, compararla con la original utilizando otros criterios.

3.2.2. Evaluación de la predicción frente a evaluación del modelado

El problema de reconstrucción de la dinámica puede definirse como aquel que permite representar la dinámica de la señal correctamente y, al mismo tiempo, identificar un predictor (Principe *et al.*, 1992). Es necesario contar con criterios que determinen si estas condiciones se cumplen. En predicción lineal, la minimización del error cuadrático medio produce un modelo preciso de la dinámica; en sistemas no lineales, sin embargo, esta no es una medida consistente para determinarlo. El problema radica en que dos trayectorias en el mismo atractor pueden ser muy diferentes muestra a muestra y dos puntos muy cercanos pueden corresponder a trayectorias diferentes. Esto ocurre en ciertos sistemas no lineales relativamente comunes que se denominan caóticos (Haykin y Principe, 1998; Bakker *et al.*, 2000). En estos sistemas, su comportamiento "extraño" en el tiempo genera geometrías

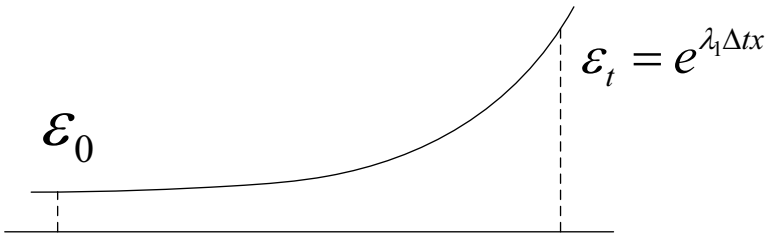


Figura 3.1. Divergencia de trayectorias causada por un exponente de Lyapunov positivo.

complicadas en el espacio de fase. Es decir, aunque la minimización del error puede ser una condición necesaria para un buen modelado, no es suficiente.

El error de predicción medio a largo plazo (calculado tras múltiples pasos), aunque útil, tampoco es válido como única métrica de evaluación, pues continúa siendo una comparación muestra a muestra que solamente refleja una aproximación local del modelo. No es capaz de separar el efecto de la imperfección del modelo entrenado y la incertidumbre causada por el propio sistema. No obstante, su minimización continúa siendo una condición importante que puede utilizarse para dirigir la selección de parámetros, como en la validación cruzada multipaso (McNames, 2002) y como objetivo del entrenamiento.

Será necesario describir, en primer lugar, el tipo de sistemas no lineales que, por su dificultad, marcan la pauta del procedimiento que debe ser seguido para cuantificar la calidad del modelado y se constituyen como los más apropiados para las pruebas experimentales. La **dinámica** de los **sistemas caóticos** es determinista, acotada y aperiódica (Small, 2005) y se asocia a la respuesta autónoma de ciertos sistemas dinámicos no lineales que, sin ninguna entrada aleatoria, presentan una salida con un comportamiento complejo y con un espectro similar al ruido. Es decir, la incertidumbre de estos sistemas procede de su propia dinámica, por lo que las técnicas estocásticas tradicionales no pueden reproducir su estructura correctamente.

Un sistema dinámico caótico tiene, al menos, un **exponente de Lyapunov** positivo. Los exponentes de Lyapunov caracterizan la divergencia de las trayectorias en el espacio de fase causada por la dinámica del sistema (Small, 2005) y un exponente de Lyapunov positivo implica que las trayectorias divergen de forma exponencial (ver Fig. 3.1). Esto dota al sistema de tal sensibilidad a las condiciones iniciales que el comportamiento dinámico es impredecible a largo plazo. Debido a que las predicciones son imperfectas, las señales caóticas divergen de su predicción de forma inexorable y el error a largo plazo es irreducible.

En Haykin y Principe (1998) se propone generar otra serie temporal, a partir

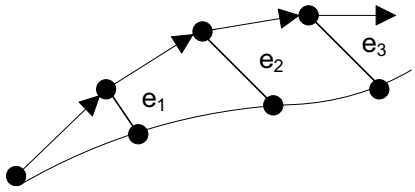


Figura 3.2. Predicción iterada.

del modelo, y compararla con la original con respecto a medidas de la dinámica que sean **invariantes** a su reconstrucción. Para reconstruir la serie de un modo similar a cómo se genera la serie original, se propone realizar una predicción iterada, alimentando la salida predicha $\hat{\mathbf{x}}(t+1)$ a la entrada con cada estimación, como si de un sistema autónomo se tratase. Si el modelado es correcto, la dinámica de la serie temporal reconstruida debe ser consistente con la de la serie original.

Con la estrategia anterior, se puede determinar si el modelo representa la dinámica correctamente; pero, para considerar que la reconstrucción de la dinámica es satisfactoria, también es importante la minimización del **error de predicción a largo plazo** de la serie hasta un horizonte de predicción. El cálculo de este error se puede realizar de forma directa o iterada. La predicción directa n pasos adelante es cuestionable, pues una función que realice esa predicción será, generalmente, mucho más complicada de modelar que otra que simplemente realice una predicción a un paso (McNames, 1999). No obstante, la predicción iterada (ver Fig. 3.2) presenta el problema de ignorar los errores acumulados en el vector de entrada. La mayor parte de los autores (Casdagli, 1989; McNames, 2002) trabajan con esta última. Ante sistemas caóticos, el error de predicción a largo plazo solamente se puede mantener controlado hasta que se alcanza el horizonte de predicción, a partir del cual las trayectorias diverge irremisiblemente y es necesario reiniciar el procedimiento.

3.2.3. Invariantes dinámicos y su estimación

Los exponentes de Lyapunov son **invariantes dinámicos**, que se definen como cantidades que describen el comportamiento dinámico de un sistema y no dependen del sistema de coordenadas (Small, 2005). Es decir, aquellas en las que el valor en la reconstrucción sería igual que el del sistema original si tanto la reconstrucción como la estimación fueran perfectas.

Además de los exponentes de Lyapunov, otro de los invariantes dinámicos más ampliamente utilizados es la **dimensión de correlación**, d_2, que es una extensión de la noción tradicional de dimensión al caso fraccional y una medida de la

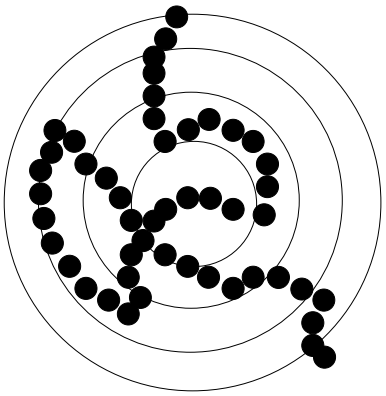


Figura 3.3. Estimación de la dimensión de correlación.

complejidad del atractor. Aunque no hay una equivalencia entre sistemas caóticos y sistemas con atractor fractal, esta dimensión es un indicador útil, pues permite descartar que el sistema dinámico esté dominado por el ruido, que tiene una dimensión de correlación infinita, o por una órbita periódica, que tiene una dimensión de correlación entera (Small, 2005). No obstante, para la aplicación que nos ocupa, solamente se busca que la estimación de la dimensión de correlación y del mayor exponente de Lyapunov del sistema real y del reconstruido se aproximen.

Es decir, la estimación de estos invariantes dinámicos es necesaria para validar el modelo (Principe *et al.*, 1992). Existen diversos algoritmos para estimar a partir de los datos ambos invariantes (Small, 2005). El algoritmo clásico para la estimación de la dimensión de correlación es el propuesto por Grassberger y Procaccia (1983), que utiliza la integral de correlación $C(\varepsilon)$, proporcional al número medio de pares de puntos contenidos en una esferoide de radio ε (ver Fig. 3.3). La dimensión de correlación se calcula por medio del gradiente de $\ln C(\varepsilon)$ frente a $\ln \varepsilon$ cuando ε tiende a 0. El problema de este algoritmo es que es muy fácil malinterpretar los resultados. Por eso, se han propuesto otros algoritmos mejorados para esta tarea (Small, 2005), como el algoritmo de Judd, que con un punto de partida similar propone un procedimiento más elaborado, y el de kernel gaussiano (GKA), que permite calcular simultáneamente la dimensión de correlación, la entropía de correlación y el nivel de ruido. Por otra parte, el primer algoritmo propuesto para estimar el mayor exponente de Lyapunov es el propuesto por Wolf *et al.* (1985) que, no obstante, presenta la desventaja de que supone la existencia de divergencia exponencial y puede proporcionar resultados engañosos. El algoritmo propuesto independientemente por Rosenstein *et al.* (1993) y Kantz (1994) calcula directamente la divergencia exponencial, por lo que permite decidir si tiene sentido el cálculo del exponente y es más apropiado.

Existen otros invariantes dinámicos, que potencialmente podrían utilizarse para evaluar la reconstrucción de la dinámica, como el espectro de los exponentes de Lyapunov, otras generalizaciones del concepto de dimensión o la entropía de Kolmogorov-Sinai (Kantz y Schreiber, 2003), las cuales están interrelacionadas entre sí.

3.2.4. Métodos objeto de comparación

Como se introdujo al comienzo de este capítulo, es interesante evaluar si las diferentes propuestas de modificación del SOM mejoran la reconstrucción de la dinámica. A continuación se presentan las diversas estrategias relevantes para este propósito, así como el algoritmo más representativo de cada una, que será el que se considerará para su comparación.

Modificación de la regla de aprendizaje. De entre las técnicas que introducen modificaciones en la regla de aprendizaje del algoritmo SOM, resulta de especial relevancia la aproximación de **aprendizaje dinámico** o SOM-DL (Principe *et al.*, 1998), cuyo objetivo es, precisamente, mejorar el rendimiento del algoritmo en tareas de modelado dinámico. Para conseguir este objetivo, incorpora el error de predicción en el entrenamiento de la red, de tal manera que las partes difíciles de modelar ocupen mayores regiones en el espacio de salida. Eso se consigue mediante la modificación de la tasa de aprendizaje de acuerdo a la siguiente expresión

$$\eta_\varepsilon = \frac{1 - \exp(-\mu(\eta + \overline{\varepsilon}))}{1 + \exp(-\mu(\eta + \overline{\varepsilon}))}, \tag{3.1}$$

donde $\overline{\varepsilon}$ es el error de predicción normalizado

$$\overline{\varepsilon} = \frac{|x(n+1) - (\mathbf{a}_{i^*}^T \mathbf{x}(n) + b_{i^*})|}{|x(n+1)|}. \tag{3.2}$$

Es decir, este algoritmo cierra el lazo para ajustar el aprendizaje al error obtenido en cada paso. Aunque es más costoso computacionalmente, puesto que es necesario estimar los modelos y calcular el error de predicción durante el entrenamiento de la red, ha proporcionado mejores resultados que el algoritmo clásico en tareas de modelado local de la dinámica (Principe *et al.*, 1998).

Modificación de la regla de activación. Siguiendo esta aproximación, cabe destacar la red **SARDNET** (James y Miikkulainen, 1995), que excluye las neuronas recientemente activadas de entre las seleccionables por la regla de activación, forzando a que una secuencia de datos de entrada active una secuencia de BMUs

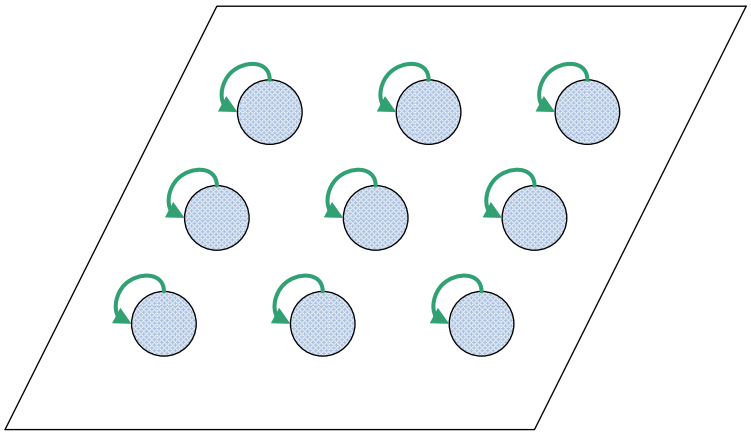


Figura 3.4. SOM recurrente.

diferentes, de tal manera que pequeñas variaciones de esas secuencias se aprendan con mayor detalle. La longitud de la secuencia depende del valor de decremento d, ya que las activaciones se actualizan en cada iteración mediante

$$\eta_i(t+1) = d\eta_i(t), \tag{3.3}$$

donde $0 < d < 1$. Esta red crea representaciones muy densas pero descriptivas. Ha sido aplicada al reconocimiento de secuencias con éxito pero no se tiene constancia de su uso para la reconstrucción de la dinámica.

Recurrencia. De entre las redes que introducen recurrencia en sus neuronas se selecciona el método **SOM Recurrente** (ver sección 2.6.2), pues se obtienen mejores resultados que con otras variantes recurrentes como el TKM. Se utiliza un integrador *leaky* a la entrada (ver Fig. 3.4), de tal manera que la activación se define como

$$\mathbf{y}_i(t) = (1-\alpha)\mathbf{y}_i(t-1) + \alpha(\mathbf{x}(t) - \mathbf{w}_i(t)), \tag{3.4}$$

donde α determina la profundidad de la memoria a corto plazo.

Para entrenar esta red, se presentan a la entrada episodios de muestras consecutivas cuya longitud depende del coeficiente de *leaking* y, al final de cada episodio, se determina la neurona ganadora por medio de la distancia al vector $\mathbf{y}$. Los pesos de las neuronas se actualizan mediante una variación de la regla tradicional en la que $(\mathbf{x}_t - \mathbf{w}_{i(t)})$ es sustituido por $\mathbf{y}_t$. En cada episodio, $\mathbf{y}$ es inicializado a cero. Este algoritmo ya se ha aplicado a la predicción de series temporales basada en el modelado local de la dinámica (Koskela *et al.*, 1998), con la justificación, expresada al principio de este capítulo, de que su recurrencia puede ayudar a capturar la

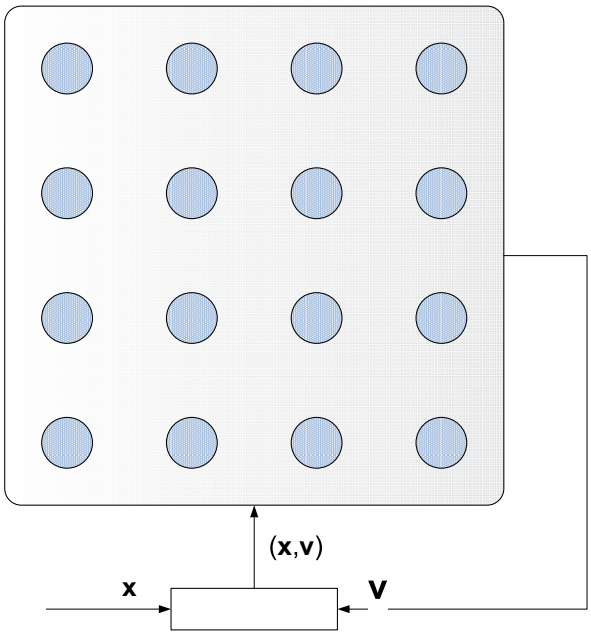


Figura 3.5. SOM recursivo.

información de contexto que se haya perdido durante el *embedding*.

Recursividad. Las modificaciones recursivas del algoritmo también implementan recurrencia global pero, en este caso, de un modo que afecta a todo el sistema, tal y como se esquematiza en la Fig. 3.5. Siguiendo esta aproximación, se han descrito varios algoritmos, los cuales han sido comparados en la bibliografía (Hammer *et al.*, 2004). De entre los mismos, cabe destacar el algoritmo ***Merge* SOM** (Strickert y Hammer, 2005), que presenta resultados equivalentes a otros algoritmos de este tipo que son bastante más complejos o modifican la malla neuronal. Este algoritmo permite almacenar información acerca de la neurona ganadora en pasos anteriores por medio de la información de contexto, que se integra mediante unos vectores adicionales $\mathbf{c}_i$ asociados a cada neurona. Como BMU se selecciona aquella que minimiza la distancia recursiva

$$d_i(t) = (1-\alpha)\|\mathbf{x}(t) - \mathbf{w}_i\|^2 + \alpha\|\mathbf{c}(t) - \mathbf{c}_i\|^2, \tag{3.5}$$

donde el descriptor de contexto $\mathbf{c}(t)$ es la combinación lineal de las propiedades del ganador anterior i^*_{t-1}

$$\mathbf{c}(t) = (1-\beta)\mathbf{w}_{i^*_{t-1}} + \beta\mathbf{c}_{i^*_{t-1}}. \tag{3.6}$$

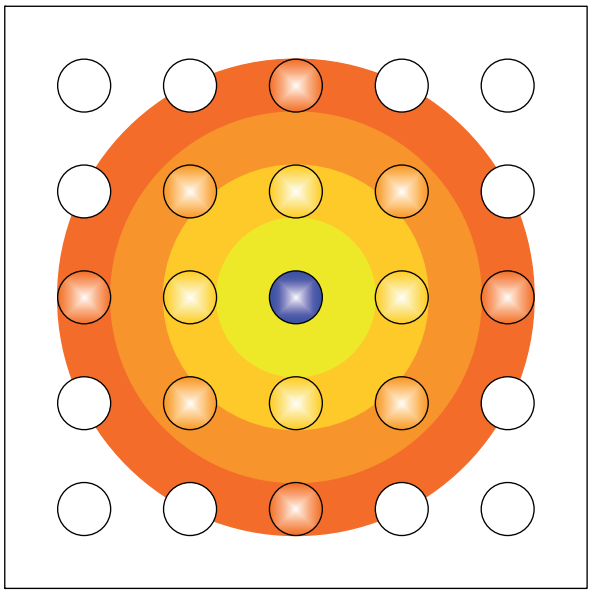


Figura 3.6. Difusión de la actividad.

La regla de aprendizaje es la estándar para los pesos, pero también se actualizan los $\mathbf{c}_i$ de la siguiente manera:

$$\Delta \mathbf{c}_i = \eta \cdot h_{ii^*} \cdot (\mathbf{c}(t) - \mathbf{c}_i). \tag{3.7}$$

En (Hammer *et al.*, 2004), ha sido evaluado su error de cuantización a lo largo del tiempo ante series temporales caóticas, aunque no se tiene constancia de ninguna aplicación en la que el aprendizaje de la dinámica se haya utilizado para su posterior reconstrucción y predicción.

Difusión de la actividad. En la sección 2.6.2 se presenta el **SOMTAD** (Euliano, 1998; Principe *et al.*, 2002) como una red que establece la correlación temporal deseada por medio de la difusión de la actividad de las neuronas ganadoras tanto en el espacio como en el tiempo. Sin necesidad de cambios bruscos en la operación subyacente y sin aumentar considerablemente la complejidad del algoritmo, modifica el umbral de activación de las neuronas por medio de ondas de actividad que decaen con el tiempo. De este modo, produce un mayor error de cuantización que el SOM convencional pero, a cambio, consigue mejores resultados en condiciones de ruido u ordenación temporal.

Aunque los autores (Principe *et al.*, 2002) utilizan la idea 2-dimensional de "ondas generadas en una charca" como metáfora intuitiva de la actividad de salida de la red (Fig. 3.6), el SOMTAD presentado se formula en 1 dimensión. No obstante,

establecen las líneas generales para aplicarlo a 2 dimensiones y utilizar esta estrategia para extender el algoritmo *Neural Gas*. Aunque esta extensión (GASTAD) es, en efecto, apropiada para problemas más complejos, sigue siendo conveniente contar con un SOMTAD 2-D para ciertos problemas como el que nos ocupa.

Por esa razón, se propone aquí un algoritmo SOMTAD de dos dimensiones, consistente con la estrategia 1-D, que respeta las reglas competitiva y de actividad originales y cuyo único cambio radica en el modo de definir la mejora de cada neurona. De este modo, se define la regla de selección de BMU como

$$i^* = \arg\min_i(d(\mathbf{w}_i, \mathbf{x}(t)) - \beta e_i(t)), \tag{3.8}$$

donde la distancia es, como de costumbre, $d(\mathbf{w}_i, \mathbf{x}) = \|\mathbf{x} - \mathbf{w}_i\|$. La mejora de cada neurona, incluyendo factores de normalización, se define de la forma siguiente:

$$e_i(t) = \frac{\mu e_{ady(i)}(t-1) + a_i(t)}{1+\mu}, \tag{3.9}$$

donde $a_i(t)$ indica la activación, que a su vez se calcula de la siguiente manera

$$a_i(t) = \frac{(1-\mu)a_i(t-1) + d(\mathbf{w}_i, \mathbf{x})}{2-\mu}, \tag{3.10}$$

y $e_{ady(i)}(t)$ es la mejora propagada a la neurona desde las neuronas adyacentes. Aquí radica la principal diferencia con el algoritmo presentado en (Principe *et al.*, 2002), ya que en ese caso solamente se propaga la mejora de la neurona anterior en la línea 1-dimensional. Para el caso 2D, se propone la siguiente ecuación de mejora:

$$e_{ady(i)}(t) = \max_{k \epsilon Ady(i)} e_k(t), \tag{3.11}$$

que, amortiguada por μ, provoca el efecto de onda propagada por el medio de un modo sencillo. Para evitar efectos no deseados en los extremos, es conveniente utilizar un mapa toroidal. El algoritmo SOMTAD 2-D, con las consideraciones que acaban de presentarse, es el método basado en difusión de la actividad que se utiliza para la comparación.

3.2.5. Ajustes objeto de comparación

También es conveniente conocer si la selección de método para calcular los coeficientes de los modelos es clave en el éxito de la reconstrucción. Por eso, es necesario evaluar los dos métodos de ajuste más utilizados.

Ajuste de los modelos locales a partir de los datos. Una vez entrenada la red, es necesario ajustar los coeficientes del modelo local, para lo cual existen básicamente dos posibilidades. Mediante la primera de ellas, la estimación de los coeficientes del modelo local se realiza utilizando los vectores de datos contenidos en la región de Voronoi de la neurona ganadora[1] (Vesanto, 1997; Cho *et al.*, 2006). En el caso en que los datos asignados a la región de Voronoi sean menos que la dimensión de entrada, se utilizan también los vectores de neuronas vecinas (Cho *et al.*, 2006), aunque el conjunto de estimación podría, alternativamente, ampliarse con los datos de la región de Voronoi de la segunda BMU y sucesivas, como propone Vesanto (1997). El mayor rendimiento de una estrategia u otra dependerá de la calidad de la preservación de la topología (Barreto, 2007). El ajuste de modelos locales de acuerdo a las particiones generadas por el mapa auto-organizado es una estrategia encaminada a hacer a los modelos locales más robustos al ruido y los *outliers* y, por tanto, mejorar la generalización de la red (Cho *et al.*, 2006).

Ajuste de los modelos locales a partir de los pesos. La otra opción para estimar los coeficientes es utilizar los propios pesos de las neuronas (Principe *et al.*, 1998), lo que permite un cálculo mucho más eficiente. No obstante, es necesario utilizar un número de neuronas apropiado para poder realizar la estimación de mínimos cuadrados. Esto puede implicar que, para estimar modelos de alto orden, deban considerarse vectores prototipo muy alejados del vector de entrada. Es necesario determinar cuál de las estrategias de ajuste es la más apropiada, si existen diferencias, o distinguir en qué condiciones es más conveniente una u otra.

3.3. Visualización de la dinámica

3.3.1. Motivación

Como se ha repetido a lo largo de este texto, el objetivo de esta tesis es analizar aspectos relacionados con la extracción de conocimiento de procesos industriales. En este ámbito, uno de los aspectos más importantes para su análisis y supervisión es el referente a su **comportamiento dinámico**, ya que es el primer paso para el diseño y optimización del control del sistema. Con este mismo objetivo y basándose en las aplicaciones del SOM tanto para el modelado local de la dinámica

[1]Es necesario aclarar que, en los casos en que se trabaje con algoritmos que introducen algún cambio en la selección de la BMU (SARDNET, RSOM, MSOM, SOMTAD), la asignación de datos de entrada a las diferentes regiones de Voronoi para la estimación de los modelos se realizará de forma consistente con el entrenamiento, contando con información de contexto si el algoritmo de entrenamiento la utilizase.

(Principe *et al.*, 1998) como para la visualización de otros aspectos de los procesos industriales (Alhoniemi *et al.*, 1999), Díaz *et al.* (2008) definió los **mapas de dinámica**. Estos mapas permiten representar, de forma ordenada, los distintos comportamientos locales para los puntos de funcionamiento del proceso y mantienen la consistencia con la representación de variables del proceso basada en planos de componentes. Por ello, permiten explorar características de la dinámica del proceso de manera visual y encontrar relaciones entre las mismas y con respecto a las variables del proceso, sobre una representación unificada.

Como ejemplo de la utilidad de esta estrategia, Díaz *et al.* (2008) presentan mapas de visualización de características de la respuesta en frecuencia que permiten representar la ganancia a una determinada frecuencia, la frecuencia de resonancia o el ancho de banda. No obstante, la aplicación de esta estrategia tiene un horizonte mucho más amplio, ya que las características dinámicas con significado físico útil para los ingenieros de control son generalmente función de los parámetros que definen los modelos locales y el método de visualización de los **mapas de dinámica** permite **visualizar características** asociadas a **cualquier modelo paramétrico**. Por esta razón, el objetivo de visualización de esta tesis es la definición de nuevos mapas interpretables de características dinámicas aplicados a conceptos habituales del ámbito industrial, como la respuesta temporal de procesos monovariables o el acoplamiento y direccionalidad de los procesos multivariable. Para estos procesos, muy estudiados, existe un amplio conjunto de medidas y procedimientos que guían el análisis. No obstante, la visualización de esos conceptos clásicos de ingeniería de control mediante los mapas de dinámica propuestos permite un estudio más completo, que revele el comportamiento del proceso en todo el rango de funcionamiento y que, por su representación gráfica unificada y ordenada, facilite la búsqueda de correlaciones con otras características y con las variables del sistema.

3.3.2. Definición de mapas de dinámica

El objetivo de los mapas de dinámica es la obtención de representaciones visualizables y ordenadas de modelos paramétricos por medio de una reducción de la dimensión de los parámetros, los cuales se consideran conocidos de antemano u obtenidos mediante un proceso de identificación previo.

El conjunto de datos de entrada se subdivide en N subconjuntos, a cada cual se asocia la dinámica por medio de uno de los vector de parámetros $\mathbf{p}(i)$. En función del modelo paramétrico utilizado, las componentes de $\mathbf{p}$ pueden ser, por ejemplo, los coeficientes de la función de transferencia de un sistema lineal invariante en el tiempo o de un modelo no lineal de tipo NARX (procesos autorregresivos no lineales con variables exógenas).

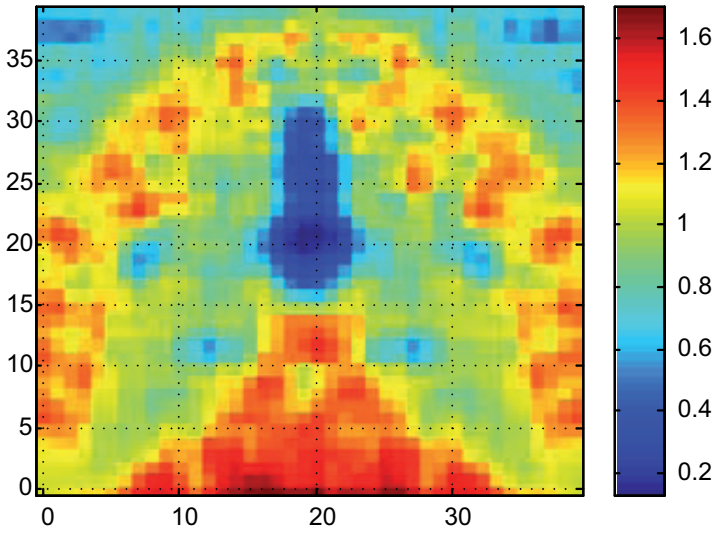


Figura 3.7. Ejemplo de mapa de dinámica de característica escalar.

Tras este proceso, se crea un espacio de entrada multidimensional que combina esos parámetros y las variables del proceso que cuentan con cierta capacidad para discriminar la dinámica local, es decir, aquellas que determinan diferentes comportamientos dinámicos para diferentes puntos de funcionamiento. Estas variables, seleccionadas por medio de conocimiento previo o tras una extracción de características, reciben el nombre de **selectores de la dinámica** (Díaz *et al.*, 2008) y permiten enlazar las visualizaciones de la dinámica con las visualizaciones del estado.

Es decir, sea $\mathbf{p}_i$ un conjunto de parámetros que define un modelo paramétrico $y(k) = f(\phi(k), \mathbf{p}_i)$ asociado a un punto de funcionamiento y $\mathbf{x}_i$ el conjunto de variables selectoras de la dinámica. El espacio de entrada de entrada al mapa auto-organizado se compone de vectores $\mathbf{q}$ que se definen como

$$\mathbf{q} = [\mathbf{p}^T, \mathbf{x}^T, \mathbf{t}]^T, \tag{3.12}$$

donde el tiempo t puede añadirse opcionalmente en el caso de comportamientos no estacionarios. La proyección de estos datos por medio del SOM produce una representación conjunta y consistente del punto de operación y su dinámica, en la que comportamientos similares se agrupan en regiones, lo que permite una representación ordenada de la dinámica del proceso.

Tras el entrenamiento de la red, cada neurona se encuentra representada por el vector prototipo $\mathbf{m}_i$, de la misma dimensión de entrada, y la posición $\mathbf{g}_i$ en el espacio de salida. Denominemos k a la característica escalar a representar en el

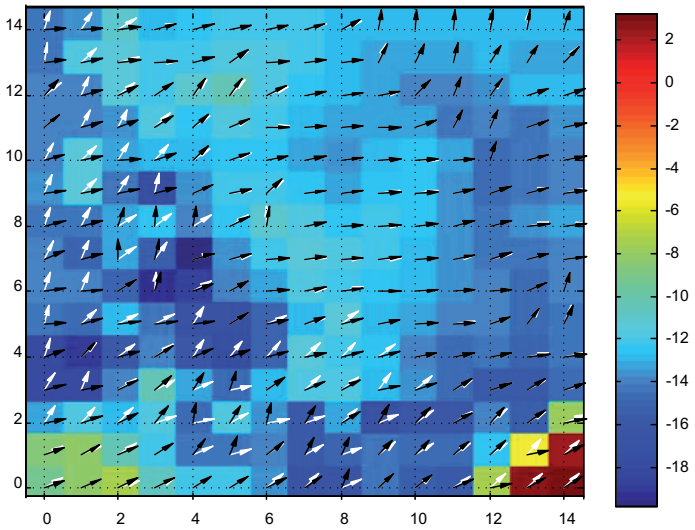


Figura 3.8. Ejemplo de mapa de dinámica de característica vectorial.

mapa de dinámica. Esta característica de la dinámica depende, de algún modo, de los parámetros $\mathbf{p}$ del modelo,

$$k = g(\mathbf{p}), \tag{3.13}$$

los cuales están disponibles para cada neurona de la red, es decir, para cada punto de operación. Por tanto, en el caso escalar existe una manera directa de representar la característica k para cada punto de operación: asociar a la posición $\mathbf{g}_i$ un código de color que depende del valor de k_{p_i}, de igual manera que en otros mapas de visualización clásicos de los mapas auto-organizados, como el plano de componentes (ver Fig. 3.7). Para los casos en que se trabaje con una propiedad $\mathbf{k}$, también función de $\mathbf{p}$, que tenga una naturaleza vectorial, existen diversas opciones de representación. Una de ellas es representar cada una de sus componentes mediante código de color, lo que será apropiado para vectores de 3 o más dimensiones. Los vectores de 2 dimensiones, no obstante, pueden ser representados como tal en cada punto del mapa, sin aumentar drásticamente la complejidad de la representación y sin perder intuitividad (ver Fig. 3.8).

La posibilidad de visualizar la relación entre un punto de funcionamiento del proceso y el comportamiento dinámico del mismo abre vías importantes de análisis de la dinámica para encontrar y clasificar modos dinámicos y estudiar la influencia de los factores que producen cambios en el comportamiento. Además, el uso de mapas que representan la información de un modo intuitivo y estandarizado y para los que cada punto de trabajo se representa siempre en la misma posición es una ventaja. El analista o supervisor puede interpretarlos sin tener que

aprender complejos sistemas de representación, solamente recurriendo a sus conocimientos de ingeniería de control. Además, puede descubrir visualmente, mediante comparación, regiones y correlaciones entre características y asociar estados con comportamientos.

En cuanto al entrenamiento del mapa auto-organizado, es conveniente trabajar con un conjunto de datos de entrada normalizado. Si esto es recomendable en general, en este caso lo es aún más, pues el objetivo es que los vectores *codebook* se parezcan lo más posible a los datos originales, pues se tiene más interés en la preservación de su topología (bajo error de cuantización). Un gran error en una componente con pequeña magnitud podría provocar cambios sustanciales, especialmente si esa componente corresponde a uno de los parámetros/coeficientes dinámicos (y más si se trata de sistemas físicos muestreados, mucho más sensibles a este tipo de errores). Por esa razón, en este aplicación del SOM no conviene que ninguna variable, debido a su magnitud, domine el proceso.

Por otra parte, al comienzo de este capítulo se encuadraron tanto la utilización del SOM para modelado como su aplicación para visualización dentro del marco de extracción del conocimiento de la dinámica. En efecto, ambos procedimientos se pueden aplicar de forma simultánea y, aunque la obtención de los modelos locales asociados a cada neurona se ha descrito como un paso independiente de esta tarea, también se puede utilizar el enfoque de múltiples modelos locales lineales (Díaz *et al.*, 2007) para la obtención de los parámetros dinámicos. De este modo, la identificación se integra en el proceso y los datos de entrada a la red neuronal son únicamente las muestras de las variables. El único procedimiento que ha sido propuesto hasta el momento (Díaz *et al.*, 2007) se basa en el algoritmo propuesto por Cho *et al.* (2006), con la diferencia de que el SOM no se auto-organiza de acuerdo a los vectores de retardos sino a los selectores de dinámica. Una vez entrenado el SOM, los modelos locales asociados a cada neurona se calculan de forma equivalente pero, debido al uso de las variables selectoras, continúa siendo posible conectar el comportamiento dinámico con el punto de funcionamiento. Pese a que el uso conjunto del SOM para modelado y visualización no es objeto de estudio en esta tesis, sí que constituye una interesante línea futura de investigación, pues es necesario identificar los procedimientos y algoritmos más adecuados para su utilización.

3.3.3. Mapas de visualización de sistemas SISO

Existen una serie de características de la respuesta de un sistema que permiten analizar de una forma efectiva su comportamiento y que, por su uso común, son fácilmente interpretables por cualquier ingeniero de control (Ogata, 2001). Estas características revelan aspectos importantes de la respuesta transitoria y perma-

nente en el tiempo a diferentes excitaciones de entrada. Por ello, se considera apropiado definir mapas de visualización que permitan evaluar esas características de la respuesta de un modo global, en todo el rango de funcionamiento y con la posibilidad de comparar y relacionar esa información.

De entre las posibles características, se han escogido aquellas más habituales en el estudio de sistemas de segundo orden, pues muchos sistemas importantes exhiben ese comportamiento y otros sistemas de orden superior se pueden aproximar a ellos para analizarlos. En este apartado se repasan brevemente algunas de esas características y se presentan los mapas de visualización correspondientes a las mismas, generadas mediante el procedimiento descrito en el apartado anterior.

Los sistemas de segundo orden se pueden expresar con la siguiente función de transferencia

$$G(s) = \frac{K\omega_n^2}{s^2 + 2 \cdot \zeta \cdot \omega_n \cdot s + \omega_n^2}, \tag{3.14}$$

donde K es la **ganancia**, ζ es el **coeficiente de amortiguamiento** y ω_n la **frecuencia natural no amortiguada**. Estos parámetros definen el comportamiento dinámico del sistema, que depende de las características de las raíces del polinomio característico (denominador), que son los **polos** de la función de transferencia. Estas raíces son de la forma

$$s_1 = -\sigma + j\omega_d \tag{3.15}$$

$$s_2 = -\sigma - j\omega_d, \tag{3.16}$$

donde $\sigma = \zeta \cdot \omega_n$ es la **atenuación** o constante de amortiguamiento y $\omega_d = \omega_n \cdot \sqrt{1 - \zeta^2}$ es la **frecuencia natural amortiguada** o frecuencia forzada.

El valor de ζ define la naturaleza de los polos:

- Si $\zeta = 0$, las raíces son imaginarias conjugadas y el sistema es un sistema sin amortiguamiento, cuya respuesta a un impulso corresponde en el tiempo a una senoide cuya amplitud y frecuencia es la frecuencia natural del sistema.

- Si $0 < \zeta < 1$, las raíces son complejas conjugadas y su respuesta a un impulso en el tiempo es una señal senoidal amortiguada de frecuencia ω_d.

- Si $\zeta = 1$, hay una raíz real doble y el sistema es críticamente amortiguado, es decir, no presenta oscilaciones en su respuesta.

- Si $\zeta > 1$, las raíces son reales y el sistema es sobreamortiguado. La respuesta es similar a la de un sistema de primer orden y tiende asintóticamente al valor en régimen permanente. El polo que tiene más influencia en la respuesta es aquel que se encuentra más cerca del eje $j\omega$.

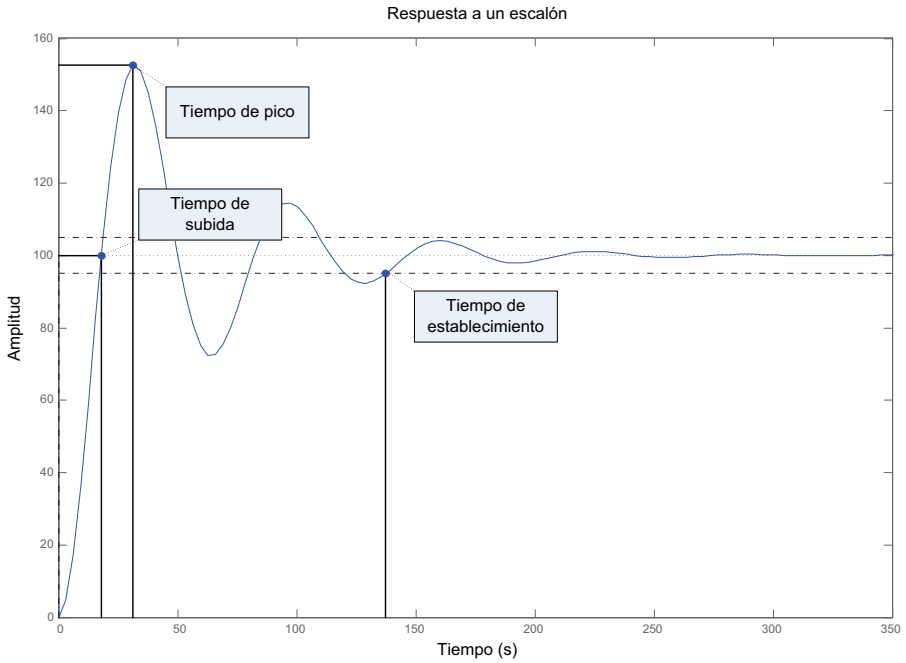


Figura 3.9. Características más significativas de la respuesta temporal de un sistema de segundo orden.

La estabilidad del sistema depende de los polos, ya que un sistema sólo es estable si todos sus polos tienen una parte real negativa. Por lo tanto, la visualización del coeficiente de amortiguamiento, y de la frecuencia natural no amortiguada es muy útil para reconocer el sistema.

Frente a un escalón unitario, existen otras características significativas de los sistemas de segundo orden, de entre las que caben destacar

- El **tiempo de establecimiento**, que en este caso se puede calcular como $t_s = \frac{\pi}{\sigma}$, es decir, es inversamente proporcional a la atenuación.
- El **tiempo de pico**, que es el tiempo requerido para que la respuesta alcance el primer pico de sobreoscilación, $t_p = \frac{\pi}{\omega_d}$, es decir, inversamente proporcional a la frecuencia forzada.
- La **sobreoscilación**, que es el tanto por ciento del valor en que el sistema sobrepasa el valor estabilizado de la respuesta en el tiempo de pico

$$M_p(\,\%) = e^{(-\zeta\pi/\sqrt{1-\zeta^2})} \cdot 100 \tag{3.17}$$

- El **tiempo de subida**, que generalmente se calcula como el tiempo necesario

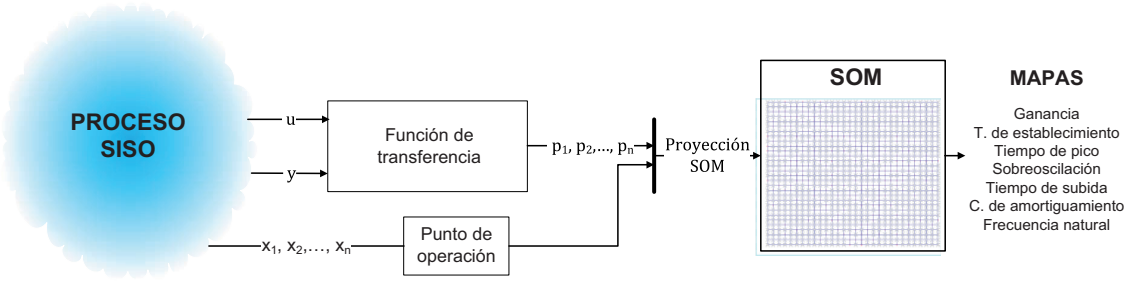


Figura 3.10. Generación de mapas de dinámica en sistemas monovariable.

para alcanzar el 100 % del valor final en sistemas subamortiguados y el tiempo para que la salida pase del 10 % al 90 % en sistemas sobreamortiguados.

En la Fig. 3.9 se muestran algunas de estas características de la respuesta. Para todas es posible generar mapas de dinámica que muestren el valor escalar correspondiente a cada punto de operación.

Por tanto, se definen mapas de visualización para la ganancia , el tiempo de establecimiento, el tiempo de pico, la sobreoscilación, el tiempo de subida, el coeficiente de amortiguamiento y la frecuencia natural (ver figura 3.10). Los mapas se generan de acuerdo al procedimiento descrito en el apartado anterior. Todas estas características son escalares, así que se utiliza un código de color para representar su valor. Las componentes real (atenuación) e imaginaria (frecuencia forzada) del polo no se representan porque son redundantes con respecto al mapa de tiempo de establecimiento y el mapa de tiempo de pico.

Puesto que es posible visualizar diferentes cualidades de un punto de operación al mismo tiempo, un simple análisis visual puede servir al ingeniero de control para determinar qué regiones presentan una respuesta más adecuada o cuáles pueden resultar problemáticas. Asimismo, la visualización conjunta de estos mapas y los planos de componentes de los selectores de dinámica puede permitir ahondar en la influencia del valor de ciertas variables en un punto de funcionamiento y su comportamiento dinámico asociado.

Existen otros parámetros, habituales en el análisis de sistemas, que también serían susceptibles de ser visualizados mediante el método que se describe en la presente tesis, como los ceros, o raíces del numerador, cuyo signo es importante pues en caso de ser positivos imponen limitaciones fundamentales al control. Otras características importantes son el tiempo de retardo (t_d) y, en el caso de sistemas realimentados, los coeficientes estáticos de error de posición (K_p), velocidad (K_v) o aceleración (K_a).

3.3.4. Mapas de visualización de sistemas MIMO

El análisis y supervisión de procesos multivariable es un área que se puede beneficiar especialmente de las herramientas de visualización objeto de esta tesis, ya que el carácter matricial de sus modelos y la importancia de la direccionalidad de sus características para la mera interpretación de los mismos, dificultan una comprensión intuitiva.

Por ello, se propone definir mapas de dinámica que revelen características importantes de un proceso multivariable, las cuales se consideran en los métodos habitualmente utilizados para su análisis, como las direcciones de máxima y mínima ganancia, el condicionamiento de la matriz, los ceros en el semiplano derecho que dificultan el control o la interacción entre las entradas y las salidas (Albertos y Sala, 2004; Skogestad y Postlethwaite, 1996).

Ante un sistema multivariable, existe la posibilidad de utilizar una configuración de control descentralizada, en la que la planta se descompone en múltiples lazos de control SISO. En esta situación, la medida de la interacción entre entradas y salidas es de crucial importancia para seleccionar el mejor emparejamiento entre las mismas. Para este propósito, se utiliza generalmente el **array de ganancias relativas** (RGA o $\Lambda(G)$), un procedimiento relativamente simple e independiente de la escala. La matriz RGA se define como

$$\Lambda(G) = [\lambda_{ij}] = G(s) \otimes (G(s)^T)^{-1}, \tag{3.18}$$

donde $\otimes$ es el producto de Schur. Cada λ en la RGA representa el cociente de la ganancia de un par entrada-salida en lazo abierto entre la ganancia del mismo par cuando el resto de pares están perfectamente controladas con sus entradas. Generalmente se utiliza la ganancia en régimen permanente o a las frecuencias de cruce para el cálculo de la misma (Skogestad y Postlethwaite, 1996).

Si el emparejamiento seleccionado para el lazo SISO es un buen candidato, es decir, si la presencia de otros lazos no tiene gran influencia en su comportamiento, se obtendrá para el mismo una ganancia relativa λ cercana a 1. Consideremos un proceso (2×2). Puesto que los elementos de cada fila y columna de Λ suman 1, se pueden extraer todo un conjunto de reglas a partir del valor escalar λ_{11}:

1. Si $\lambda_{11} = 1$, los pares adecuados son $y_1 - u_1$ e $y_2 - u_2$ y no existe interacción entre los mismos.

2. Si $\lambda_{11} = 0$, es necesario seleccionar los pares cruzados, pero no existe interacción entre los lazos.

3. Si $\lambda_{11} = 0,5$, se trata del peor caso, ya que ambas entradas afectan a ambas salidas en la misma medida.

4. Si $0,5 < \lambda_{11} < 1$, los pares más adecuados son $y_1 - u_1$ y $y_2 - u_2$, pero existe interacción.

5. Si $0 < \lambda_{11} < 0,5$, los pares más adecuados son, al contrario, $y_1 - u_2$ y $y_2 - u_1$.

6. Si $\lambda_{11} > 1$, los pares a seleccionar son $y_1 - u_1$ y $y_2 - u_2$, pero las respuestas se ven frenadas por la interacción con el otro lazo. Es necesario, por tanto, utilizar ganancias grandes en el controlador. Los pares $y_1 - u_2$ y $y_2 - u_1$ no se pueden seleccionar porque sus ganancias relativas son negativas y serían inestables.

7. Si $\lambda_{11} < 0$, los pares a seleccionar son $y_1 - u_2$ y $y_2 - u_1$, pero sometidos al mismo efecto negativo del caso anterior.

Cuando no se puede conseguir una buena descomposición para el control descentralizado, una estrategia es transformar la matriz de funciones de transferencia en una matriz diagonal, es decir, realizar un *desacoplado*. Así se pueden conseguir mejores resultados que mediante la descentralización directa, a costa de perder significado intuitivo. La **descomposición en valores singulares** (SVD), que es una factorización de matrices, es útil para obtener ecuaciones desacopladas entre combinaciones lineales de sensores y combinaciones lineales de actuadores cuando se aplica la matriz de ganancias en régimen permanente. La idea en que se basa el desacoplamiento es que si consideramos las combinaciones de variables de entrada y de salida como vectores, dada una entrada con magnitud constante, la magnitud de salida depende de la dirección de entrada. Y, por lo tanto, es interesante conocer en qué direcciones se presentan la ganancia mínima y máxima.

La descomposición de una matriz G de dimensiones $n \times m$ en valores singulares

$$G = U\Sigma V^T, \tag{3.19}$$

resulta en tres matrices donde U es una matriz ortonormal $n \times n$, V otra matriz ortonormal $m \times m$ y Σ es una matriz diagonal cuyos elementos son valores singulares ordenados de forma descendente, $\sigma_1 \geq sigma_2 \geq \ldots \geq \sigma_m \geq 0$. Matemáticamente, U y V representan una rotación de coordenadas. En el campo de control multivariable, no obstante, las tres matrices resultantes tienen una interpretación física. Si G es una matriz de ganancias del proceso, entonces U puede verse como una matriz de sensores en la que la primera columna es la mejor combinación de salida, es decir, la más sensible a los cambios en las entradas, la segunda columna es la siguiente mejor y así sucesivamente. La matriz V equivale a las entradas y la primera columna es la combinación de entradas con el mayor efecto en el proceso. La segunda columna es la siguiente mejor y así sucesivamente. La dirección que se obtiene con las mejores combinaciones de entrada y salida es la más fuerte o de

mayor ganancia, la cual es proporcionada por σ_1. Puesto que los σ_i están ordenados de forma descendente, el par entrada-salida que corresponde a σ_m es el de menor ganancia y, por tanto, el más difícil de controlar. Puesto que la descomposición en valores singulares separa la direccionalidad de la magnitud, los criterios SISO basados en el valor absoluto de una característica pueden ser generalizados al caso multivariable considerando el mayor valor singular.

Además de su aplicación a la matriz de ganancias en estado permanente ($G(s = 0)$ o $G(z = 1)$) para desacoplar el sistema, también es útil para analizar el rendimiento en el dominio de frecuencia, cuando se realiza sobre matrices de ganancias $G(j\omega)$, o para calcular aproximadamente las direcciones de zeros y polos, cuando se aplica a $G(cero)$ y $G(polo)$.

La dificultad del control debido a la direccionalidad del sistema no sólo puede estimarse por medio de la RGA. También resulta útil estudiar el condicionamiento de la matriz de ganancias, para lo que se hace uso de nuevo de la descomposición en valores, ya que depende de la diferencia de magnitud de los valores de Σ. Un valor representable de esta cualidad es el **número de condición**,

$$\gamma(G) = \frac{\sigma_1(G)}{\sigma_m(G)}. \tag{3.20}$$

Si existe una diferencia muy grande entre las ganancias extremas σ_1 y σ_m, la matriz está mal condicionada y el control del sistema es más complicado, debido a que la ganancia de la respuesta varía drásticamente para diferentes direcciones de entrada y eso implica mayor sensibilidad a errores de modelado.

Por otra parte, al igual que en procesos monovariable, los **ceros multivariable** en el semiplano derecho del plano complejo imponen limitaciones que dificultan control. Por esa razón, el análisis de la posición del cero para descubrir si el sistema presenta características de fase no mínima continúa siendo un factor importante. Sin embargo, existe una diferencia crucial con el caso monovariable, ya que los ceros de sistemas multivariable son vectores y, por tanto, poseen direcciones relevantes para el análisis, pues son las que determinan su influencia. Se define un cero de un sistema MIMO como aquel valor de s para el que la matriz $G(s)$ pierde rango, es decir, aquel valor que genera filas o columnas dependientes. No obstante, para matrices multivariable $G(s)$ cuadradas, los ceros son esencialmente los del determinante de la matriz[2].

Las direcciones de esos ceros pueden obtenerse de forma aproximada por medio de la descomposición de valores singulares. Sea z un cero y U_z, Σ_z y V_z las matrices resultantes de la descomposición en valores singulares de $G(z)$, las direcciones de

[2]Aunque este método podría cancelar polos y ceros cercanos que tuviesen diferentes direcciones.

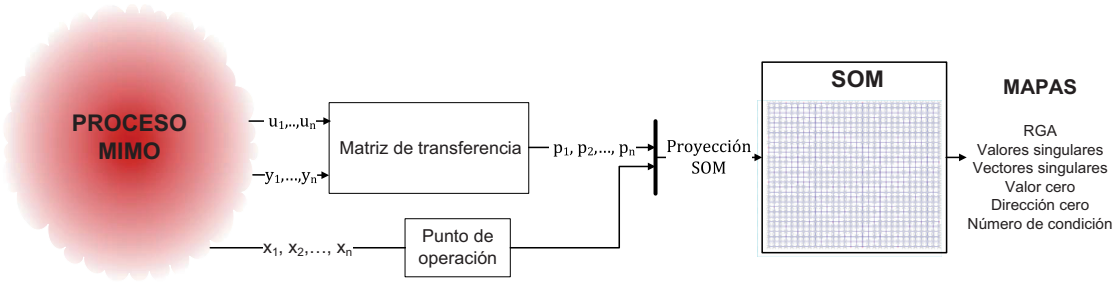


Figura 3.11. Generación de mapas de dinámica en sistemas multivariable.

entrada y salida de z son, respectivamente, la última columna de V_z y la última columna de U_z.

Se pueden definir mapas de visualización para todas las propiedades anteriores, tanto de sus valores como de sus direcciones, es decir, *mapas de dinámica de ganancias relativas, de valores y vectores singulares, de número de condición y de ceros multivariable*. En el caso de sistemas 2×2, la representación de las ganancias máxima y mínima y de sus direcciones de salida y entrada asociadas puede realizarse en el mismo mapa, utilizando el color para representar el escalar y mostrando un vector 2-D en cada punto de funcionamiento. Lo mismo se puede decir de los ceros y su dirección de salida. Para generar las flechas se toman los valores absolutos de las componentes, pues se pretende hacer más sencilla su interpretación y estamos interesados en sus magnitudes. También es útil combinar la visualización del array de ganancias relativas con la de planos de las variables selectoras de la dinámica, para buscar correlaciones entre determinados valores de variables y la conveniencia de usar un emparejamiento entrada-salida u otro. Los mapas se generan de acuerdo al procedimiento definido al principio de este capítulo (ver 3.11).

Además de las características propuestas para su visualización, la técnica de mapas de dinámica es susceptible de ser utilizada para representar otras características relacionadas con sistemas multivariables (Albertos y Sala, 2004; Skogestad y Postlethwaite, 1996), como los polos multivariable y sus direcciones, que permitirían estudiar las posibles limitaciones al control, o el índice de Niederlinski, que podría sugerir correlaciones entre la estabilidad del lazo cerrado y otras características dinámicas.

Capítulo 4

Definición de los experimentos

4.1. Introducción

En este capítulo, se describen el equipamiento utilizado, los experimentos propuestos y los procedimientos de desarrollo y evaluación que se utilizan para verificar experimentalmente la metodología propuesta.

Puesto que el objetivo de esta tesis es analizar y proponer herramientas para la extracción del conocimiento en procesos industriales, es necesario recurrir a sistemas reales sobre los que poder definir experimentos realistas. Para ello, se han utilizado las maquetas de procesos industriales reales (Domínguez *et al.*, 2004, 2005), desarrolladas por el grupo de Automática y Control del Instituto de Automática y Fabricación de la Universidad de León, al que pertenece el autor, con fines educativos y de investigación. Una de ellas permite controlar diversas variables mientras que la otra implementa un proceso multivariable ampliamente conocido y estudiado.

Asimismo, se utiliza la arquitectura de adquisición y control sobre la que se sostiene el Laboratorio Remoto de Automática de la Universidad de León (Domínguez *et al.*, 2005, 2008) para facilitar la adquisición, almacenamiento y posterior recuperación de los datos del proceso objeto de experimentación.

Para los experimentos de modelado local de la dinámica, también se utilizan series temporales sintéticas, que debido a su complejidad se utilizan a menudo como *benchmarks* de predicción.

4.2. Sistemas físicos

4.2.1. Maqueta industrial de 4 variables

Esta maqueta (ver Figs. 4.2 y 4.1) permite controlar cuatro variables físicas (presión, caudal, nivel y temperatura) y está compuesta por un circuito principal de proceso y dos circuitos auxiliares. La relación de variables se muestra en la Tabla 4.1.

El circuito de proceso incorpora dos depósitos en cascada de 10 y 15, 5 litros de capacidad, una bomba centrífuga con accionamiento a velocidad variable que aporta un caudal de hasta 22 l/m, una válvula de aporte neumática, una electroválvula de desagüe y los sensores necesarios para implementar los lazos de control de presión, caudal, nivel y temperatura del fluido de proceso.

El circuito de calentamiento produce agua caliente por medio de resistencias eléctricas con accionamiento variable estático. Un intercambiador de placas de alto rendimiento permite transferir calor al circuito de proceso. El caudal de agua caliente se regula por medio de una válvula de tres vías motorizada.

El circuito de enfriamiento hace posible disminuir la temperatura del proceso mediante agua obtenida de un circuito externo de enfriamiento. La regulación se realiza por medio de una válvula de dos vías motorizada y la transferencia al proceso se lleva a cabo mediante un intercambiador de placas de similares características al del circuito de calentamiento.

Este sistema se utiliza como banco de pruebas para validar métodos presentados en esta tesis, que se describen en las secciones 4.4.3 y 4.5.

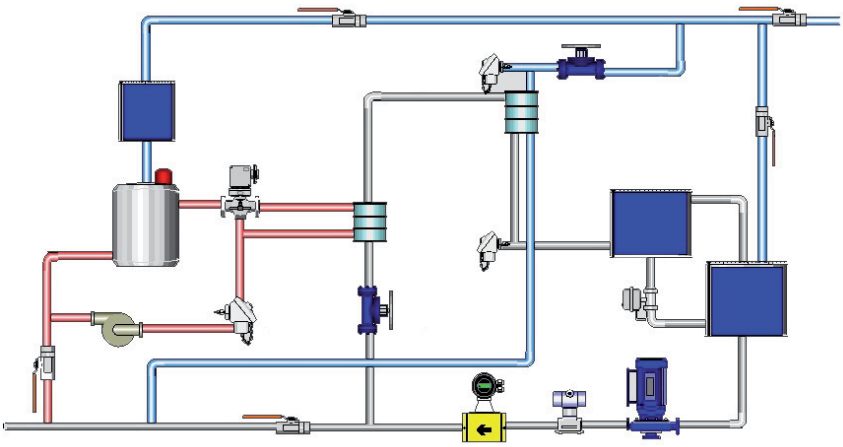


Figura 4.1. Esquema de la maqueta industrial de 4 variables.

TAG	DECRIPCIÓN	TIPO	SEÑAL	RANGO	E/S
TT21	TRANSMISOR DE Tª AGUA CALIENTE	ANALÓGICA	-20...20 mA	-150º - 100ºC	ENTRADA
TT23	TRANSMISOR Tª AGUA FRIA	ANALÓGICA	-20...20 mA	-150º - 100ºC	ENTRADA
TT22	TRANSMISOR TEMPERATURA PROCESO	ANALÓGICA	-20...20 mA	-20 - 20	ENTRADA
LT21	TRANSMISOR NIVEL DEPÓSITO D03	ANALÓGICA	-20...20 mA	-20 - 20	ENTRADA
FT21	TRANSMISOR DE CAUDAL DE PROCESO	ANALÓGICA	-20...20 mA	-20 – 20	ENTRADA
PT21	TRANSMISOR DE PRESIÓN DE PROCESO	ANALÓGICA	-20...20 mA	-20 – 20	ENTRADA
JZ21	CONTROLADOR DE POTENCIA	ANALÓGICA	4...20 mA	0 – 100%	SALIDA
TV22	VÁLVULA TEMPERATURA AGUA FRIA	ANALÓGICA	4...20 mA	0 – 100%	SALIDA
FV21	VÁLVULA CAUDAL DE PROCESO	ANALÓGICA	4...20 mA	0 – 100%	SALIDA
SZ21	CONVERTIDOR BOMBA PROCESO	ANALÓGICA	4...20 mA	0 – 100%	SALIDA
TV21	VÁLVULA DE 3 VÍAS AGUA CALIENTE	ANALÓGICA	4...20 mA	0 – 100%	SALIDA
LSH21	INTERRUPTOR NIVEL ALTO DEPÓSITO D02	DIGITAL	10 voltios		ENTRADA
LSL21	INTERRUPTOR NIVEL BAJO DEPÓSITO D02	DIGITAL	10 voltios		ENTRADA
LSH22	INTERRUPTOR NIVEL ALTO DEPÓSITO D04	DIGITAL	10 voltios		ENTRADA
LSL22	INTERRUPTOR NIVEL BAJO DEPÓSITO D04	DIGITAL	10 voltios		ENTRADA
TSH22	INTERRUPTOR DE TEMPERATURA	DIGITAL	10 voltios		ENTRADA
P22	BOMBA CENTRÍFUGA DE PRODUCTO	DIGITAL	10 voltios		SALIDA
P21	BOMBA AGUA CALIENTE	DIGITAL	10 voltios		SALIDA
FY22	ELECTROVÁLVULA	DIGITAL	10 voltios		SALIDA
ES21	ALTA TEMPERATURA REACTOR	DIGITAL	10 voltios		ENTRADA
ES22	FALLO BOMBA AGUA P01	DIGITAL	10 voltios		ENTRADA
ES23	FALLO VARIADOR BOMBA P02	DIGITAL	10 voltios		ENTRADA
ES24	FALLO RESISTENCIAS	DIGITAL	10 voltios		ENTRADA
ES25	CONFIRMACIÓN MARCHA BOMBA P01	DIGITAL	10 voltios		ENTRADA
ES26	CONFIRMACIÓN MARCHA BOMBA P02	DIGITAL	10 voltios		ENTRADA
ES27	CONFIRMACIÓN MARCHA RESISTENCIAS	DIGITAL	10 voltios		ENTRADA

Tabla 4.1. Variables de la maqueta industrial de 4 variables.

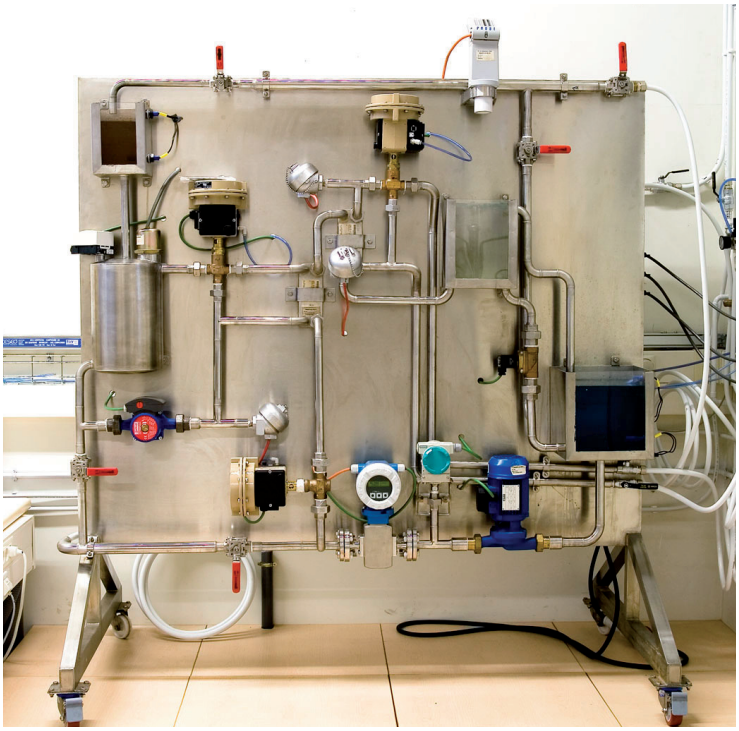


Figura 4.2. Maqueta industrial de 4 variables.

4.2.2. Maqueta de 4 tanques

Para los experimentos de visualización de la dinámica en sistemas multivariable, se ha utilizado una maqueta industrial de cuatro tanques (Domínguez *et al.*, 2005), que implementa el proceso propuesto por (Johansson, 2000). Este modelo, diseñado para mostrar los problemas provocados por la posición de los ceros en sistemas de control multivariable, está compuesto por dos bombas y cuatro tanques interconectados de tal manera que los tanques superiores desaguan en los inferiores. La maqueta mantiene la estructura original, pero se ha construido con instrumentación industrial común (ver Fig. 4.3). Además de la utilizada para los experimentos, existen réplicas de la misma en la Universidad de Oviedo, la Universidad de Almería y la UNED, también desarrolladas por el Instituto de Automática y Control de la Universidad de León.

La planta se compone de cuatro tanques de 8 litros donde el nivel se mide por medio de unos transmisores de presión *Endress & Hauser PCM 731*. El líquido es impulsado por bombas *Grundfos UPE 25-40*, equipadas con módulos de expansión *Grundfos MC 40/60*, que son regulables mediante una señal analógica. Cuenta con dos válvulas neumáticas de tres vías *Samson 3226*, de apertura regulable con

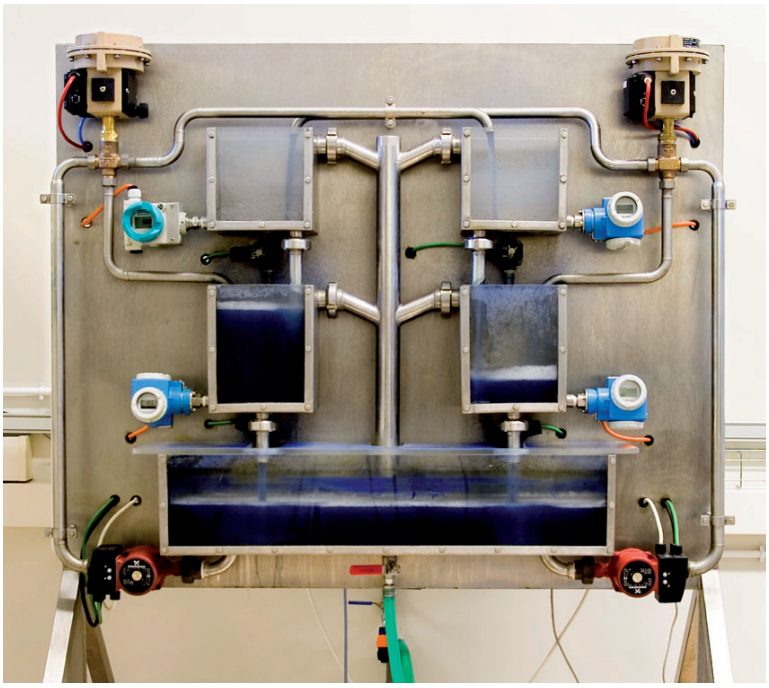


Figura 4.3. Maqueta de 4 tanques.

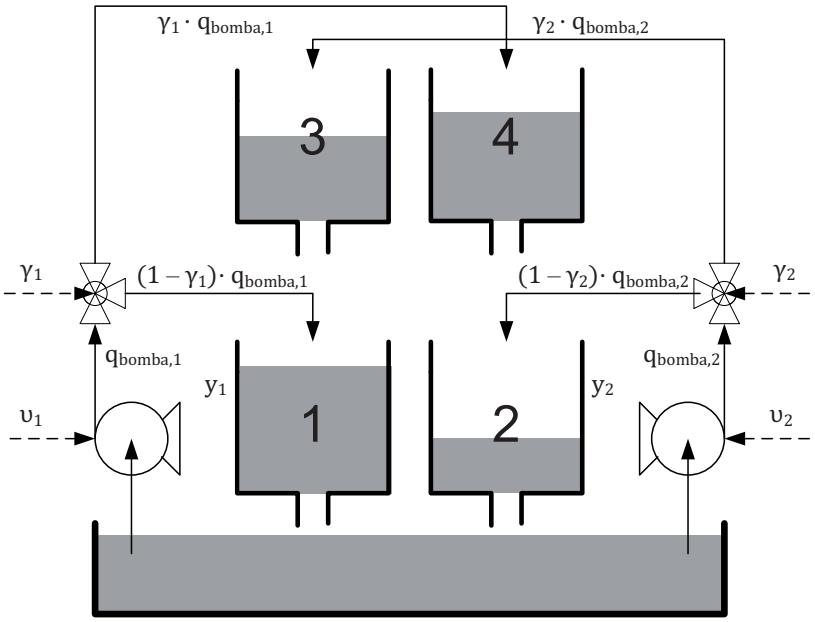


Figura 4.4. Esquema de la maqueta de 4 tanques.

posicionadores *Samson 3760*, que determinan la distribución del caudal entre los tanques de la diagonal correspondiente, y electroválvulas todo/nada *SMC*, cuyo propósito es generar perturbaciones en el nivel de los tanques, y que se encuentran instaladas en la base de los mismos. El diagrama 4.4 muestra un esquema del proceso y la tabla 4.2 una descripción de las variables accesibles de la maqueta.

Sean υ_1 y υ_2 las consignas de entrada a las bombas, h_1 y h_2 los niveles de los tanques inferiores (salidas) y γ_1, γ_2 las posiciones de las válvulas de tres vías. Una tabla con las variables del modelo se presentan a continuación.

Variable	Unidad	Descripción
h_i	cm	Nivel en los tanques
h_i^0	cm	Estado inicial de los tanques
x_i	cm	Variación de nivel $x_i = h_i - h_i^0$
q_i	cm^3/s	Caudal de las bombas a los tanques
υ_j	$0-100$	Consigna de las bomba
υ_j^0	$0-100$	Consigna inicial
u_j	$0-100$	Variación del voltaje de la bomba $u_i = \upsilon_j - \upsilon_j^0$
$q_{bomba,j}$	cm^3/s	Caudal total de las bombas
γ_j	$0-1$	Ratio de las válvulas
Constante	**Unidad**	**Descripción**
A_i	cm^2	Sección de los tanques
a_i	cm^2	Sección de las salidas
g	cm^2/s	Aceleración debido a la gravedad
k_j	cm^3/s	Constante de las bombas
k_c	cm	Constante de nivel

Tabla 4.2. Variables de la maqueta de cuatro tanques

Aplicando la ley de Bernoulli y el balance de masas, se puede obtener el modelo del sistema. Si las ecuaciones diferenciales se linealizan en torno a los puntos de

trabajo v_1^0, v_2^0, se puede obtener una representación en espacio de estados:

$$\frac{dx}{dt} = \begin{pmatrix} -\frac{1}{T_1} & 0 & \frac{A_3}{A_1 T_3} & 0 \\ 0 & -\frac{1}{T_2} & 0 & \frac{A_4}{A_2 T_4} \\ 0 & 0 & -\frac{1}{T_3} & 0 \\ 0 & 0 & 0 & -\frac{1}{T_4} \end{pmatrix} x + \begin{pmatrix} \frac{\gamma_1 k_1}{A_1} & 0 \\ 0 & \frac{\gamma_2 k_2}{A_2} \\ 0 & \frac{(1-\gamma_2)k_2}{A_3} \\ \frac{(1-\gamma_1)k_1}{A_4} & 0 \end{pmatrix} u \tag{4.1}$$

$$y = \begin{pmatrix} k_c & 0 & 0 & 0 \\ 0 & k_c & 0 & 0 \end{pmatrix} x$$

donde las constantes de tiempo son

$$T_i = \frac{A_i}{a_i}\sqrt{\frac{2h_i^0}{g}} \quad i = 1, \cdots, 4 \tag{4.2}$$

Aplicando la transformada de Laplace, se obtiene la matriz de funciones de transferencia del sistema

$$\mathbf{G}(s) = \begin{pmatrix} \frac{\gamma_1 c_1}{1+sT_1} & \frac{(1-\gamma_2)c_1}{1+(T_3+T_1)s+T_3T_1s^2} \\ \frac{(1-\gamma_1)c_2}{1+(T_4+T_2)s+T_4T_2s^2} & \frac{\gamma_2 c_2}{1+sT_2} \end{pmatrix}. \tag{4.3}$$

La utilidad de este modelo para la didáctica de control multivariable proviene de la relación directa que existe entre sus características dinámicas y la posición de las válvulas.

Este sistema se utiliza como banco de pruebas para validar métodos presentados en esta tesis, que se describen en la sección 4.6.

4.3. Arquitectura de comunicaciones, control y adquisición de datos

4.3.1. Laboratorio Remoto de Automática

Para la implementación y adquisición de datos de los experimentos propuestos para esta tesis, se ha utilizado la arquitectura de red, adquisición y control sobre

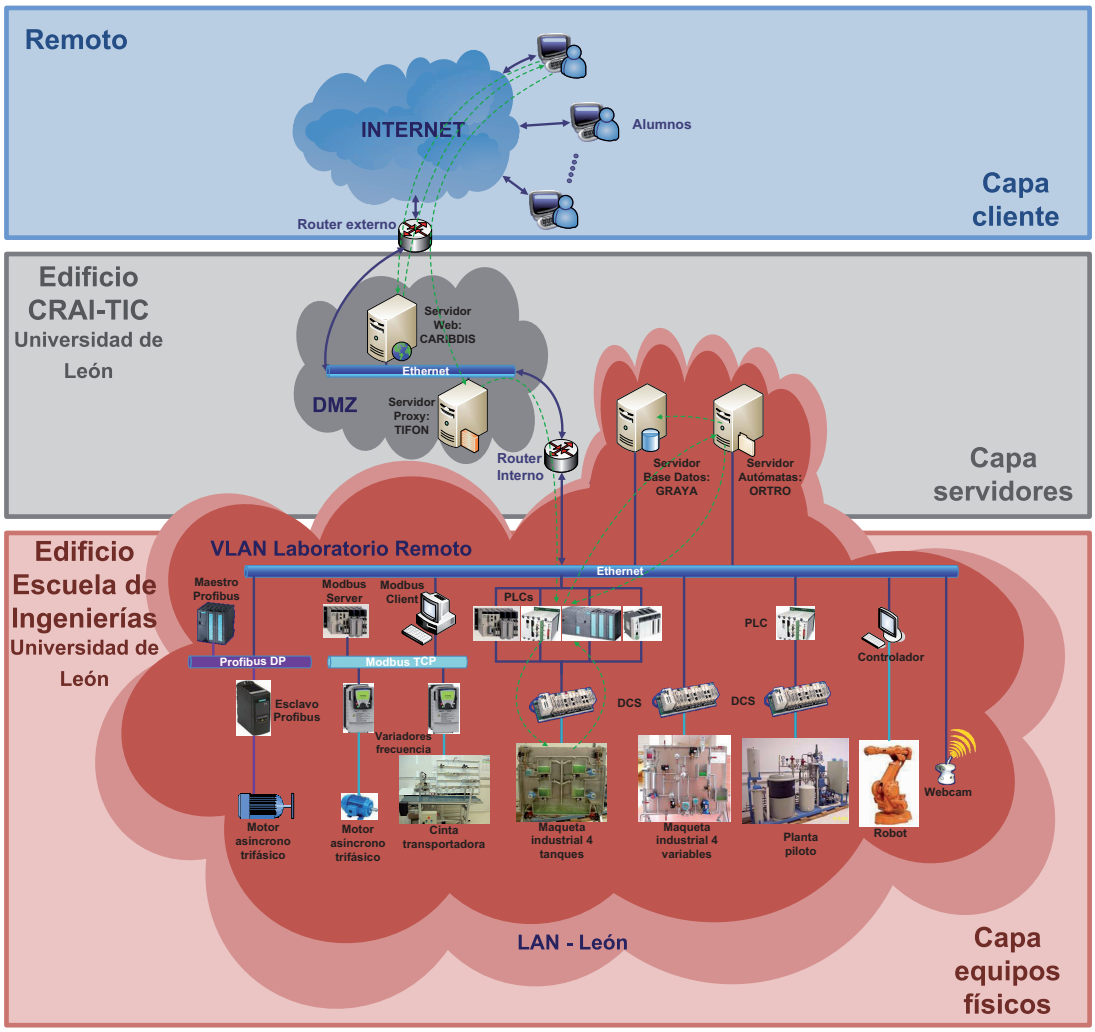


Figura 4.5. Arquitectura de comunicaciones.

la que se sostiene el Laboratorio Remoto de Automática de la Universidad de León (Domínguez *et al.*, 2005, 2008).

Esta plataforma tecnológica tiene como propósito el hacer disponibles remotamente una serie de equipos físicos reales, de forma flexible y cómoda para el usuario. Internet proporciona una oportunidad única para hacer posible el acceso a equipamiento industrial que, ya sea por su precio u otras razones logísticas, puede no estar disponible en los laboratorios de los centros educativos (Dormido, 2004). La experimentación con este equipamiento es esencial para garantizar una experiencia de aprendizaje completa del alumno y algunos estudios (Nickerson *et al.*, 2007) apuntan a que la efectividad de los laboratorios remotos no sólo es comparable a la de los tradicionales sino que podría ser incluso mejor. Los profesores también pueden utilizar los sistemas como apoyo a sus lecciones teóricas, ya que son fácilmente accesibles a través del laboratorio. Algunos de los objetivos que se pretenden alcanzar con este laboratorio remoto son:

- Disponibilidad de diferentes tecnologías de automatización, control y supervisión.
- Información acerca del estado del proceso activo mediante vídeo online, gráficas y sinópticos.
- Facilitar el manejo de los datos obtenidos y su análisis con software científico.
- Interactividad y retardos temporales razonables.
- Modularidad, escalabilidad y facilidad de uso.
- Facilitar su uso autónomo junto con material de soporte didáctico adicional (descripciones, tablas, imágenes, especificaciones,...).

Para hacer posible el acceso remoto a los equipos industriales y garantizar ciertas condiciones de rendimiento, seguridad y usabilidad, se necesita disponer de una estructura de red compleja que determina tanto la arquitectura local como la conexión a Internet. A continuación se describen brevemente esa estructura, sus principales componentes y la tecnología que se utiliza para implementarla.

La arquitectura local del laboratorio cuenta con una red Ethernet que conecta autómatas programables, sistemas de adquisición y PCs. Para la conexión con Internet se utilizan cuatro servidores. Dos de ellos, el servidor Web y el servidor proxy, se encuentran conectados al exterior, mientras que los otros dos, el servidor de datos y el servidor de enlace de controladores, están en la intranet. En la figura 4.5 se presenta una visión global de la arquitectura de red del laboratorio.

Los usuarios pueden acceder al laboratorio remoto a través de un interfaz web, que ofrece los entornos de operación y otros contenidos didácticos adicionales (ver Fig. 4.6). El interfaz web ha sido implementado en PHP y HTML con un sistema gestor de contenidos Drupal y sobre un servidor web IIS 6.0-Windows 2003. El sistema gestor de contenidos se ocupa de la autenticación de usuarios. Los elementos

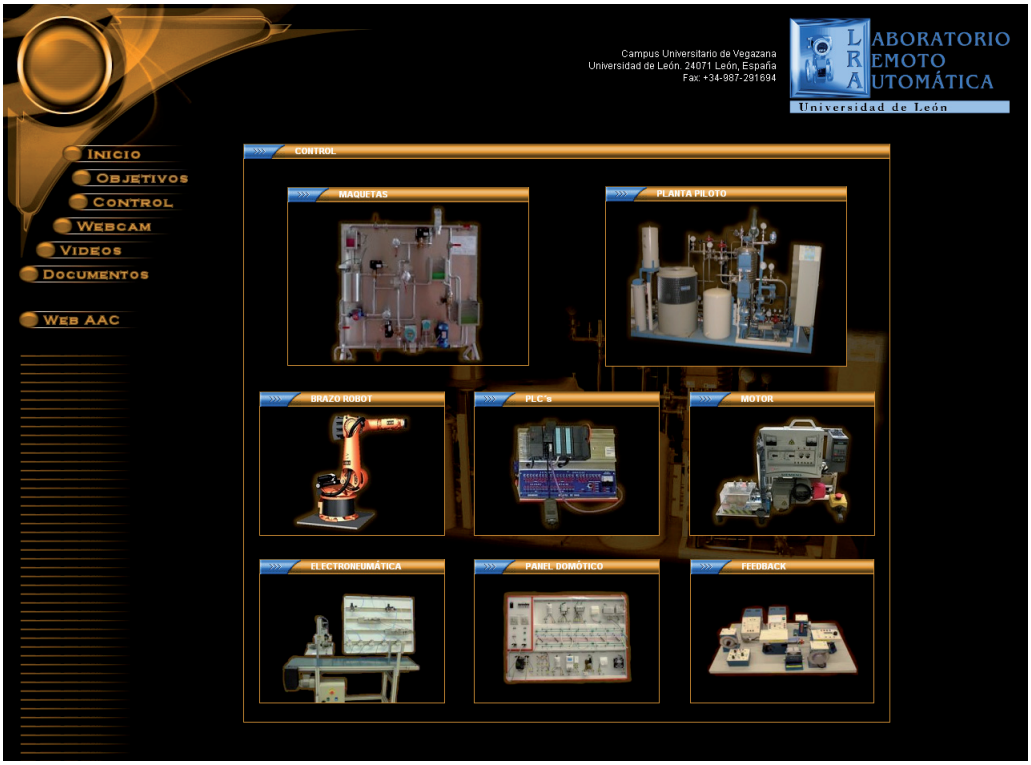


Figura 4.6. Laboratorio remoto de automática de la Universidad de León.

dinámicos como las gráficas, sinópticos o vídeo, que ofrecen al usuario la respuesta a sus acciones sobre los sistemas físicos, han sido implementadas en Java o con *virtual instruments* de LabVIEW. La lectura de los datos se realiza por medio de un CGI (*common gateway interface*) implementado en Visual C++.

Uno de los objetivos del laboratorio remoto es conseguir independencia del sistema de control para el manejo de los sistemas físicos, es decir, que los usuarios puedan utilizar el mismo sistema con diferentes controladores. Así pues, se necesita un sistema que realice el cambio de sistemas de control y establezca el enlace tanto hacia el sistema físico como hacia la web y base de datos. Esto se consigue por medio de la tecnología OPC (*Object Linking and Embedding for Process Control*), una serie de estándares definidos para facilitar la interoperabilidad y conectividad de aplicaciones de automática y control. Un *middleware* desarrollado por el grupo de Automática y Control permite enlazar el servidor OPC asociado a la tarjeta de adquisición de datos cableada a cada sistema con el servidor OPC del PLC requerido, que potencialmente puede ser de otro fabricante. El almacenamiento de los datos en el sistema gestor de bases de datos y la obtención de los mismos a través del CGI también son gestionados por esta aplicación. Salvo error de hardwa-

re, también garantiza que el controlador seleccionado se encuentra en la condición adecuada, pues revierte los posibles cambios nocivos causados por usuarios anteriores. Tanto el middleware como los servidores OPC se encuentran en el servidor de enlace de controladores. Actualmente, se pueden utilizar PLCs Opto LCM4, Opto SNAP B3000-ENET, Siemens S7 300, Schneider TSX Premium o Moeller XC-CPLI201.

Los datos históricos del proceso son almacenados en el servidor de base de datos, cuando existen cambios en los dispositivos físicos. Como sistema gestor de bases de datos se utiliza Microsoft SQL Server 2005 sobre Windows 2003. La cadencia de almacenamiento podría llegar a ser de hasta 50 muestras por segundo, por lo que es necesario contar con políticas que garanticen que el rendimiento del sistema en su conjunto no empeore. El almacenamiento de los datos históricos permite a un usuario del laboratorio remoto obtener los datos correspondientes a una experiencia que haya realizado previamente para su análisis posterior.

Por otra parte, también se pretende que el usuario pueda utilizar el propio software del fabricante del PLC para conectarse a los sistemas remotos, de tal manera que pueda programar una estrategia de control en alguno de los lenguajes soportados y cargarla en el sistema como si fuera accesible localmente. Puesto que los sistemas físicos poseen direcciones IP privadas, no es posible establecer una conexión directa desde internet. Por esa razón, se hace necesario el uso de un proxy y de la traducción de direcciones de red (NAT). No obstante, cualquier red corporativa impide, por política de seguridad, el establecimiento de una conexión desde el exterior a cualquier equipo de la misma que no se encuentre en lo que se denomina zona desmilitarizada (DMZ) o *screened subnet*. Esta zona es una subred que expone los servicios de la organización a Internet y restringe las conexiones entrantes para añadir una capa adicional de seguridad, útil en el caso de que un atacante comprometa uno de los equipos expuestos (Dubrawsky *et al.*, 2006). Para ello, cuenta con al menos dos firewalls, uno interno que bloquea el tráfico entre la DMZ y la VLAN y otro externo que bloquea el tráfico desde y hacia Internet. Por tanto, para no incrementar los riesgos de seguridad de la red corporativa, es necesario limitar al máximo las conexiones permitidas hacia dentro de la red.

Se justifica el uso de un servidor proxy que se encargue del filtrado de red, rastreo de la conexión y NAT estático y que solamente permita el acceso de usuarios autenticados a los sistemas físicos (Toderick *et al.*, 2005). El servidor proxy establece, por tanto, un túnel temporal entre un sistema físico concreto y un usuario físico concreto, con una IP determinada. Este servidor realiza todo este proceso de una manera transparente que esconde totalmente la complejidad de los enlaces al usuario.

En resumen, la estructura lógica de la red sigue, en cierto modo, lo que en diferentes contextos se denomina arquitectura de tres capas o patrón modelo-

vista-controlador (Gamma *et al.*, 1995). En este caso, el modelo o capa de datos está constituido por los sistemas físicos, que actúan como fuentes o receptores de datos. La capa de aplicación o controlador se encarga de controlar la funcionalidad del sistema y reaccionar a eventos, por lo que incluye gran parte de la lógica implementada en los servidores, salvo la interfaz web, que constituye la capa de presentación o vista del sistema.

Además de las maquetas industriales descritas en los apartados anteriores, existen otros sistemas disponibles para su acceso remoto a través del laboratorio:

- Una planta piloto industrial, desarrollada por el propio grupo de investigación para realizar estrategias avanzadas de control y supervisión. Cuenta con dos reactores, tres circuitos de utilities (frío, calor y vapor) y permite trabajar sobre varias variables físicas simultáneamente.
- Un robot de seis grados de libertad ABB IRB 1400 S4.
- Una célula electroneumática, desarrollada también por el grupo, que permite simular una cadena de montaje. Para ello, dispone de una cinta transportadora, actuadores neumáticos SMC, una mesa de desplazamiento en dos ejes y un manipulador con tres grados de libertad. El sistema es accionado por variadores de frecuencia Schneider Electric Altivar 71.
- Panel domótico diseñado y desarrollado por el grupo GENIA del Área de Sistemas y Automática de la Universidad de Oviedo, que cuenta con diversos sensores y actuadores.

4.3.2. Uso de la plataforma para los experimentos

La plataforma tecnológica que soporta el laboratorio remoto de automática facilita, en gran medida, la conexión de diversos controladores a las plantas y la adquisición, almacenamiento y posterior recuperación de los datos históricos del proceso objeto de experimentación. Esta accesibilidad y flexibilidad ha hecho posible obtener, de forma sencilla, datos de procesos reales que se ajustasen a las necesidades de experimentación. Concretamente, ha resultado muy útil la funcionalidad de almacenamiento de los datos históricos, que posteriormente han sido recuperados, mediante una conexión ODBC, para su análisis en Matlab (ver Fig. 4.7).

Las estrategias de control han sido implementadas en ioControl (lenguaje de programación basado en diagramas de flujo) para un sistema de control distribuido Opto SNAP B3000-ENET. Para el análisis de los datos se han utilizado funciones específicamente implementadas para tal propósito en Matlab y/o Octave, los

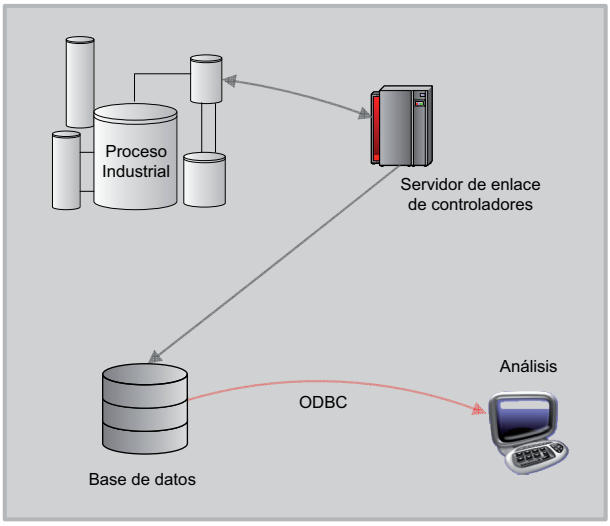


Figura 4.7. Adquisición de datos.

toolboxes para mapas auto-organizados de la Helsinki University of Technology (Vesanto, 2000) y de la Universidad de Oviedo y algunas librerías para la estimación de invariantes dinámicos, que se citan en el siguiente apartado.

4.4. Experimentos de comparación de métodos de modelado local de dinámica basados en SOM

4.4.1. Consideraciones iniciales

Tal y como se propuso en la sección 3.2, en este experimento se lleva a cabo una comparación, en términos de calidad del modelado o reconstrucción de la dinámica, de algoritmos derivados de la idea de utilizar el mapa auto-organizado o sus variantes como infraestructura para el modelado mediante agrupaciones de modelos locales. *Los algoritmos que se utilizan para la comparación son el SOM, el SOM con aprendizaje dinámico, el SOM recurrente, la red SARDNET, la red Merge SOM y el SOMTAD*, los cuales han sido propuestos en la bibliografía y descritos con anterioridad en este documento. Asimismo, en cada uno de los algoritmos anteriores, *la estimación del modelo correspondiente a cada neurona se va a realizar de dos maneras diferentes: por medio de los vectores prototipo correspondientes a la neurona ganadora y sus vecinas o por medio de los vectores de entrada asociados a la región de Voronoi de la neurona (y de sus vecinas, si fuese necesario),* excepto

en el caso del SOM recurrente, que por sus características específicas no puede utilizar un ajuste por medio de vectores prototipo[1].

Como se presentó en el citado apartado, para evaluar el modelado de la dinámica de un sistema es necesario evaluar la preservación de los invariantes dinámicos y su error a largo plazo. Para ese propósito, se pueden utilizar el mayor exponente de Lyapunov y la dimensión de correlación como invariantes dinámicos a estimar y el NRMSE (raíz del error cuadrático medio normalizado) como métrica de error de predicción varios pasos adelante.

Las modificaciones del mapa auto-organizado para procesamiento temporal presentan más parámetros libres que el SOM tradicional. Aunque algunos de ellos pueden seleccionarse a partir del conocimiento previo, los parámetros más críticos deben ajustarse siguiendo algún procedimiento. La calidad del modelo va a depender en gran medida de los valores que tomen los parámetros libres de cada algoritmo, entre los que destaca por su importancia el número de retardos que forman las muestras de entrada. Existe multitud de algoritmos y criterios para la selección de variables de entrada en modelos dinámicos, tanto para secuencias autónomas como para sistemas con entradas (Arahal *et al.*, 2009a). Como ejemplos más representativos, para seleccionar un modelo que proporcione buenos resultados sin perder generalización se puede recurrir a una medida como el criterio de información de Akaike (AIC), que busca el modelo más ajustado a los datos con un número de parámetros mínimo, o a un método como la **validación cruzada**, en la que el conjunto de entrenamiento se divide en N subconjuntos y el proceso de entrenamiento se itera N veces de tal manera que, cada vez, uno de los subconjuntos es el conjunto de validación y los otros $N-1$ se utilizan para entrenar. Se obtiene una medida del error medio obtenido en estas validaciones y el proceso se repite para diferentes inicializaciones del algoritmo, seleccionando para la parametrización final aquella inicialización que minimiza el error medio.

McNames (2002) sugiere un método que permite calcular el error de validación cruzada para métodos con múltiples modelos locales, como los que son objeto de estudio en esta tesis, de un modo fácil, relativamente eficiente y que garantiza una buena generalización, ya que penaliza el sobreentrenamiento. Propone que se extraiga un vector del conjunto, construyendo el modelo con los vectores restantes, con los que se trata de estimar el valor predicho por ese vector, y repetir este proceso con todos los vectores. De este modo, el proceso da lugar a una validación cruzada ***leave-one-out***. La medida de error de la validación puede basarse en el error de predicción n pasos adelante, construyendo el modelo sin los puntos $i_1, \ldots, i_n$

[1]Los vectores prototipo se ajustan a $\mathbf{y}$ en lugar de a las muestras de entrada.

$$\overline{C} = \frac{1}{n_c n_s} \sum_{i=1}^{n_c} \sum_{i=1}^{n_s} p(y_{c(i)+j+1} - \widehat{g}^{-}(x_{c(i)+j})), \tag{4.4}$$

lo que resulta bastante apropiado pues refleja el coste real de la predicción iterada y penaliza el sobreentrenamiento. Este planteamiento recibe el nombre de error de validación cruzada multi-paso (MSCVE) y tiene la desventaja de requerir más tiempo de computación, pero (McNames, 1999) da cuenta de sus buenos resultados. Este es el método de ajuste que se va a utilizar en las pruebas.

Puesto que el coste computacional de esta tarea es alto y se desea evitar una explosión combinatoria, la elección de parámetros se realiza entre un conjunto limitado de los mismos que contiene los que se consideran más apropiados a partir del conocimiento previo o a la experiencia. Por ejemplo, en la elección del número de retardos, esencial para obtener una buena reconstrucción, se trabaja con valores cercanos al *embedding* sugerido por el método de índices de Lipschitz (He y Asada, 1993).

Finalmente, un conjunto de datos de test que no haya sido utilizado previamente (para evitar conclusiones sesgadas) se utiliza para evaluar el rendimiento de los modelos, cuyos parámetros han sido fijados previamente en la fase de entrenamiento-validación.

En los apartados siguientes, se detallan los criterios de evaluación y estimación de los invariantes y las series utilizadas, con el objetivo de posibilitar su reproducción. Estas características no siempre se hacen explícitas en la bibliografía, lo que produce medidas y afirmaciones aparentemente contradictorias, que pueden confundir al lector.

4.4.2. Método de evaluación

En primer lugar, procedamos a presentar los criterios de evaluación y los algoritmos utilizados para llevarla a cabo. Como se sugirió anteriormente, para la validación se tienen en cuenta tanto el error de predicción como la preservación de invariantes.

La medida de error para evaluar la predicción iterada n pasos adelante conviene que sea ampliamente usada, para facilitar así la comparación con propuestas anteriores. Por esta razón, se utiliza la **raíz del error cuadrático medio normalizado** (NRMSE), es decir, $NRMSE = \sqrt{NMSE}$, donde el error cuadrático

medio normalizado o NMSE se calcula del siguiente modo:

$$NMSE_p \triangleq \frac{\frac{1}{n_v}\sum_{i=1}^{n_v}(y_{v_i+p} - f_{v_i+p}^{-(v_i+p)^*}(x_{v_i}))^2}{\frac{1}{n}\sum_{i=1}^{n}(y_i - \overline{y})^2}. \tag{4.5}$$

En la ecuación anterior, $f_{v_i+p}^{-(v_i+p)^*}(x_{v_i})$ es la predicción en el momento $v_i + p$ de un modelo iterado p veces. Si $NRMSE_p \geq 1$, el error de predicción no es mejor que utilizar la media.

Los invariantes dinámicos como el máximo exponente de Lyapunov o la dimensión de correlación son estimados a partir de los datos y, como se señaló en la sección 3.2.3, se han propuesto diversos algoritmos para estimarlos. En general, los resultados proporcionados por estos algoritmos son coherentes entre sí bajo condiciones normales. No obstante, para estimar los dos invariantes dinámicos (tanto a partir de las series originales como a partir de las series modeladas), se utilizan dos algoritmos de estimación que presentan mejores características que el resto.

Como estimador de la dimensión de correlación, se utiliza el **método de kernel gaussiano** (GKA), que suaviza las condiciones de otros algoritmos de estimación utilizando una función de base gaussiana (Yu *et al.*, 2000). La mayor ventaja de este algoritmo frente a otros es que permite estimar simultáneamente la dimensión de correlación, la entropía de correlación y el nivel de ruido. Se utiliza la implementación de uno de los autores del algoritmo[2], Small (2005).

Para estimar el máximo exponente de Lyapunov se utiliza el algoritmo propuesto por Kantz (1994), que calcula la tasa de divergencia $S(\Delta n)$ a lo largo del tiempo para diveros valores de *embedding*. Si para algún rango de Δn la función $S(\Delta n)$ exhibe un incremento lineal robusto (es decir, presente con diferentes valores de dimensión de *embedding* y diámetro de vecindad), la pendiente es una estimación del mayor exponente de Lyapunov λ por unidad de tiempo. Se utiliza la implementación disponible en el software TISEAN (Hegger *et al.*, 1999), desarrollado, entre otros, por uno de los autores del algoritmo.

Para estimar estos invariantes, tanto a partir de la señal original como de las modeladas, se genera una serie larga, de 8000 muestras, puesto que un pequeño número de datos puede causar una mala estimación. En el caso de las señales modeladas, la serie se genera de forma autónoma, alimentando la salida predicha a la entrada con cada nueva estimación $\hat{\mathbf{x}}(t+1)$.

[2]Disponible públicamente en http://www.eie.polyu.edu.hk/ ensmall/matlab/

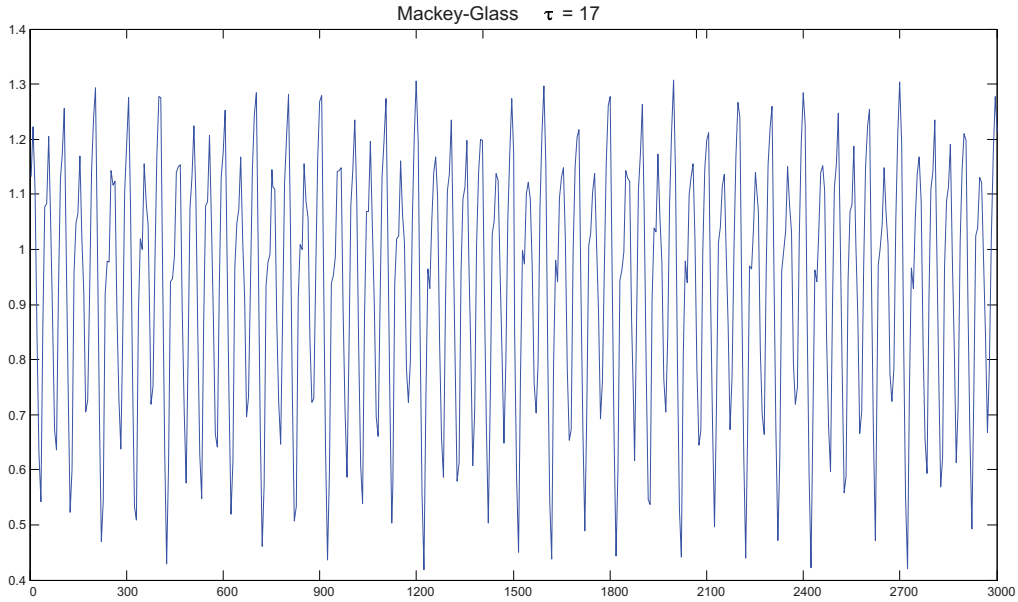


Figura 4.8. Serie de Mackey-Glass ($t_s = 17$).

4.4.3. Series temporales objeto de prueba

Series caóticas clásicas Como se indicó en el capítulo 3, las señales caóticas tienen unas características que les hacen idóneas para valorar métodos de modelado de sistemas no lineales. Por eso, la mayor parte de los autores utilizan este tipo de señales para evaluar sus modelos. De hecho, existe un conjunto de series temporales que pueden considerarse *benchmarks* clásicos de predicción, por su extensivo uso en la bibliografía. Entre ellas se incluyen la ecuación de Mackey-Glass y el mapa de Hénon, que son las seleccionadas para evaluar el modelado y se describen a continuación.

La ecuación diferencial de **Mackey-Glass** (Mackey y Glass, 1977)

$$\frac{dx}{dt} = \frac{ax(t-t_d)}{1+(x(t-t_d))^c} - bx(t), \tag{4.6}$$

inicialmente propuesta como un modelo de la producción de glóbulos blancos, es probablemente el *benchmark* más utilizado para evaluar métodos de predicción de series temporales caóticas (ver Fig. 4.8). Los valores a, b y c generalmente reciben los valores $0,2$, $0,1$ y 10, respectivamente, mientras que para t_d se seleccionan los valores 17 ó 30.

La ecuación MG con $t_d = 17$ resulta especialmente útil como *benchmark* de modelado, pues prácticamente todos los autores calculan el error de predicción con

un horizonte de 84 muestras. Esto supone una predicción de 14 pasos adelante, ya que generalmente se utiliza un retardo de *embedding* $\tau = 6$. Tanto ese t_d como ese horizonte de predicción son los seleccionados para realizar los experimentos de esta tesis. Sin embargo, no existe un consenso tan claro en las estimaciones de los invariantes dinámicos, ya que los resultados presentes en la bibliografía, aún siendo del mismo orden, varían ampliamente debido al uso de diferentes algoritmos de estimación, métodos de preprocesamiento, parámetros e, incluso, escalas (algunos autores expresan el exponente de Lyapunov con base 2, aunque se defina con base e)[3].

Las series que se utilizan en los experimentos se obtienen resolviendo la ecuación diferencial mediante un método Runge-Kutta de orden cuatro con un tamaño de paso igual a $0,01$, $x(0) = 1,2$ y los valores por defecto previamente especificados[4]. Para el entrenamiento de los modelos, se han obtenido 4500 muestras a $1/6Hz$, de las cuales se descartan las 500 primeras para eliminar el estado transitorio. Los 3000 primeros datos se utilizan para entrenamiento, mientras que los restantes se utilizan en la validación. Se utiliza una secuencia más larga (8000 muestras) para estimar los invariantes dinámicos.

Los invariantes estimados para la serie de Mackey-Glass con $t_d = 17$ son $\lambda_1 = 0,0064 \pm 0,0014$ y $D_2 = 1,95 \pm 0,01$, que, respectivamente, se aproximan a los calculados en referencias previas como Gencay y Dechert (1992) y Grassberger y Procaccia (1983). Para estimar estos invariantes se han utilizado los algoritmos presentados en el apartado anterior. En la Fig. 4.9, se presenta la salida de los algoritmos para las dos series originales.

Por otra parte, el **mapa de Hénon** (Hénon, 1976) es un sistema dinámico discreto que presenta un comportamiento caótico y que se define mediante la siguientes ecuación:

$$x_{n+1} = 1 - ax_n^2 + bx_{n-1}. \tag{4.7}$$

En el mapa canónico, los parámetros toman los valores $a = 1,4$ y $b = 0,3$. La serie (ver Fig. 4.10) ha sido obtenida con estos parámetros[5], un retardo $\tau = 1$, que es común en la bibliografía, y un horizonte de predicción 8 pasos adelante.

Para el entrenamiento de los modelos, se han obtenido 4500 muestras a $1/6Hz$, de las cuales se descartan las 500 primeras, para eliminar el estado transitorio.

[3]El máximo exponente de Lyapunov se expresa en esta tesis como exponente con base e, es decir, en $nats/s$.

[4]Para ello, se ha utilizado el código implementado por Eric Wan y disponible públicamente en http://www.cse.ogi.edu/ ericwan/data.html

[5]Se utiliza la implementación de Eric Wan y disponible públicamente en http://www.cse.ogi.edu/ ericwan/data.html.

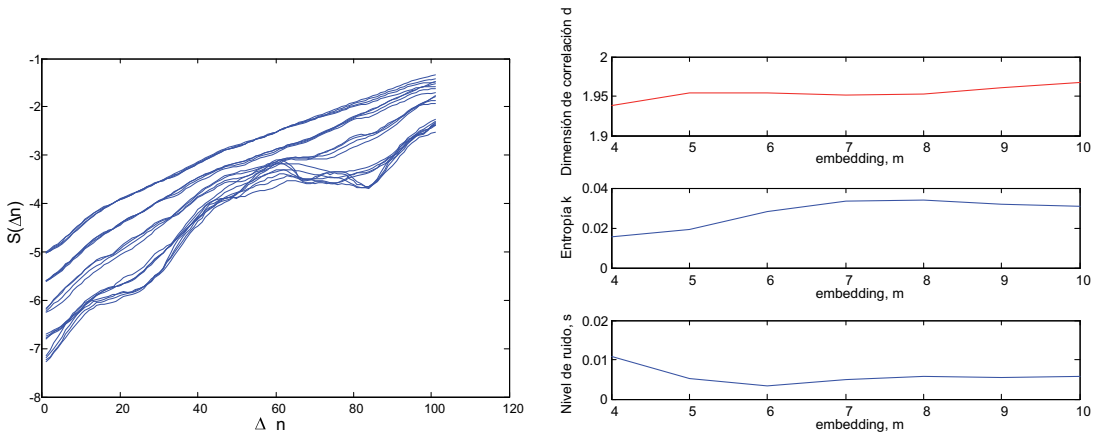


Figura 4.9. Estimación de invariantes para la serie de Mackey-Glass. A la izquierda, curvas que se utilizan para el cálculo del exponente de Lyapunov estimado. A la derecha, dimensión de correlación y otras magnitudes estimadas con respecto a la dimensión de *embedding*.

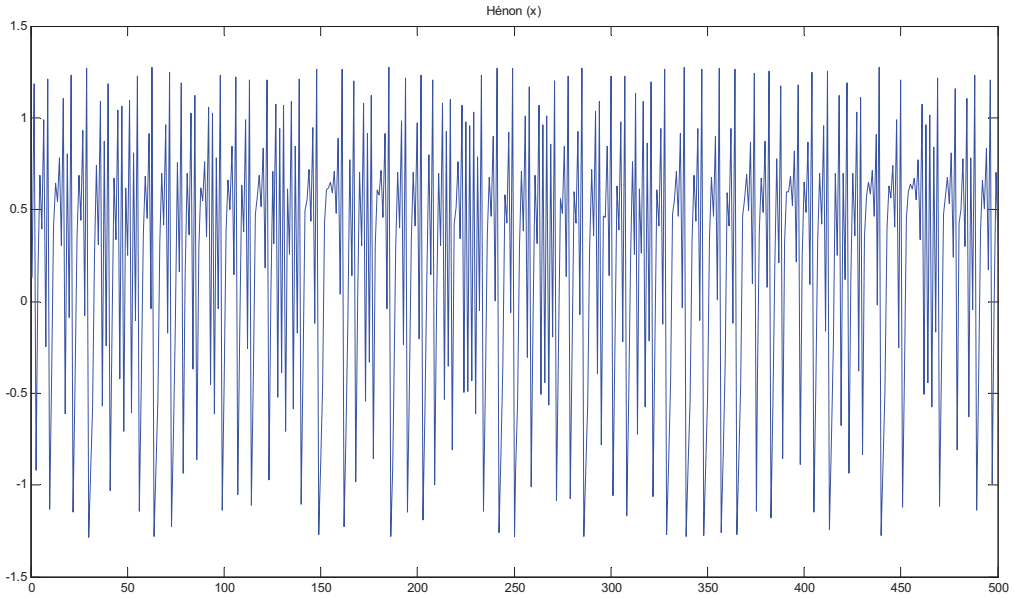


Figura 4.10. Serie del mapa de Hénon.

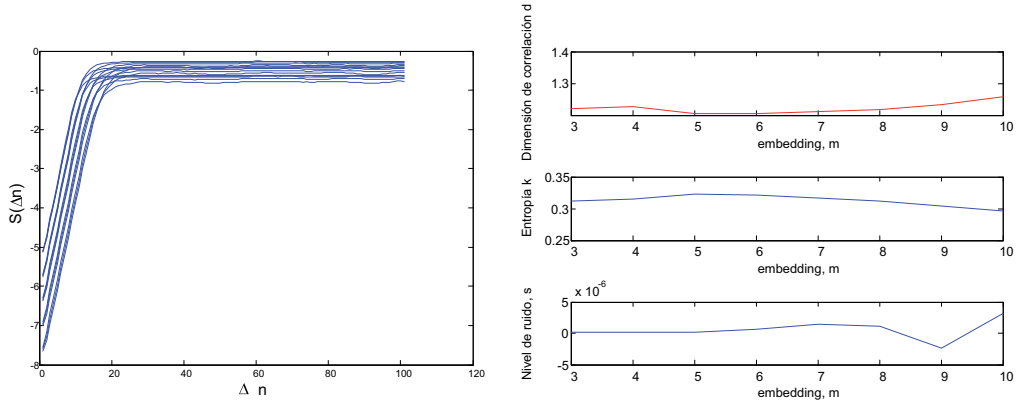


Figura 4.11. Estimación de invariantes para la serie Hénon.

Los 3000 primeros datos se usan para entrenamiento y el resto para validación. Los invariantes dinámicos se han estimado mediante una serie de 8000 muestras, con los algoritmos previamente especificados, y los resultados obtenidos también coinciden en gran medida con las estimaciones más comunes en la bibliografía (Wolf *et al.*, 1985; Grassberger y Procaccia, 1983). Estos valores calculados son $\lambda_1 = 0,440 \pm 0,003$ y $D_2 = 1,22 \pm 0,01$ (ver Fig. 4.11).

Se comparará la capacidad de reconstrucción de la dinámica de estas series por parte de los algoritmos descritos en el apartado 3.2, teniendo en cuenta tanto el error de predicción como la preservación de los invariantes dinámicos.

Datos reales de sistema físico Es apropiado también extender la verificación de los algoritmos a señales reales, para mostrar si los algoritmos proporcionan un escenario de modelado robusto que se pueda aplicar a señales con características no lineales diversas. Para generar una señal de prueba, se ha utilizado la maqueta industrial de 4 variables (ver 4.2.1), sobre la que se ha implementado una estrategia de control de nivel que se repite periódicamente mientras se simula la progresiva obturación de la válvula de proceso, lo que provoca cambios en la evolución de la variable presión.

La evolución de la variable presión, que se muestra en la Fig. 4.12, se toma como serie autónoma objeto de prueba, ignorando las entradas al sistema. Se trabaja con 2750 datos de entrenamiento y 200 de validación.

Se calcula el error de predicción obtenido por medio de los algoritmos descritos en el apartado 3.2.

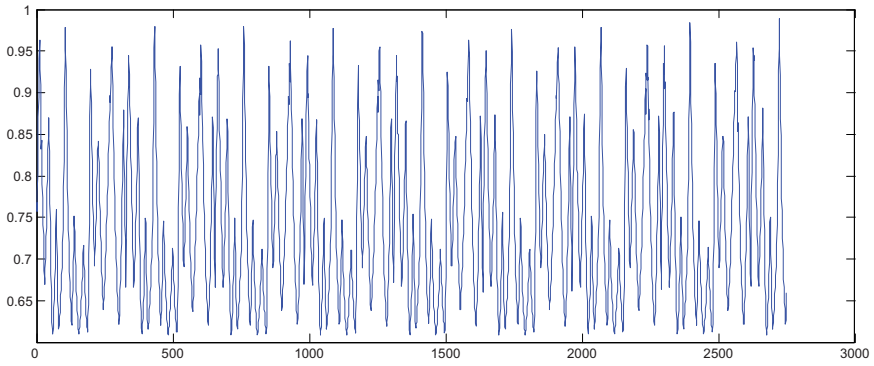


Figura 4.12. Serie de la evolución de la variable presión.

4.5. Experimentos de visualización en sistemas SISO

4.5.1. Introducción y objetivo

El objetivo de los experimentos es determinar si los mapas de dinámica previamente definidos permiten al supervisor o analista **extraer información relevante** acerca de la dinámica de un proceso real, inferir correlaciones o distinguir regiones diferenciadas. Se aprovecha el conocimiento previo acerca del sistema, formal o intuitivo, para valorar los conocimientos que puedan extraerse mediante la exploración visual de los mapas de dinámica. La evaluación se basará, por lo tanto, en determinar si los mapas generados son informativos y coherentes.

Concretamente, estos experimentos se centrarán en los mapas propuestos en el apartado 3.3.3, que representan características útiles en el análisis de la respuesta de sistemas: ganancia, tiempo de establecimiento, tiempo de pico, sobreoscilación, tiempo de subida, coeficiente de amortiguamiento y frecuencia natural.

Para este propósito, se llevan a cabo dos experimentos: el primero genera mapas de dinámica a partir de los valores obtenidos de un modelo matemático, mientras que el segundo trabaja con datos muestreados de un sistema real. El objetivo es validar el método tanto ante modelos matemáticos como ante procesos industriales reales. Ambos experimentos están muy relacionados, puesto que el banco de pruebas del primer experimento es una simulación de un modelo simplificado del sistema real que constituye el banco de pruebas del segundo experimento.

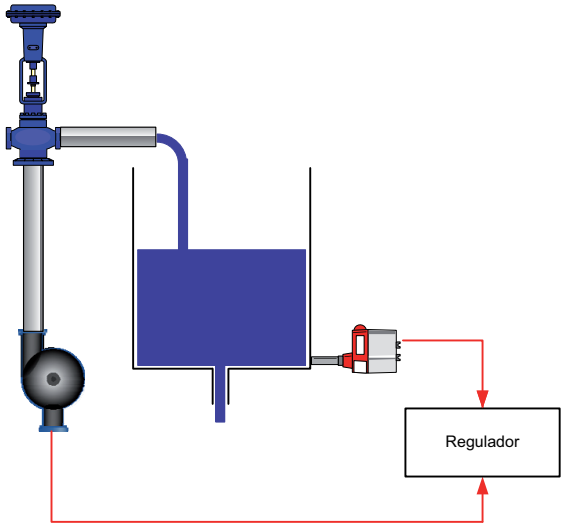


Figura 4.13. Modelo de control de nivel.

4.5.2. Experimento SISO1: Mapas de visualización para control de nivel simulado

El objeto de las pruebas en este experimento es un modelo de un sistema de llenado de un depósito en lazo cerrado, cuya configuración se representa de forma esquemática en la figura 4.13. Como datos de entrada al SOM se utilizan 480 puntos de equilibrio, que se obtienen mediante las combinaciones entre 24 niveles y 20 aperturas de válvula, y sus respectivos comportamientos dinámicos, los cuales se obtienen directamente a partir de las ecuaciones. El conjunto de entrada a la red tiene dimensión 7, pues incluye, como selectores de la dinámica, las variables válvula y nivel y, como parámetros dinámicos, los coeficientes de una función de transferencia de segundo orden con un cero.

Sean $q_e(t)$ y $q_s(t)$ los caudales de entrada y salida, $h(t)$ el nivel del depósito, A y a las áreas del depósito y de salida, $b(t)$ y k la velocidad y la constante de la bomba, k' y k_i las constantes del regulador yg la aceleración debida a la gravedad. Para obtener el modelo matemático se aplica balance de masas

$$A\frac{dh(t)}{dt} = q_e(t) - q_s(t) \tag{4.8}$$

y la ecuación de Bernoulli

$$A\frac{dh(t)}{dt} = k\gamma b(t) - a\sqrt{2gh(t)}. \tag{4.9}$$

El sistema se linealiza en torno al punto h_0

$$A\Delta\frac{dh}{dt} = k\gamma\Delta b - \frac{ag}{\sqrt{2gh_0}}\Delta h \tag{4.10}$$

y se obtiene la transformada de Laplace

$$sH(s) = \frac{k\gamma b(s)}{A} - \frac{ag}{A\sqrt{2gh_0}}H(s) \tag{4.11}$$

$$\frac{H(s)}{b(s)} = \frac{k\gamma}{As + a\sqrt{\frac{g}{2h_0}}}. \tag{4.12}$$

Se realiza un control en lazo cerrado de ese sistema, para el que se introduce un regulador proporcional integral ISA

$$r(t) = k'(e(t) + k_i\int e(t)dt + k_d\frac{de(t)}{dt}) \tag{4.13}$$

$$R(s) = (k' + \frac{k'k_i}{s} + k'k_ds)e(s) \tag{4.14}$$

$$\frac{R(s)}{e(s)} = \frac{k's + k'k_i}{s} \tag{4.15}$$

$$\frac{R(s)}{e(s)} = \frac{k'(s + k_i)}{s} \tag{4.16}$$

Así, la función de transferencia del sistema completo, incluyendo el regulador, corresponde a la expresión

$$G(s) = \frac{\frac{k'(s+k_i)k\gamma}{s(As+a\sqrt{\frac{g}{2h_0}})}}{1 + \frac{k'(s+k_i)k\gamma}{s(As+a\sqrt{\frac{g}{2h_0}})}} \tag{4.17}$$

$$G(s) = \frac{k'k\gamma(s + k_i)}{s(As + a\sqrt{\frac{g}{2h_0}}) + k'k\gamma(s + k_i)} \tag{4.18}$$

$$G(s) = \frac{k'k\gamma(s + k_i)}{As^2 + (a\sqrt{\frac{g}{2h_0}} + k'k\gamma)s + k'kk_i\gamma}. \tag{4.19}$$

El área del depósito es $0,04m^2$ y el de salida $0,001m^2$. La constante de la bomba es $4,55 \cdot 10^{-6} m^3/sV$. En el experimento se visualizan las características de la respuesta en el tiempo (ganancia, tiempo de establecimiento, tiempo de pico, sobreoscilación, tiempo de subida, coeficiente de amortiguamiento y frecuencia natural). Los resultados de este experimento se presentan en el apartado 5.2.1.

4.5.3. Experimento SISO2: Mapas de visualización para control de nivel real

En este experimento se implementa una estrategia de llenado del depósito superior del circuito de proceso de la maqueta de cuatro variables, descrita en la sección 4.2.1. En la Fig. 4.14 se muestra el circuito que se utiliza para este propósito. El modelo teórico presentado para el primer experimento es un modelo simplificado de este sistema.

Figura 4.14. Circuito principal de la maqueta industrial de 4 variables.

La variable a controlar es el nivel, medido por medio de un sensor de ultrasonidos, y se considera como entrada la velocidad en porcentaje del variador de frecuencia de la bomba. El depósito de agua en el que se controla el nivel tiene una capacidad de 10 litros y una capacitancia muy pequeña que provoca que sea difícil mantener el nivel del mismo, ya que la electroválvula de desagüe permanece abierta durante todo el experimento para permitir el vaciado del depósito. El diámetro de las tuberías de aporte y desagüe es de $25mm$. Por tanto, además de un rápido vaciado, se originan turbulencias. Una válvula de aporte neumática modifica el caudal de entrada.

Para su estabilización, se cierra el lazo en torno al depósito y se incluye un regulador ISA proporcional integral, al igual que en el experimento anterior.

Puesto que el objetivo es la obtención de modelos válidos en el entorno de varios puntos de funcionamiento, que se pueden obtener excitando el sistema para diferentes valores iniciales del nivel y de la apertura de la válvula de aporte, se lleva a cabo una identificación en escalera o por tramos de funcionamiento (ver Fig. 4.15). Para ello, el punto de equilibrio del nivel en torno al que varía la consigna se incrementa desde el 20 % al 95 % en escalones del 15 % y, posteriormente, se decrementa en la misma magnitud. Para obtener una excitación persistente en cada tramo, una vez estabilizado el nivel en el punto de equilibrio, se utiliza una secuencia binaria pseudoaleatoria (señal PRBS) con niveles $+2,5$ y $-2,5$ (Ljung, 1999). Las señales binarias pseudoaleatorias son señales periódicas y deterministas con propiedades parecidas al ruido blanco. Este proceso se repite para diversos valores de apertura de la válvula.

Haciendo uso de la estructura de comunicaciones previamente descrita, se adquieren los datos con un periodo de muestreo de $125ms$ y se almacenan en la base de datos, de donde pueden ser posteriormente recuperados para su análisis. El preprocesamiento previo a la identificación incluye la eliminación de datos espurios y la resta del valor medio. En cada tramo, los datos se dividen en un conjunto de trabajo y otro de validación, de tal manera que el primero se utiliza para estimar los parámetros de la identificación y el segundo permite calcular el ajuste del modelo.

Para obtener un modelo del proceso entre la consigna y la salida, se lleva a cabo, en cada tramo, una identificación directa de la función de transferencia continua, mediante un método de estimación de parámetros basado en la minimización del error de predicción y disponible en Matlab (Ljung, 2002), que aprovecha el conocimiento a priori que se posee acerca del sistema.

El resultado de la identificación es un conjunto de modelos (60) cuyas funciones de transferencia son subamortiguadas con dos polos y un cero y cuyo ajuste a los datos de validación resulta aceptable. El ajuste, que se define como

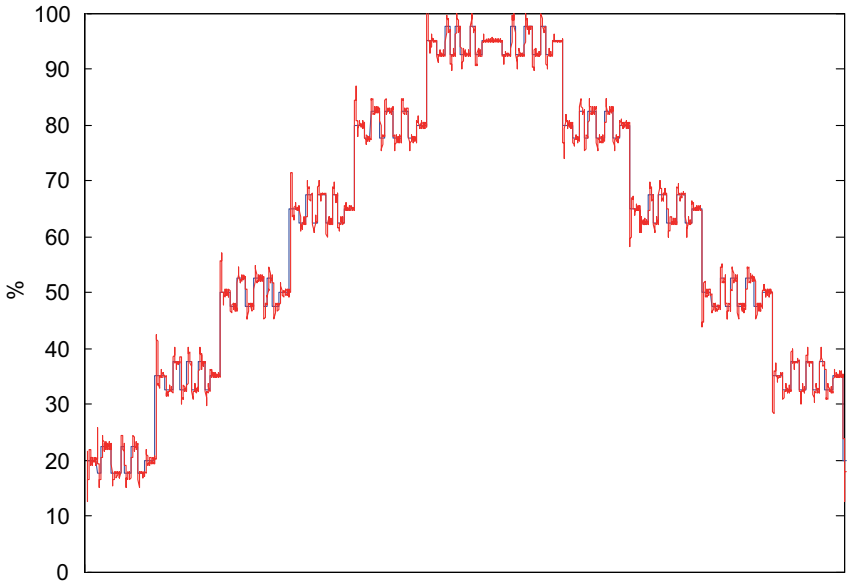


Figura 4.15. Fragmento de las señales consigna y nivel utilizadas en la identificación.

$$fit(\,\%) = 1 - \frac{\|y - \widehat{y}\|}{\|y - \overline{y}\|} \cdot 100 \tag{4.20}$$

donde $\widehat{y}$ es la salida identificada por el modelo y $\overline{y}$ es la media, tiene un valor medio de $86,51\,\%$.

El conjunto de datos con el que se trabaja consta, por tanto, de 60 vectores de 7 dimensiones (nivel, apertura de válvula y coeficientes de la función de transferencia en el entorno del punto de equilibrio). De nuevo, en el experimento se visualizan las características de respuesta en el tiempo. Los resultados de este experimento se presentan en el apartado 5.2.2

4.6. Experimientos de visualización en sistemas MIMO

4.6.1. Introducción y objetivo

El sistema seleccionado para estos experimentos es la maqueta de 4 tanques que fue descrita al principio de este capítulo, ya que se trata de un sistema muy apropiado para validar la técnica de los mapas de dinámica como método de apoyo

al análisis exploratorio de sistemas multivariable. El motivo es que el modelo del proceso de cuatro tanques fue definido para ilustrar conceptos de control multivariable en la enseñanza de ingeniería de control, especialmente las limitaciones de rendimiento provocadas por ceros multivariables en el semiplano derecho.

El sistema cuenta con un cero cuya posición y dirección tienen una interpretación física directa e ilustrativa y puede estar localizado tanto en el semiplano derecho como en el izquierdo para diferentes valores de las válvulas. Esto es especialmente útil, pues permite relacionar de forma directa los valores de las variables con los comportamientos en los diversos puntos de funcionamiento. Si a esto añadimos que el modelo de cuatro tanques ha sido ampliamente estudiado en la literatura, se pone de manifiesto la utilidad de este sistema para el propósito que nos ocupa.

Se definen dos experimentos para validar los mapas de dinámica de sistemas multivariable. En primer lugar, se aplican a un sistema simulado, obtenido a partir de las ecuaciones del modelo. En segundo lugar, se trabaja con modelos identificados del sistema industrial real. Ambos experimentos siguen una estrategia similar, que se describe en los respectivos apartados.

Los objetivos de ambos experimentos son la obtención de mapas de la dinámica que muestren información significativa acerca de las posiciones y direcciones de los ceros multivariable, de los mejores emparejamientos posibles vía RGA, de las ganancias y sus direcciones y del condicionamiento de la matriz. Se aprovecha el amplio conocimiento que existe acerca del sistema físico para poder validar los resultados y se establecen como condiciones para considerarlos satisfactorios:

- Poder extraer correspondencias que correlacionen características dinámicas entre sí o con variables del sistema. En general, debería ser posible visualizar regiones claramente diferenciables.

- Que las mismas coincidan con el conocimiento previo que se posee del sistema o que sean inferibles a partir del mismo.

4.6.2. Experimento MIMO1: Mapas de visualización para el modelo de 4 tanques

En este experimento se trabaja con el modelo matemático de la maqueta de 4 tanques, presentado en la sección 4.2.2 y en Johansson (2000), que relaciona 2 entradas (las bombas u_1 y u_2) con 2 salidas (los niveles de los tanques inferiores $y1$ e $y2$). Se calculan 3240 puntos de funcionamiento y sus respectivos comportamientos dinámicos, combinando diferentes niveles y posiciones de las válvulas, los cuales constituyen el conjunto de datos de entrada. Los vectores $\mathbf{q}$ del espacio de entrada

tienen 18 dimensiones, ya que las 4 primeras corresponden a un punto de operación (apertura de las válvulas y niveles de los 2 tanques inferiores) y las restantes son los parámetros que definen el modelo. En este caso, el modelo paramétrico considerado es la matriz 2×2 de funciones de transferencia del sistema. Así pues, de acuerdo a la ecuación 4.3, los vectores de entrada se definen de la siguiente manera:

$$\mathbf{q} = [\mathbf{p}^T, \mathbf{x}^T]^T = [\gamma_1, \gamma_2, h_1, h_2, \gamma_1 c_1, 1, T_1, (1-\gamma_2)c_1, 1, (T_3+T_1), \tag{4.21}$$

$$T_3 T_1, (1-\gamma_1)c_2, 1, (T_4+T_2), T_4 T_2, \gamma_2 c_2, 1, T_2] \tag{4.22}$$

En este caso, se tiene interés en visualizar información útil para seleccionar los mejores emparejamientos para un control descentralizado, localizar regiones de funcionamiento en las que el control sea complicado, analizar la direccionalidad del sistema y descubrir relaciones entre los selectores de dinámica y sus comportamientos asociados. Concretamente, se visualizan los siguientes mapas:

- Mapa de modelo de $\gamma_1+\gamma_2$, pues el conocimiento previo sugiere que determina el comportamiento dinámico de la planta.
- Mapa de ganancia relativa en régimen permanente, para visualizar el acoplamiento entre variables y las zonas problemáticas.
- Mapa de valores y direcciones singulares, en régimen permanente, para analizar los valores y direcciones de máxima y mínima ganancia y sugerir el modo de desacoplar la matriz.
- Mapa de condición, para visualizar zonas con problemas para el control
- Mapa de ceros multivariable, para revelar dónde el sistema es de fase no mínima.

Los mapas se construyen mediante el procedimiento previamente descrito y los resultados del experimento se presentan en el apartado 5.3.1.

4.6.3. Experimento MIMO2: Mapas de visualización para un sistema real de 4 tanques

En este experimento, al igual que en el anterior, se relacionan 2 entradas (las bombas u_1 y u_2) con 2 salidas (los niveles de los tanques inferiores y_1 e y_2), pero los comportamientos dinámicos del sistema en cada punto de equilibrio no proceden de sus ecuaciones sino que son identificados a partir de los datos adquiridos de la operación real de la maqueta descrita en el apartado 4.2.2.

Se pretende obtener modelos válidos en el entorno de varios puntos de funcionamiento, excitando el sistema para diferentes valores iniciales de los niveles y de las válvulas. Para ello, se implementa una estrategia en escalera en la que a cada tramo se introduce una señal PRBS con niveles $+2,5\,\%$ y $-2,5\,\%$. Los diferentes puntos de equilibrio se consiguen variando cada nivel en escalones del $15\,\%$. En cada ciclo también se incrementa simultáneamente el valor de ambas válvulas en un $20\,\%$.

Para que el sistema sea estable, es necesario proceder del siguiente modo: el experimento se realiza en lazo cerrado y con un regulador proporcional y se cruza la relación entrada-salida en el control para los casos en los que sea necesario.

Se adquieren los datos con un periodo de muestreo de $125ms$ y se almacenan en la base de datos, de donde son recuperados para su análisis en Matlab. Se lleva a cabo un diezmado de la señal $10:1$, por lo que la frecuencia de muestreo final es $0,8Hz$. Se realiza un preprocesamiento previo a la identificación que incluye la eliminación de datos espurios y la resta del valor medio. En cada tramo, los datos se dividen en un conjunto de trabajo y otro de validación, de tal manera que el primero se utiliza para estimar los parámetros de la identificación y el segundo para calcular el ajuste del modelo.

En este caso, se realiza una identificación directa del sistema a partir de las medidas de entrada y y de salida u (Ljung, 2002), en la que se ajustan modelos ARX (autorregresivos con entrada exógena) de orden $(3,3)$. Estos modelos se pueden expresar en forma de funciones de transferencia en tiempo discreto.

El resultado de la identificación proporciona un ajuste medio de $82,74\,\%$, sin considerar 60 casos, que se omiten del conjunto de entrada del SOM porque su ajuste no es suficientemente satisfactorio.

Finalmente se obtiene un conjunto de datos de entrada al SOM con 93 vectores de 32 dimensiones (apertura de válvulas, niveles de los tanques inferiores y coeficientes de la matriz de funciones de transferencia en tiempo discreto).

Los objetivos de este experimento son los mismos que los del experimento anterior. Se generarán los mismos mapas de dinámica que en el experimento anterior, con el método general, salvo por los relativos a los ceros y sus direcciones, puesto que al trabajar en tiempo discreto y con diferente orden, la equivalencia no es directa. Los resultados se presentan en el apartado 5.3.2.

Capítulo 5

Resultados experimentales

5.1. Comparación de métodos de modelado local de la dinámica basados en SOM

5.1.1. Series caóticas

A continuación se presentan tanto el desarrollo de los experimentos como los resultados de la comparación de los algoritmos SOM, SOM con aprendizaje dinámico, SOM recurrente, SARDNET, *Merge* SOM y SOMTAD y de la estimación de modelos por medio de los vectores prototipo correspondientes a la vecindad de la neurona ganadora y por medio de los vectores de entrada asociados a la región de Voronoi de la neurona ganadora. Como se indicó anteriormente, el objetivo es evaluar la calidad de la reconstrucción de la dinámica de dos series caóticas, ampliamente utilizadas como banco de pruebas para este propósito y otros relacionados con la predicción de series temporales.

Los algoritmos que se comparan en este apartado han sido implementados en Matlab, utilizando funciones de dos implementaciones previas del SOM, como son la *SOM Toolbox*[1], desarrollada por la Universidad de Helsinki (Vesanto *et al.*, 1999), y la librería UniOviSOM, desarrollada por la Universidad de Oviedo. Para evitar diferencias que perturben la comparación, se han utilizado los mismos criterios en el desarrollo de cada algoritmo.

Como características comunes al entrenamiento de todos los modelos, se puede destacar el uso de un entrenamiento secuencial en todos los casos, en el que la vecindad es gaussiana y disminuye, al igual que la tasa de aprendizaje, de forma gradual durante el entrenamiento. En cuanto a los algoritmos de estimación de

[1] http://www.cis.hut.fi/projects/somtoolbox/

Parámetros de los algoritmos								
SOM								
	Dimensiones	**Retardos**	**Vec. Estimación**		**Épocas**	**Vecindad Inicial**	**Vecindad Final**	
MG17	25x25, 35x35	4,5,6	1; $\sqrt{2}$; $\sqrt{3}$		200	10	0,5	
Hénon	25x25, 35x35	2,3,4	1; $\sqrt{2}$; $\sqrt{3}$		200	10	0,5	
SOM DL								
	Dimensiones	**Retardos**	**Vec. Estim.**	**Épocas**	**V. Inicial**	**Vec. Final**	**μ**	
MG17	25x25, 35x35	4,5,6	1; $\sqrt{2}$; $\sqrt{3}$	200	10	0,5	1,5; 1,75; … 2,75	
Hénon	25x25, 35x35	2,3,4	1; $\sqrt{2}$; $\sqrt{3}$	200	10	0,5	1,5; 1,75; … 2,75	
RSOM								
	Dimensiones	**Retardos**	**Épocas**	**Vec. Inicial**	**Vec. Final**	**α**		
MG17	25x25, 35x35	4,5,6	200	10	0,5	0,35; 0,4; 0,45; 0,625; 0,73; 0,78; 0,95		
Hénon	25x25, 35x35	2,3,4	200	10	0,5	0,35; 0,4; 0,45; 0,625; 0,73; 0,78; 0,95		
SARDNET								
	Dimensiones	**Retardos**	**Vec. Estim.**	**Épocas**	**V. Inicial**	**Vec. Final**	***d***	
MG17	25x25, 35x35	4,5,6	1; $\sqrt{2}$; $\sqrt{3}$	200	10	0,5	0,1; 0,2; … 0,9	
Hénon	25x25, 35x35	2,3,4	1; $\sqrt{2}$; $\sqrt{3}$	200	10	0,5	0,1; 0,2; … 0,9	
MSOM								
	Dims.	**Retardos**	**V. Estim.**	**Épocas**	**V. Inicial**	**V. Final**	**α**	**β**
MG17	25x25, 35x35	4,5,6	1; $\sqrt{2}$; $\sqrt{3}$	200	10	0,5	0,1; 0,2; … 0,9	0,1; 0,2; … 0,9
Hénon	25x25, 35x35	2,3,4	1; $\sqrt{2}$; $\sqrt{3}$	200	10	0,5	0,1; 0,2; … 0,9	0,1; 0,2; … 0,9
SOMTAD								
	Dims.	**Retardos**	**V. Estim.**	**Épocas**	**V. Inicial**	**V. Final**	**μ**	**β**
MG17	25x25, 35x35	4,5,6	1; $\sqrt{2}$; $\sqrt{3}$	200	10	0,5	0,1; 0,2; … 0,9	0,1; 0,2; … 0,9
Hénon	25x25, 35x35	2,3,4	1; $\sqrt{2}$; $\sqrt{3}$	200	10	0,5	0,1; 0,2; … 0,9	0,1; 0,2; … 0,9

Tabla 5.1. Selección de parámetros de los diversos algoritmos.

modelos, se lleva a cabo una regresión lineal mediante mínimos cuadrados regularizados y ponderados en función de la distancia al vector prototipo de la BMU.

Respecto a la selección de los modelos, tal y como se indicó en la descripción de los experimentos, los parámetros libres de los algoritmos se seleccionan bien a partir del conocimiento previo, bien por medio de una validación cruzada multipaso. Para evitar un proceso de validación muy prolongado, que debe repetirse para cada serie y algoritmo, este proceso se realiza sobre un conjunto limitado de parámetros que, no obstante, incluye los parámetros más críticos y sus valores representativos dentro del rango adecuado. Concretamente, los parámetros que se incluyen en la validación cruzada son aquellos que determinan el contexto en cada modificación del algoritmo SOM, el número de retardos, las dimensiones de la red y, en la estimación por medio de vectores prototipo, el tamaño de la vecindad

que se utiliza sobre el SOM entrenado para estimar el modelo. Otros parámetros relevantes como las épocas, la vecindad de entrenamiento o la tasa de aprendizaje se fijan de antemano, de acuerdo a lo observado en las pruebas preliminares, dentro de los límites que recomiendan las reglas heurísticas. El número de retardos se escoge en el entorno de la dimensión sugerida por el método de índices de Lipschitz de He y Asada (1993)[2]. Este método sugiere un *embedding* de 4 para la serie de Mackey-Glass 17 y 3 para la serie de Hénon. De acuerdo a todas estas consideraciones, finalmente se utilizan los parámetros presentados en la Tabla 5.1.

En cada iteración de la validación cruzada, el algoritmo se entrena con una serie de 3000 datos. Los cálculos del error se realizan, de acuerdo a lo presentado en la sección 4.4.1, considerando los horizontes 14 y 8 para, respectivamente, MG17 y Hénon. En la Tabla 5.2 se presentan los parámetros seleccionados, de entre los anteriores, por medio de validación cruzada.

Para la evaluación final de cada algoritmo frente a cada una de las series, el error NRMSE de predicción multipaso se calcula sobre 500 muestras que no han sido previamente utilizadas en el entrenamiento. Por otra parte, para la estimación de los invariantes dinámicos, se generan series artificiales con cada algoritmo mediante una predicción recursiva de 8000 pasos adelante a partir de la primera muestra de la serie de entrenamiento. Posteriormente, se aplica el algoritmo de Kantz-Rosenstein para estimar el mayor exponente de Lyapunov y el GKA para estimar la dimensión de correlación, al igual que se hizo con la serie original.

Una vez presentadas las particularidades de las pruebas, se puede proceder a la presentación y análisis de los resultados.

Veamos, en primer lugar, las curvas del error de predicción hasta el horizonte definido para la serie Mackey-Glass 17 (ver Figs. 5.1 y 5.2). Comparando ambas gráficas se puede observar cómo la estimación por medio de los datos de la región de Voronoi supera claramente a la estimación por medio de los vectores prototipo. Esto era previsible, ya que en el primer caso se cuenta con más información que en el ajuste por medio de vectores prototipo, el cual, por otra parte, puede verse más perturbado si la topología no se preserva de forma correcta.

En cuanto a las modificaciones del algoritmo SOM, cabe destacar el buen rendimiento en predicción a largo plazo del SOM con aprendizaje dinámico, en el ajuste por medio de vectores prototipo, y el MSOM, en el ajuste por medio de vectores de entrada. El algoritmo SOMTAD también muestra mejores resultados que el SOM con ambos ajustes, mientras que los algoritmos SARDNET y RSOM también superan al algoritmo clásico con el segundo ajuste. La adecuación del algoritmo SOMDL es razonable pues su algoritmo está optimizado para este propósito, como ya se puso de manifiesto en Principe *et al.* (1998). El resto de modificaciones

[2]Para lo cual se utiliza la implementación de la *toolbox* NNSYSID (Nørgaard, 2000).

Parámetros finalmente seleccionados para cada algoritmo													
SOM													
Ajuste mediante vectores prototipo							**Ajuste mediante vectores de region de Voronoi**						
	Dims	**d_e**	**σ**	**Ep**	**σ_{h1}**	**σ_{hn}**	**Dims**	**d_e**	**Ep**	**σ_{h1}**	**σ_{hn}**		
MG17	35x35	6	$\sqrt{2}$	200	10	0,5	35x35	6	200	10	0,5		
Hénon	35x35	3	1	200	10	0,5	35x35	3	200	10	0,5		
SOM DL													
Ajuste mediante vectores prototipo								**Ajuste mediante vectores de region de Voronoi**					
	Dims	**d_e**	**σ**	**Ep**	**σ_{h1}**	**σ_{hn}**	**μ**	**Dims**	**d_e**	**Ep**	**σ_{h1}**	**σ_{hn}**	**μ**
MG17	35x35	6	$\sqrt{2}$	200	10	0,5	1,5	35x35	6	200	10	0,5	2
Hénon	35x35	4	1	200	10	0,5	2,5	35x35	4	200	10	0,5	2,5
SARDNET													
Ajuste mediante vectores prototipo								**Ajuste mediante vectores de region de Voronoi**					
	Dims	**d_e**	**σ**	**Ep**	**σ_{h1}**	**σ_{hn}**	***d***	**Dims**	**d_e**	**Ep**	**σ_{h1}**	**σ_{hn}**	***d***
MG17	35x35	6	$\sqrt{2}$	200	10	0,5	0,1	35x35	6	200	10	0,5	0,1
Hénon	35x35	4	1	200	10	0,5	0,9	25x25	3	200	10	0,5	0,9
MSOM													
Ajuste mediante vectores prototipo									**Ajuste mediante vectores de region de Voronoi**						
	Dims	**d_e**	**σ**	**Ep**	**σ_{h1}**	**σ_{hn}**	**α**	**β**	**Dims**	**d_e**	**Ep**	**σ_{h1}**	**σ_{hn}**	**α**	**β**
MG17	35x35	6	$\sqrt{2}$	200	10	0,5	0,2	0,9	35x35	6	200	10	0,5	0,5	0,5
Hénon	35x35	4	1	200	10	0,5	0,5	0,6	35x35	3	200	10	0,5	0,4	0,3
SOMTAD															
Ajuste mediante vectores prototipo									**Ajuste mediante vectores de region de Voronoi**						
	Dims	**d_e**	**σ**	**Ep**	**σ_{h1}**	**σ_{h1n}**	**μ**	**β**	**Dims**	**d_e**	**Ep**	**σ_{h1}**	**σ_{hn}**	**μ**	**β**
MG17	35x35	6	$\sqrt{2}$	200	10	0,5	0,1	0,2	35x35	6	200	10	0,5	0,4	0,2
Hénon	35x35	3	1	200	10	0,5	0,5	0,2	35x35	3	200	10	0,5	0,4	0,2
RSOM															
Ajuste mediante vectores de region de Voronoi															
	Dims	**d_e**	**Ep**	**σ_{h1}**	**σ_{hn}**	**α**									
MG17	35x35	6	200	10	0,5	0,95									
Hénon	35x35	3	200	10	0,5	0,95									

Tabla 5.2. Parámetros seleccionados por la validación cruzada multi-paso.

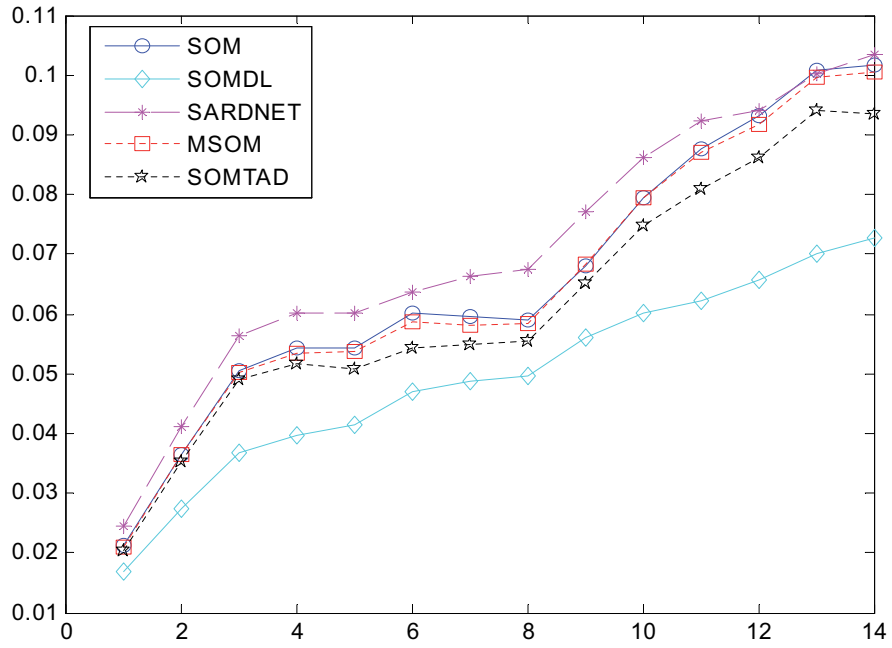


Figura 5.1. Curvas de error RNMSE de todos los algoritmos para MG17. Estimación de modelos por medio de vectores prototipo.

del algoritmo, en general, se ven beneficiadas en sus resultados por la información de contexto que incluyen. Los errores 14 pasos adelante de cada algoritmo ante esta serie se muestran también en la siguiente tabla:

	SOM	SOMDL	RSOM	SARD	MSOM	SOMTAD
NRMSE(14) Ajuste con región Voronoi	0,0431	0,0470	0,0404	0,0399	0,0344	0,0396
NRMSE(14) Ajuste con vectores prototipo	0,1018	0,0727	–	0,1034	0,1004	0,0935

Tabla 5.3. Errores de predicción 14 pasos adelante para MG17.

Los resultados obtenidos por los algoritmos con ajuste mediante vectores de la región de Voronoi son similares a los obtenidos previamente con esta estrategia (Vesanto, 1997) y también comparables con los resultados previos recogidos en la bibliografía, algunos de los cuales se resumen en el citado artículo, salvo con las redes *echo state* (Jaeger, 2004), que superan al resto cualitativamente (ver Fig. 5.4). Sin embargo, estas redes, que se valen de una memoria a corto plazo masiva basada en conexiones recurrentes generadas de forma aleatoria, carecen de la capacidad explicativa que posee la estrategia seguida en esta tesis.

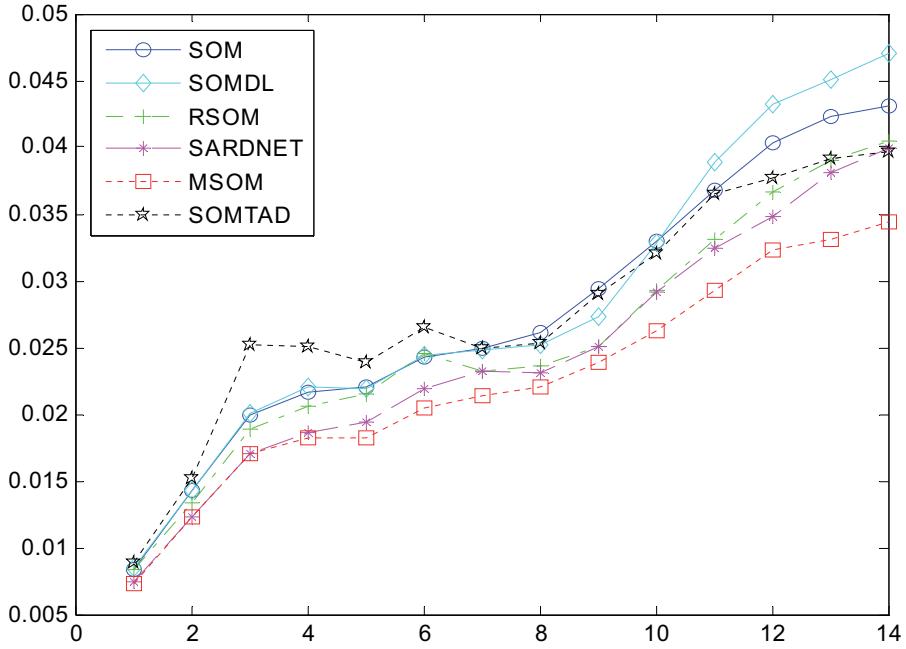


Figura 5.2. Curvas de error RNMSE de todos los algoritmos para MG17. Estimación de modelos por medio de datos de la región de Voronoi.

Métodos	NRMSE(14)
Echo State Networks	0,00006
SOM (35x35)	0,032
TDNN	0,04
DCA+LLM	0,043
RAN	0,054

Tabla 5.4. Errores de predicción 14 pasos adelante para MG17 con diversos algoritmos. Fuente: Vesanto (1997) y Jaeger (2004).

No obstante, hasta el momento solamente se ha analizado el error de predicción y es necesario conocer también en qué medida preservan los invariantes las series reconstruidas por los algoritmos objeto de prueba. Por ello, se muestran a continuación las estimaciones obtenidas para cada uno de ellos, utilizando los mismos algoritmos y criterios que para la estimación de los invariantes de la serie real:

		Ajuste mediante vectores de la region de Voronoi					
	Serie	**SOM**	**SOMDL**	**SARD**	**MSOM**	**SOMTAD**	**RSOM**
λ_1	**0,0064±0,0014**	0,0053±0,0019	0,0062±0,0014	0,006±0,0015	0,0071±0,0007	0,0059±0,0021	0,0052±0,0004
D_2	**1,95±0,01**	2,02±0,01	2,05±0,04	1,99±0,03	2,14±0,07	1,93±0,01	2,00±0,02
	Ajuste mediante vectores prototipo						
	Serie	**SOM**	**SOMDL**	**SARD**	**MSOM**	**SOMTAD**	
λ_1	**0,0064±0,0014**	0,0073±0,001	0,0069±0,0015	0,0069±0,0014	0,0061±0,0015	0,0072±0,005	
D_2	**1,95±0,01**	1,99±0,07	2,04±0,07	2,13±0,13	2,03±0,08	2,11±0,09	

Tabla 5.5. Invariantes dinámicos estimados a partir de las series MG17 reconstruidas.

De forma general, se puede observar que todos los algoritmos han sido capaces de reconstruir satisfactoriamente la dinámica original, pues sus invariantes estimados se aproximan a los estimados para la serie real. A la vista de los datos, no obstante, es difícil distinguir qué algoritmos muestran una estimación más aproximada tanto del mayor exponente de Lyapunov como de la dimensión de correlación. Para apoyar este análisis, se puede utilizar una simple representación visual, que muestre los rangos estimados para cada invariante, como en la Fig. 5.3. La gráfica muestra que el grado de coincidencia de los rangos correspondientes al máximo exponente de Lyapunov es muy alto, lo que sugiere que todos los algoritmos capturan el carácter caótico de esta serie. También sugiere que la estimación de la dimensión de correlación es menos precisa en los algoritmos con ajuste por medio de vectores prototipo, pese a que se utilizan los mismos criterios, ya que los rangos son más amplios.

Pese a que a la vista de las gráficas se podría llegar a la conclusión de que los algoritmos SOMTAD y SARDNET con ajuste basado en vectores de entrada y los algoritmos SOM y MSOM con ajuste basado en vectores prototipo son los que proporcionan una mejor reconstrucción de la dinámica, no se puede determinar de forma tan concluyente debido a la limitada precisión de los métodos de estimación, como consecuencia de su naturaleza eminentemente gráfica. De igual manera, es necesario resaltar que los parámetros seleccionados en cada caso generan un modelo que proporciona la mejor generalización para el objetivo de predicción iterada un número fijo de pasos adelante de entre un conjunto limitado de opciones presentadas. Aunque éste se trate de un objetivo apropiado para la reconstrucción dinámica, el modelo generado no tiene que ser necesariamente el óptimo en cuanto

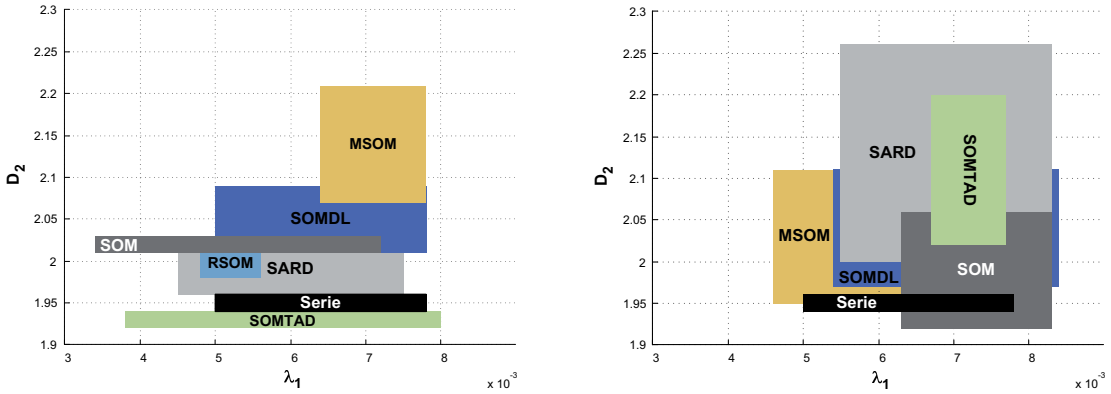


Figura 5.3. Rangos estimados para los invariantes de las series MG17 reconstruidas. A la izquierda, representación de rangos estimados para los algoritmos con ajuste por medio de vectores de la región de Voronoi. A la derecha, rangos estimados para los algoritmos ajustados por medio de vectores prototipo..

a la preservación de los invariantes. De acuerdo a esta consideración, cabe pensar que las estimaciones menos aproximadas de los algoritmos que lograron un menor NRMSE podrían estar causadas por un sobreajuste al objetivo del entrenamiento. Esta sospecha se ve confirmada al buscar el máximo horizonte de predicción de cada modelo sobre los datos de test, es decir, el número de pasos adelante hasta que el NRMSE es igual o mayor que 1. Los algoritmos con menor error de predicción para cada ajuste presentan un horizonte más corto, especialmente en el caso del MSOM con ajuste basado en vectores prototipo.

Los resultados correspondientes a la predicción de la segunda serie, el mapa de Hénon, se muestran en las figuras 5.4 y 5.5. De nuevo, los resultados revelan un mejor rendimiento del ajuste por medio de los datos de la región de Voronoi. También destaca, otra vez, el algoritmo SOMDL, mientras que los algoritmos MSOM, SOMTAD y SARDNET proporcionan buenos resultados en uno o ambos casos.

Considerando los resultados de predicción frente a ambas series, se puede observar cómo los algoritmos con contexto temporal superan, en general, al algoritmo SOM, si bien sus resultados presentan mucha variabilidad de un ajuste a otro. Esto se debe a que los algoritmos modificados cuentan con más parámetros libres que el SOM tradicional y, por tanto, son más difíciles de ajustar. Esos parámetros suelen recibir valores continuos entre 0 y 1 pero sus valores en estos experimentos solamente se han seleccionado de un limitado conjunto de posibilidades. La dependencia crítica en esos parámetros dificulta la obtención de buenos modelos y no es una característica deseable para el propósito de modelado dinámico. Pre-

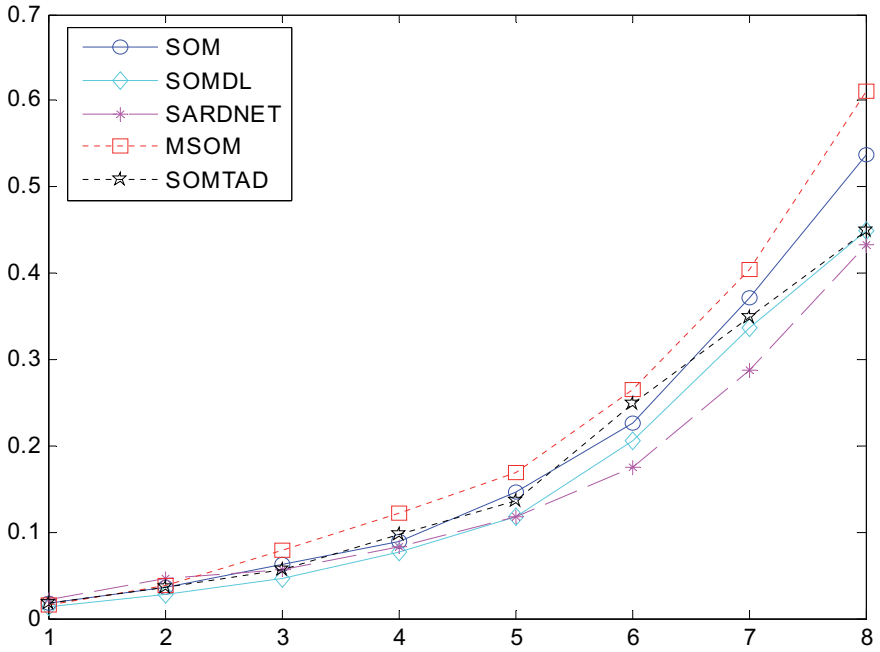


Figura 5.4. Curvas de error RNMSE de todos los algoritmos para Hénon. Estimación de modelos por medio de vectores prototipo.

cisamente, la dificultad en el ajuste del parámetro α en el algoritmo RSOM, ya detectada durante las pruebas preliminares y causada por su particular método de entrenamiento, mediante el que la red aprende episodios de una longitud fija determinada por α, causa el pobre rendimiento que muestra en este experimento.

Como en el experimento anterior, la siguiente tabla muestra los errores normalizados en el horizonte de predicción prefijado (8) para cada algoritmo y tipo de ajuste:

	SOM	SOMDL	RSOM	SARD	MSOM	SOMTAD
NRMSE(14) Ajuste con región Voronoi	0,1763	0,1164	0,359	0,1611	0,1293	0,1801
NRMSE(14) Ajuste con vectores prototipo	0,5371	0,4484	–	0,4330	0,6106	0,4496

Tabla 5.6. Errores de predicción 8 pasos adelante para Hénon.

Los invariantes dinámicos estimados en este caso muestran, en general, una buena aproximación a los de la serie original con el ajuste mediante vectores de la región de Voronoi, aunque menos precisa que en el experimento anterior. Esta peor aproximación puede deberse al comportamiento más caótico de la serie objeto

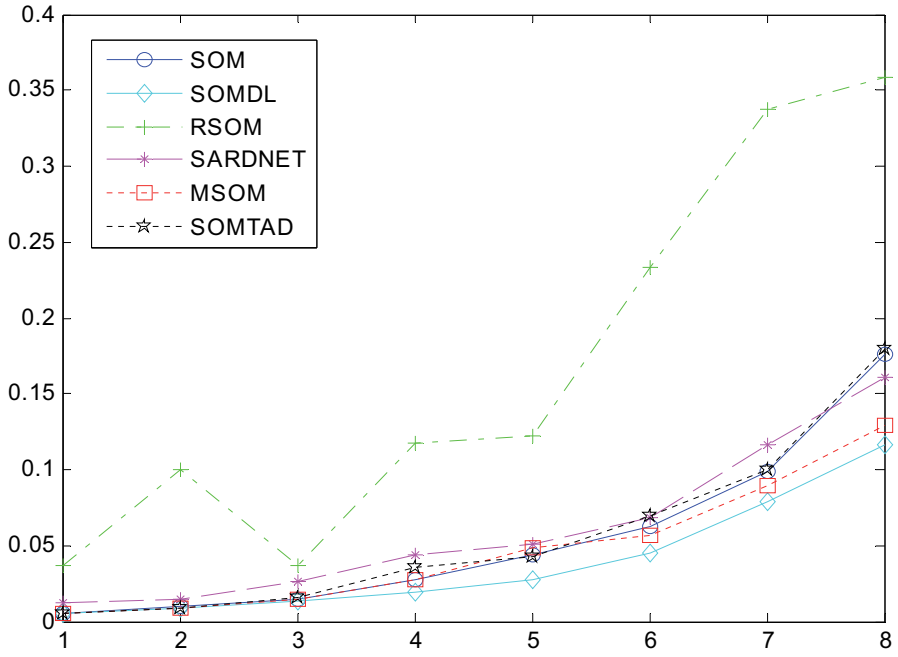


Figura 5.5. Curvas de error RNMSE de todos los algoritmos para Hénon. Estimación de modelos por medio de datos de la región de Voronoi.

de prueba, con un exponente de Lyapunov bastante más grande que el de la serie de Mackey-Glass y, por tanto, más difícil de procesar. Es necesario señalar que la reconstrucción por medio del RSOM y MSOM con el ajuste por medio de los vectores de la región de Voronoi y de SOMDL, MSOM y SARDNET con el ajuste por medio de vectores prototipo, no ha sido satisfactoria, pues la serie reconstruida diverge al cabo de un tiempo (en general, a partir de 2000 pasos adelante).

		Ajuste mediante vectores de la region de Voronoi					
	Serie	**SOM**	**SOMDL**	**RSOM**	**SARD**	**MSOM**	**SOMTAD**
λ_1	**0,429±0,021**	0,392±0,022	0,372±0,033	0,407±0,044	0,386±0,019	0,397±0,017	0,388±0,027
D_2	**1,22±0,01**	1,33±0,05	1,42±0,10	1,45±0,08	1,39±0,04	1,45±0,10	1,37±0,04
	Ajuste mediante vectores prototipo						
	Serie	**SOM**	**SOMDL**	**SARD**	**MSOM**	**SOMTAD**	
λ_1	**0,429±0,021**	0,383±0,029	0,413±0,041	0,387±0,046	0,389±0,03	0,384±0,028	
D_2	**1,22±0,01**	1,52±0,13	1,77±0,17	1,76±0,16	1,55±0,17	1,50±0,13	

Al igual que en el experimento anterior, se puede utilizar la representación visual (ver Fig. 5.6), para facilitar la interpretación de los datos. En este caso, puede observarse cómo la estimación de la dimensión de correlación por medio de los métodos con ajuste basado en vectores prototipo es claramente peor.

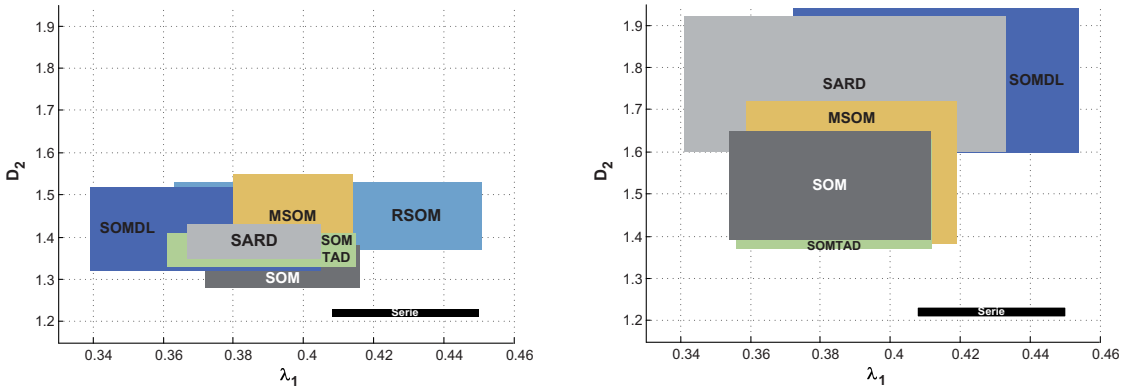


Figura 5.6. Rangos estimados para los invariantes de las series Hénon reconstruidas. A la izquierda, representación de rangos estimados para los algoritmos con ajuste por medio de vectores de la región de Voronoi. A la derecha, rangos estimados para los algoritmos ajustados por medio de vectores prototipo..

En resumen, los resultados de ambos experimentos indican que el ajuste por medio de vectores de las respectivas regiones de Voronoi proporciona predicciones más precisas y estimaciones de los invariantes generalmente mejores que el ajuste por medio de vectores prototipo. Las conclusiones con respecto a las modificaciones del algoritmo SOM tradicional no son tan definitivas, ya que la dependencia de estos algoritmos en sus parámetros adicionales provoca que una selección no óptima de los mismos, causada por el conjunto reducido de posibilidades utilizado, dé lugar a modelados comparables o peores que los proporcionados por el algoritmo clásico. No obstante, los algoritmos SOMDL, SOMTAD, SARDNET y MSOM proporcionan, mayoritariamente, mejores resultados que el algoritmo clásico en cuanto a error de predicción n pasos adelante (el objetivo utilizado para fijar los parámetros del entrenamiento), aunque en el mismo orden de magnitud. Si bien es cierto que todos ellos ofrecen estimaciones adecuadas para los invariantes cuando se utiliza un ajuste por medio de vectores de la región de Voronoi, el SOMTAD es el único algoritmo modificado cuya reconstrucción de la serie Hénon, por medio de alguno de los ajustes, no causa problemas de divergencia. Aunque el RSOM puede proporcionar también buenos resultados, como frente a la serie MG17, la crítica selección de su parámetro α debido a su especial algoritmo de entrenamiento provoca que se puedan obtener, en otras ocasiones, resultados de predicción claramente inferiores al algoritmo tradicional, como frente a la serie Hénon. El algoritmo SOMDL, como ya se comentó anteriormente, ha sido ya evaluado en la bibliografía satisfactoriamente para este propósito. En el caso del SOMTAD y el MSOM, su potencial proviene de la capacidad para mantener el contexto temporal

a nivel de toda la red. El algoritmo SARD presenta también buenos resultados, pese a la aparente sencillez de su planteamiento.

Otra conclusión a la vista de los resultados es que es conveniente contar con métodos que permitan seleccionar los parámetros correctos para el objetivo que se pretenda afrontar, de igual modo que en este caso se ha utilizado la validación cruzada multi-paso para el objetivo de predicción varios pasos adelante. La presencia de uno o más parámetros libres adicionales en los algoritmos revisados dificulta la selección directa de los mismos por medio del conocimiento previo o las reglas heurísticas.

Esfuerzos futuros en la definición de modificaciones espacio-temporales del SOM para el modelado dinámico podrían ir dirigidas a combinar la utilización de reglas que pongan énfasis en el aprendizaje de secuencias complicadas, como en el algoritmo SOMDL, y de memorias a corto plazo potentes, como las generadas a través de la recursividad del MSOM o la difusión de la actividad en el SOMTAD. Es decir, métodos que utilicen la información de error para regular la acción de la memoria a corto plazo de la red. Estos métodos deberían asimismo permitir una selección apropiada de los parámetros libres de forma automática o limitar la existencia de los mismos.

5.1.2. Datos reales

De acuerdo a los resultados obtenidos en el apartado anterior y con el objetivo de extender la verificación de los algoritmos a señales reales, se compara el modelado de los algoritmos SOM, SOM con aprendizaje dinámico, SOM recurrente, SARDNET, *Merge* SOM y SOMTAD, con ajuste por medio de los vectores de entrada asociados a las regiones de Voronoi de las neuronas. Se utiliza este ajuste porque ha mostrado mejores resultados ante las series caóticas tanto en predicción como en preservación de la dinámica. Para ello, como se indicó en el apartado 4.4.3, se utiliza una serie de prueba obtenida a partir de datos adquiridos de un sistema industrial real. La comparación, en este caso, se ciñe solamente al error obtenido en la predicción iterada de la serie.

Los procedimientos y peculiaridades del entrenamiento son equivalentes a los ya presentados en el apartado anterior. De hecho, en la Tabla 5.7 se muestran los parámetros seleccionados por medio de la validación cruzada de los modelos, que también utiliza un horizonte de 8 pasos adelante en esta ocasión.

La evaluación por medio del error de predicción varios pasos adelante confirma las conclusiones extraídas de la evaluación de las series caóticas, ya que los algoritmos que, en general, presentaban menor error frente a las series caóticas, también realizan una mejor predicción en este caso (SOMDL, SOMTAD, MSOM

Parámetros del algoritmo – Ajuste mediante region de Voronoi									
	Dims	**d_e**	**Épocas**	**σ_{h1}**	**σ_{h1n}**	**μ**	**β**	**α**	***d***
SOM	35x35	3	200	10	0,5	-------	-------	-------	-------
SOMDL	35x35	4	200	10	0,5	2	-------	-------	-------
RSOM	35x35	4	200	10	0,5	-------	-------	0,78	-------
SARDNET	35x35	4	200	10	0,5	-------	-------	-------	0,6
MSOM	35x35	4	200	10	0,5	-------	0,6	0,1	-------
SOMTAD	35x35	3	200	10	0,5	0,8	0,4	-------	-------

Tabla 5.7. Selección de parámetros para la serie objeto de prueba.

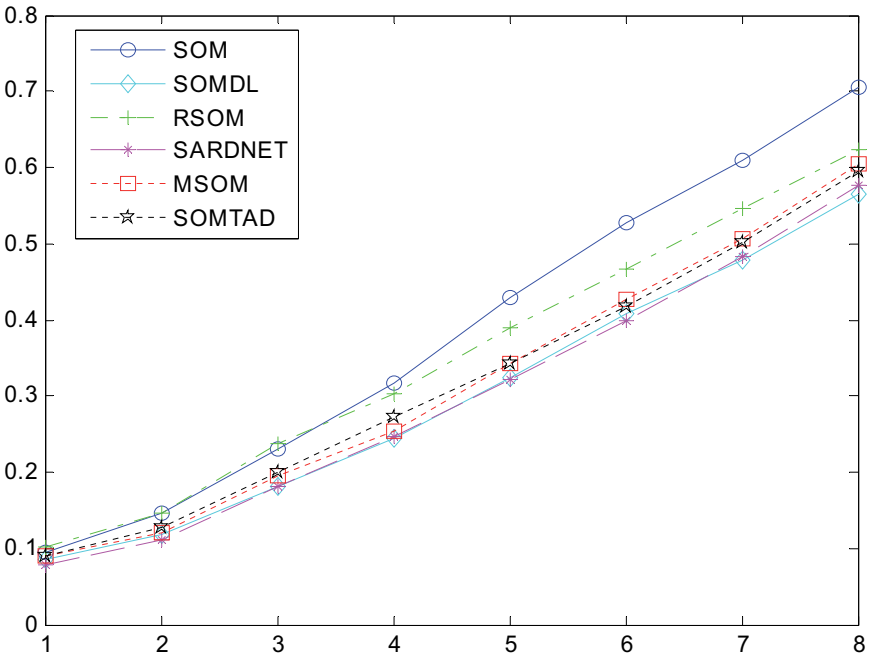


Figura 5.7. Curvas de error RNMSE para la serie de datos real.

y SARDNET). Una gráfica del error de predicción frente al número de pasos adelante puede verse en la Fig. 5.7, mientras que la Tabla 5.8 muestra numéricamente el NRMSE 8 pasos adelante.

	SOM	SOMDL	RSOM	SARD	MSOM	SOMTAD
Error	0,6675	0,5657	0,6243	0,5768	0,6047	0,5948

Tabla 5.8. Errores de predicción 8 pasos adelante.

5.2. Mapas de visualización en sistemas SISO

5.2.1. Experimento SISO1: Mapas de visualización para control de nivel simulado

El conjunto de datos de entrada, previamente normalizado, se utiliza para un entrenamiento *batch* de 100 épocas de un SOM 20×24 con vecindad gaussiana decreciente. Las dimensiones del mapa han sido elegidas para minimizar la distorsión de los datos de entrada. El error de cuantización medio del mapa es de $0,0013$.

Tras el entrenamiento se procede a visualizar los mapas de dinámica, los cuales, tal y como se indicó en el capítulo 4, se generan a partir de los vectores *codebook* debidamente desnormalizados, asociando a cada posición $\mathbf{g}_i$ el valor correspondiente a la característica k_i, que es función de los parámetros $\mathbf{p}_{m_i}$. De igual manera, se presentan los planos de componentes de las variables selectoras $\mathbf{x}_{m_i}$. Los mapas de dinámica propuestos y los planos de componentes se interpretan poniendo énfasis en las conclusiones que puedan extraerse de los mismos, su relación con los otros mapas y el grado de coherencia con respecto al conocimiento previo.

Se comienza visualizando los planos de componentes de las variables selectoras de la dinámica que forman parte del espacio de entrada: nivel (Fig. 5.8) y válvula (Fig. 5.9). Los planos de componentes son necesarios como complemento de los mapas de dinámica que proponemos, porque permiten enlazar los comportamientos dinámicos con los puntos de funcionamiento.

En cuanto a los mapas de dinámica, el mapa de tiempo de establecimiento (Fig. 5.10) muestra una clara correlación con la variable nivel, que se explica intuitivamente por el hecho de que el tanque desagua más a mayor altura del líquido. No se detecta una clara influencia de la válvula en esta característica.

Sin embargo, el mapa de tiempo de pico (Fig. 5.11) y el mapa de tiempo de subida (Fig. 5.12) revelan relación tanto con el nivel como con la válvula.

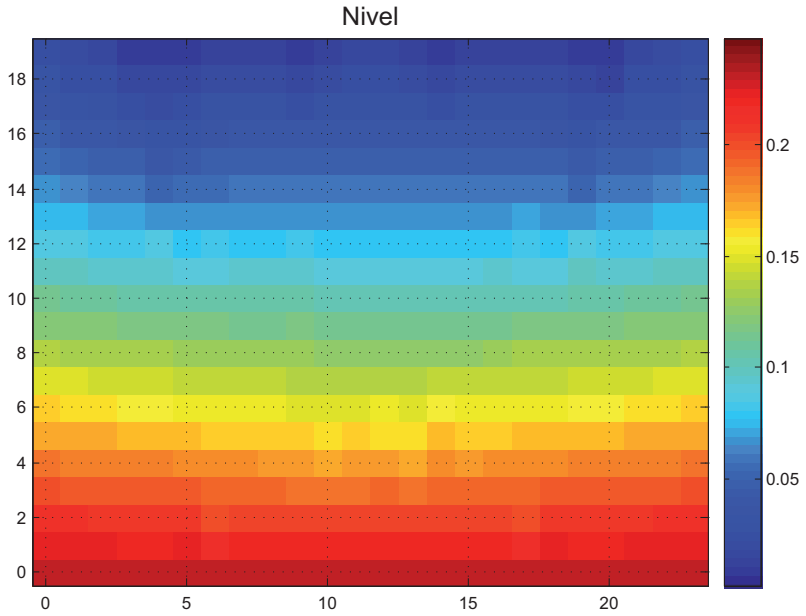


Figura 5.8. Plano del componente nivel.

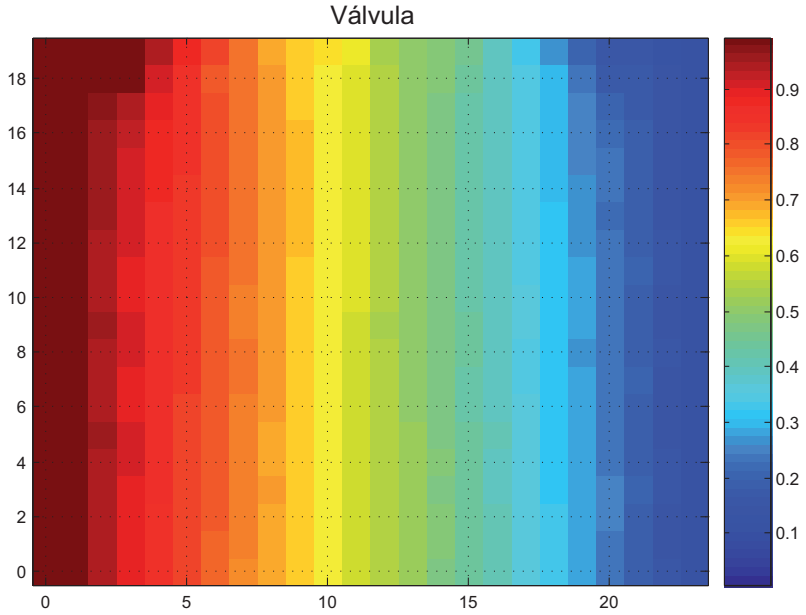


Figura 5.9. Plano del componente válvula.

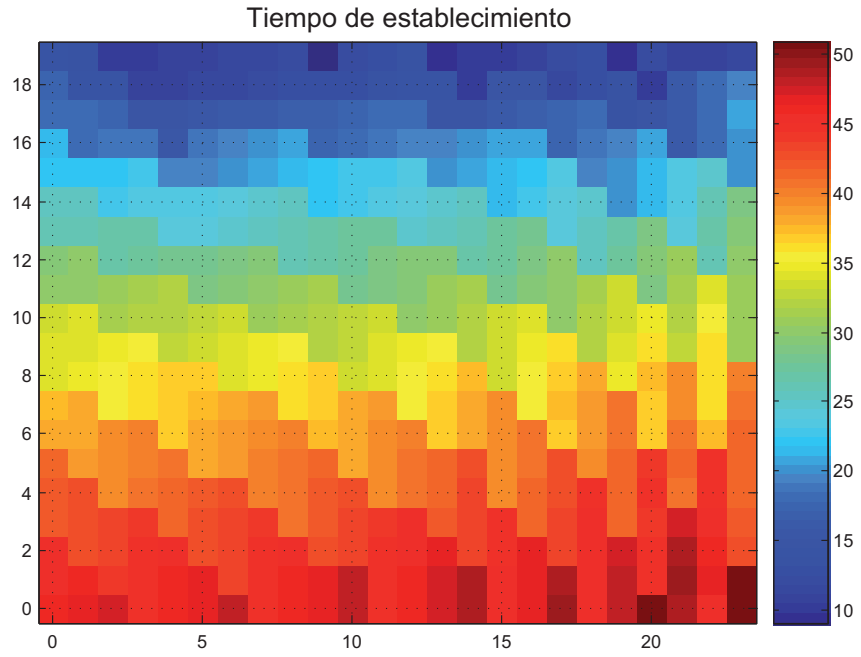


Figura 5.10. Mapa de tiempo de establecimiento.

Concretamente, sus valores aumentan de forma moderada para valores bajos de válvula y en mayor medida en zonas en las que, además, el nivel también es bajo.

Los tiempos son más lentos debido a que la disminución del nivel y la válvula también conlleva un aumento del coeficiente de amortiguamiento (Fig. 5.13) y, por tanto, una disminución de la sobreoscilación (Fig. 5.14), La frecuencia natural (Fig. 5.15), por otra parte, se encuentra muy correlacionada con la variable válvula.Finalmente, como era de esperar, la ganancia (Fig. 5.16) es unitaria en todo el rango de operación.

Por tanto, en este experimento la visualización de la dinámica permite determinar simples reglas acerca de las variaciones de las características dinámicas a lo largo del rango de trabajo, algunas de las cuales no son totalmente intuitivas a priori. También permite relacionar las características dinámicas con las variables selectoras y entre sí. Por ello, los resultados sugieren que esta información visual puede ser útil para entender globalmente el comportamiento dinámico de sistemas más complejos que el que nos ocupa.

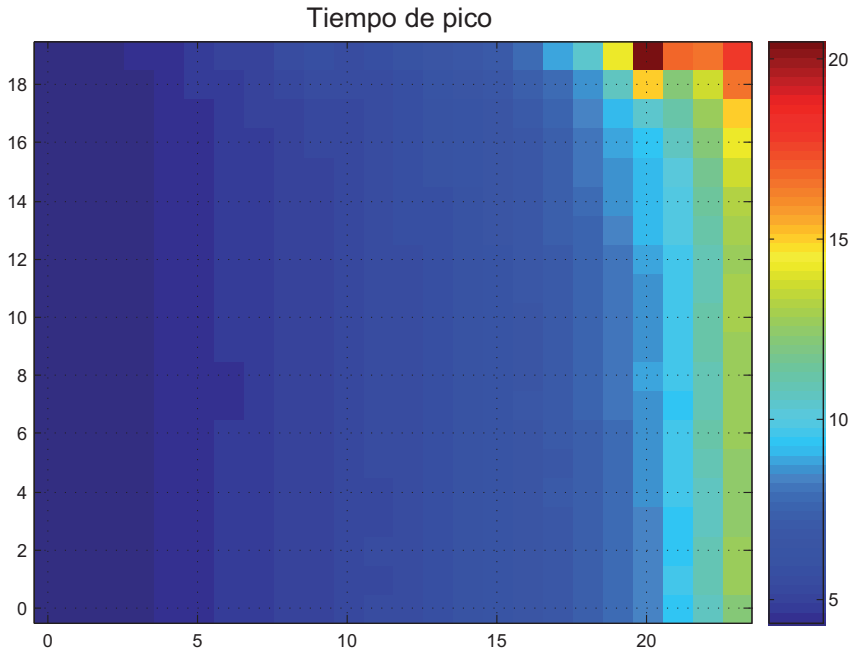


Figura 5.11. Mapa de tiempo de pico.

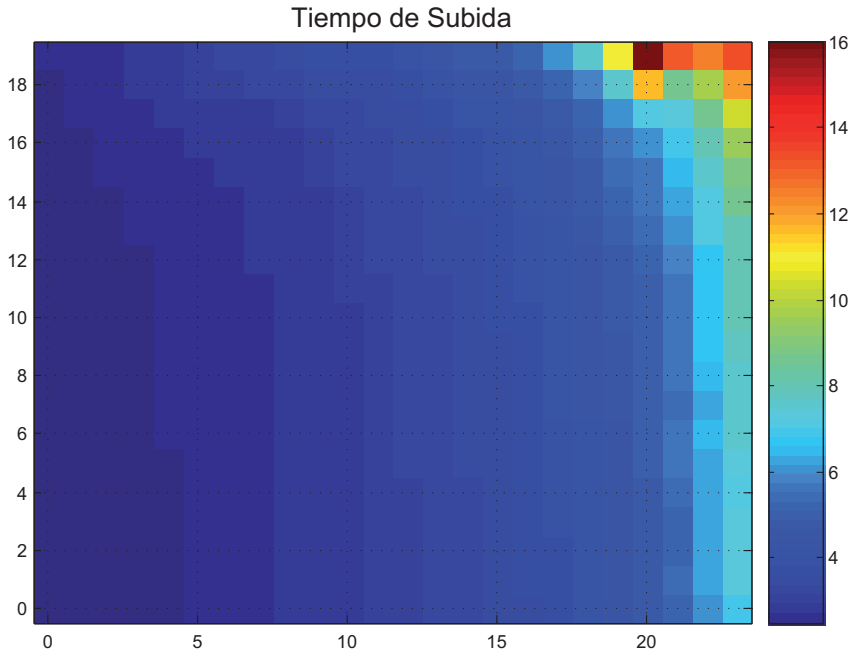


Figura 5.12. Mapa de tiempo de subida.

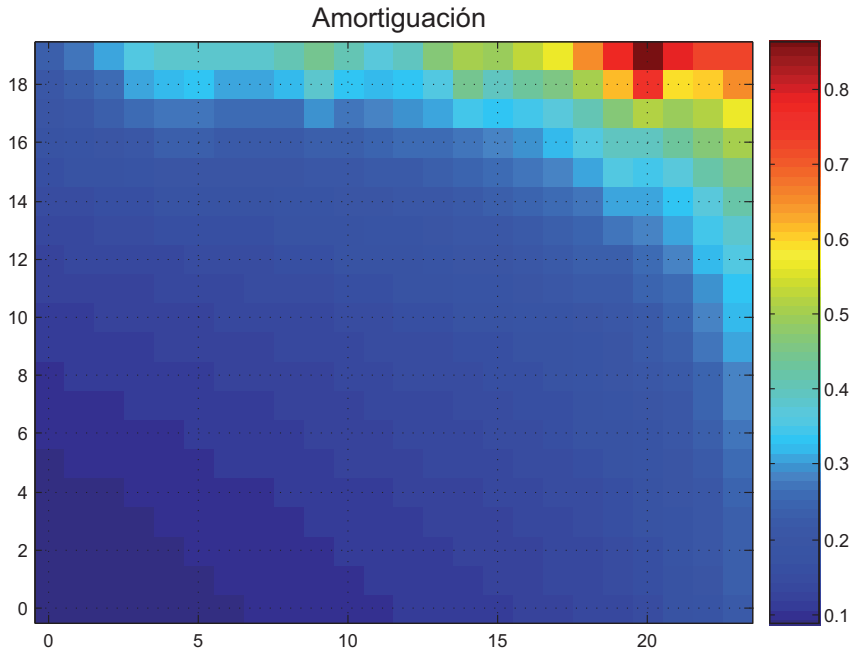


Figura 5.13. Mapa de coeficiente de amortiguamiento.

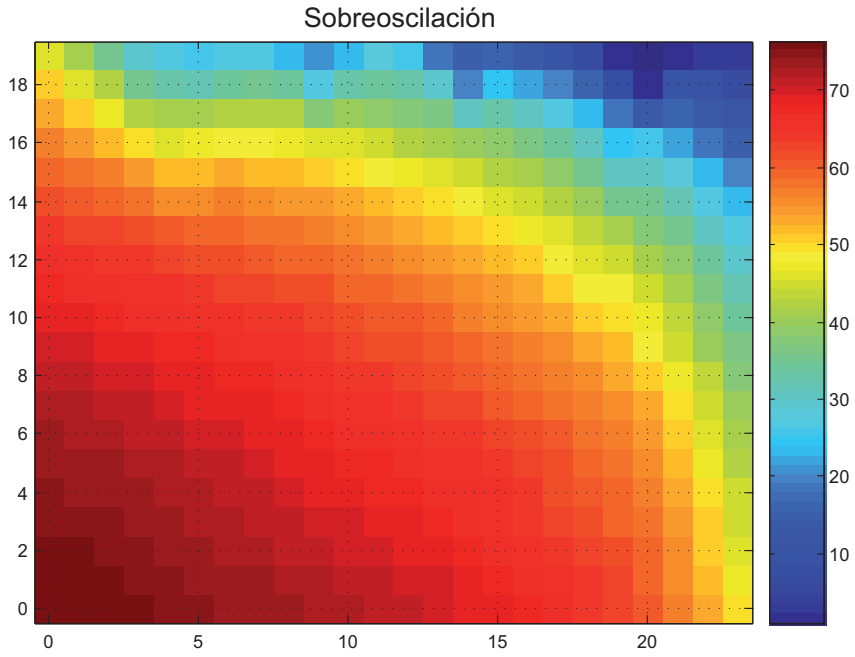


Figura 5.14. Mapa de sobreoscilación.

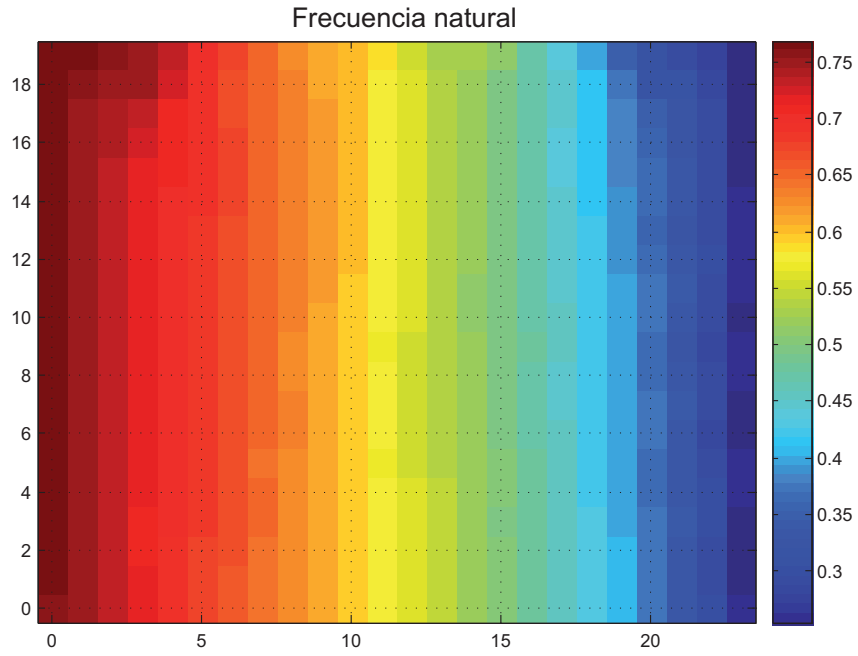


Figura 5.15. Mapa de frecuencia natural.

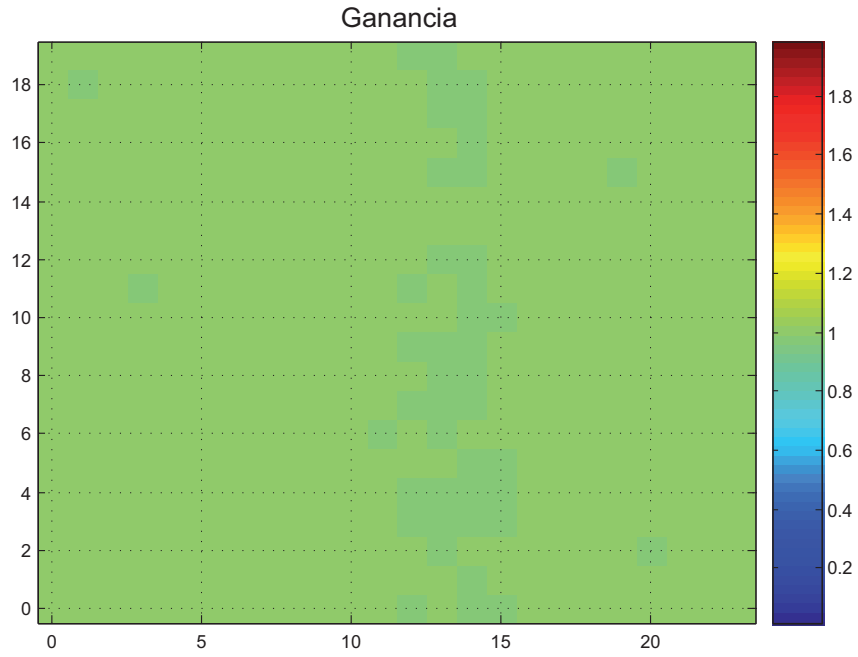


Figura 5.16. Mapa de ganancia.

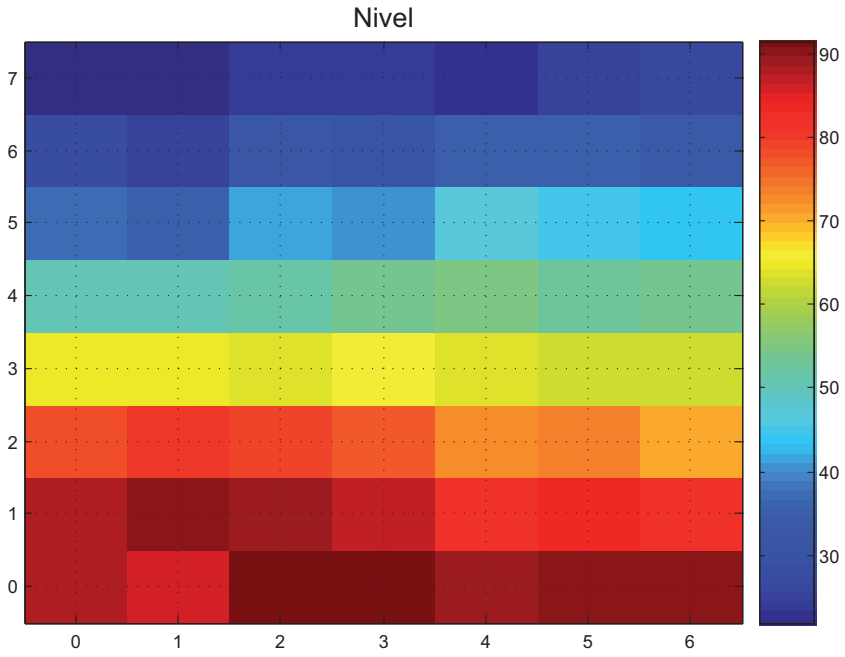


Figura 5.17. Plano del componente nivel.

5.2.2. Experimento SISO2: Mapas de visualización para control de nivel real

El conjunto de datos de entrada, previamente normalizado, se utiliza para un entrenamiento *batch* de 100 épocas de un SOM 7×8 con vecindad gaussiana decreciente. Las dimensiones del mapa han sido elegidas para minimizar la distorsión de los datos de entrada. El error de cuantización medio del mapa es de $0,0210$.

Tras el entrenamiento se procede a visualizar los mapas de dinámica, generados a partir de los vectores *codebook* debidamente desnormalizados. Los mapas de dinámica propuestos, junto con los planos de componentes, se interpretan poniendo énfasis en las conclusiones que puedan extraerse de los mismos, las interrelaciones y el grado de coherencia con respecto al conocimiento previo.

En primer lugar, se visualizan los planos de componentes de las variables selectoras de la dinámica, es decir, el nivel en el tanque (Fig. 5.17) y la apertura de la válvula (Fig. 5.18). Estos planos permiten enlazar los comportamientos dinámicos y los puntos de funcionamiento.

Al igual que en el experimento anterior, la ganancia (Fig. 5.19) permanece aproximadamente constante en todos los puntos de funcionamiento, lo que confirma las hipótesis fundadas en el conocimiento previo del sistema.

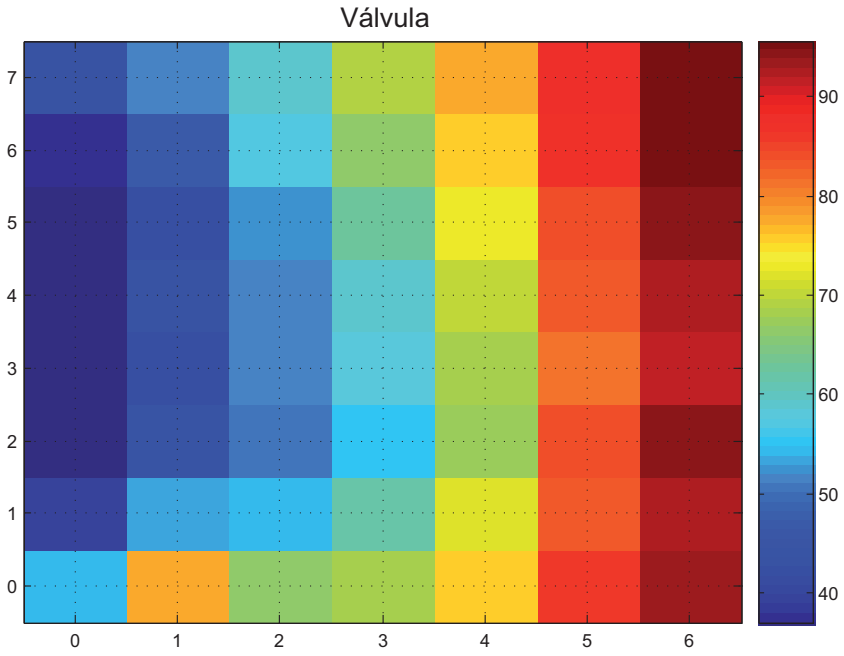


Figura 5.18. Plano del componente válvula.

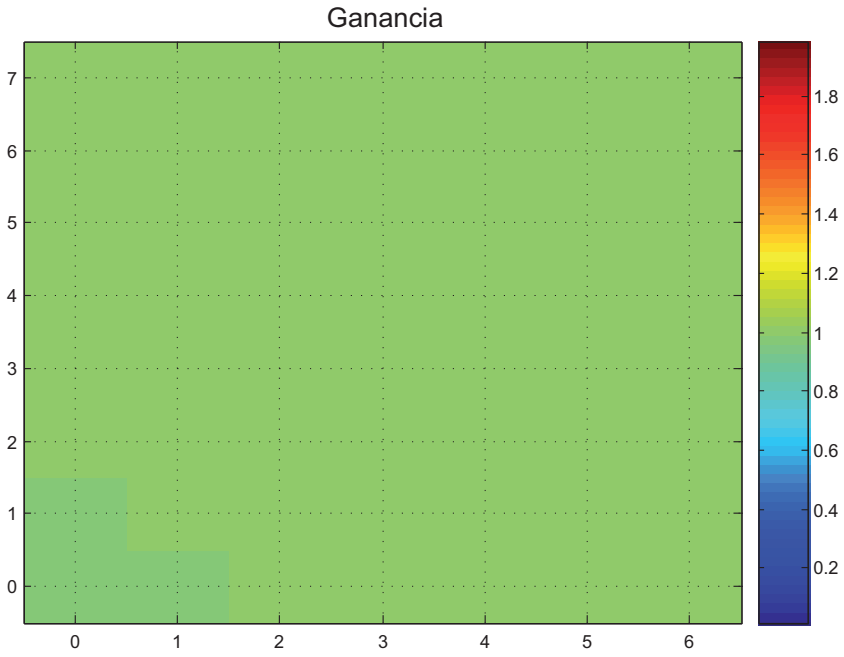


Figura 5.19. Mapa de ganancia.

Tras un análisis visual de los mapas de dinámica se puede concluir que, en este experimento, la apertura de la válvula es la que determina en mayor medida el comportamiento. Para valores bajos de la misma, es decir, cuando la válvula se encuentra parcialmente cerrada, los tiempos de establecimiento (Fig. 5.20), pico (Fig. 5.21) y subida (Fig. 5.12) son mayores. Es decir, el sistema, en líneas generales, es más lento cuanto menor es la apertura de la válvula. Esta comportamiento tiene una interpretación intuitiva, ya que a menor caudal de líquido entrante, más tiempo se tardará en alcanzar la consigna.

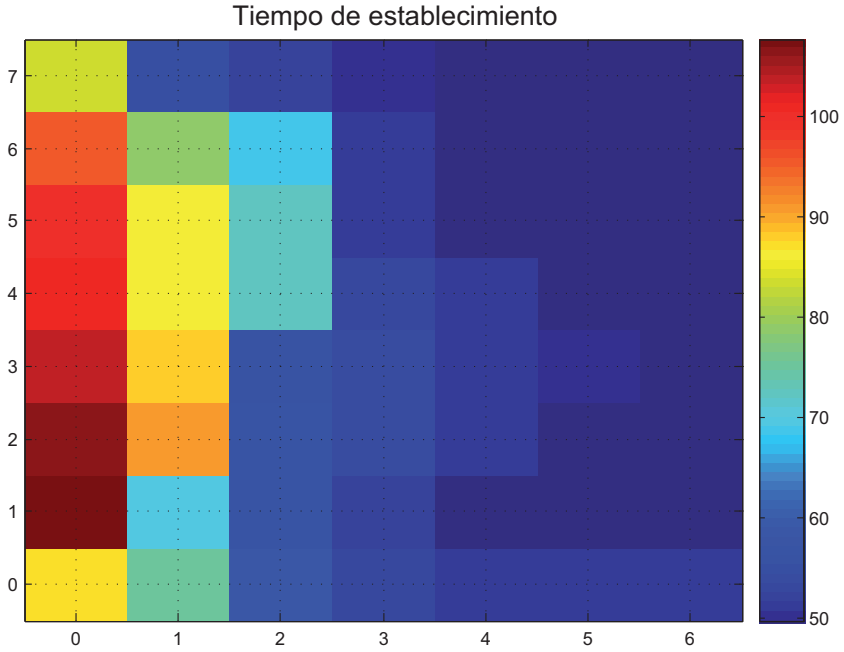


Figura 5.20. Mapa de tiempo de establecimiento.

Además de ralentizar el sistema, una apertura de la válvula pequeña hace al sistema más oscilatorio, provocando una disminución del coeficiente de amortiguamiento (Fig. 5.23) y un aumento de la sobreoscilación (Fig. 5.24). No obstante, la frecuencia natural (Fig. 5.25) del sistema disminuye en este caso.

Aunque la influencia del nivel del tanque es menor, también tiene un efecto sobre las características dinámicas. A menores valores del mismo, el tiempo de pico y de subida disminuyen, debido a un ligero decremento del coeficiente de amortiguamiento, que provoca una mayor sobreoscilación, y un pequeño aumento de la frecuencia natural.

En resumen, al igual que en el experimento anterior, la visualización de la dinámica permite obtener simples reglas acerca del valor y la variación de carac-

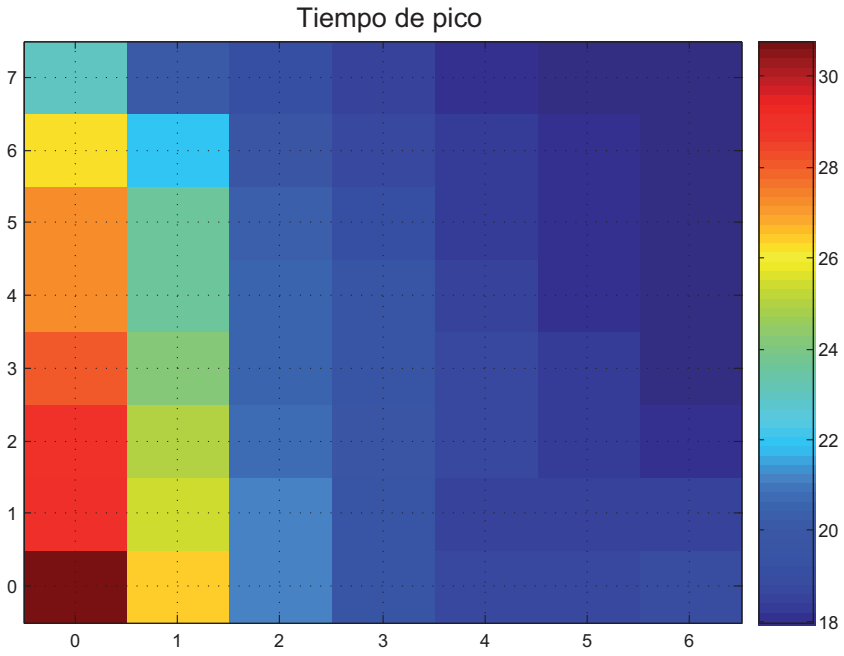


Figura 5.21. Mapa de tiempo de pico.

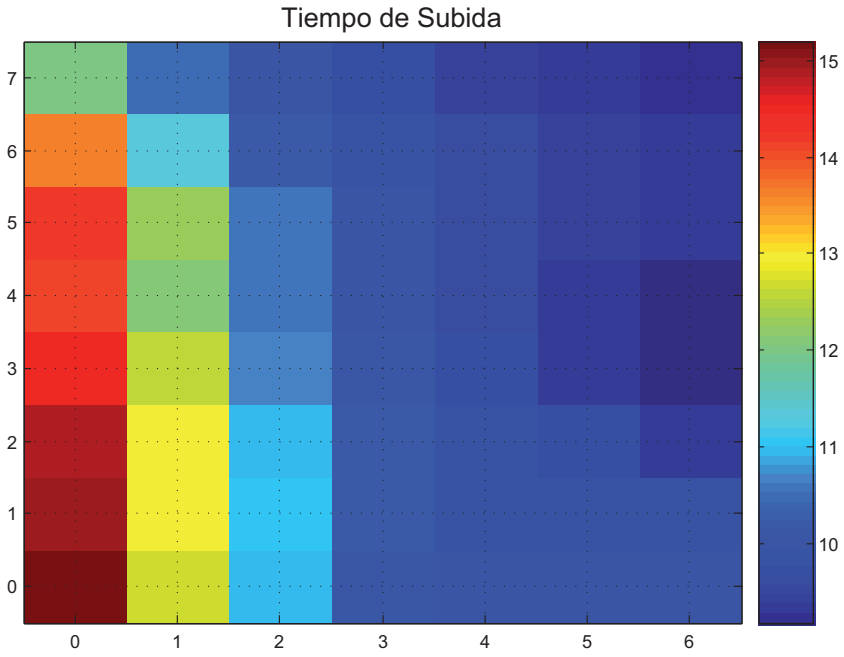


Figura 5.22. Mapa de tiempo de subida.

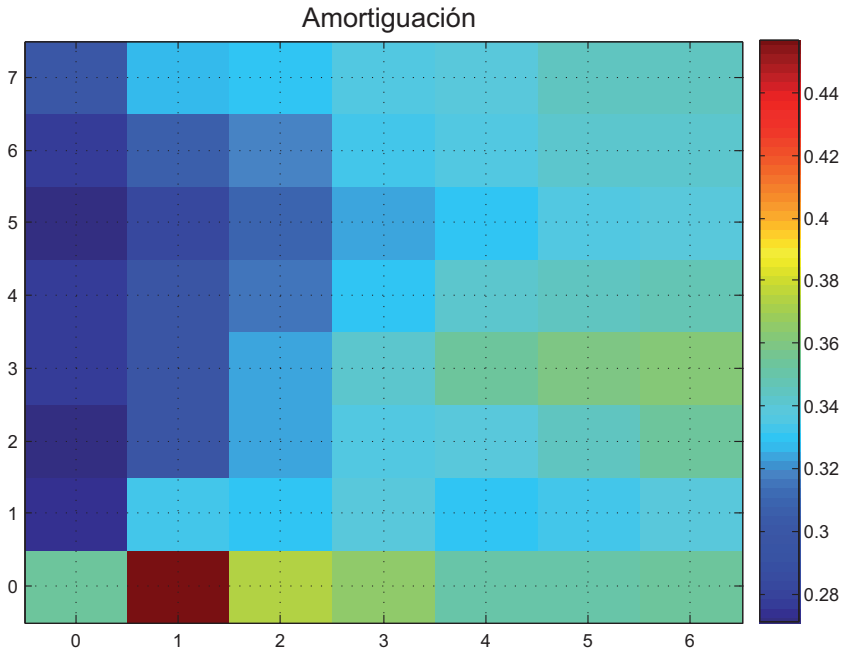


Figura 5.23. Mapa de coeficiente de amortiguamiento.

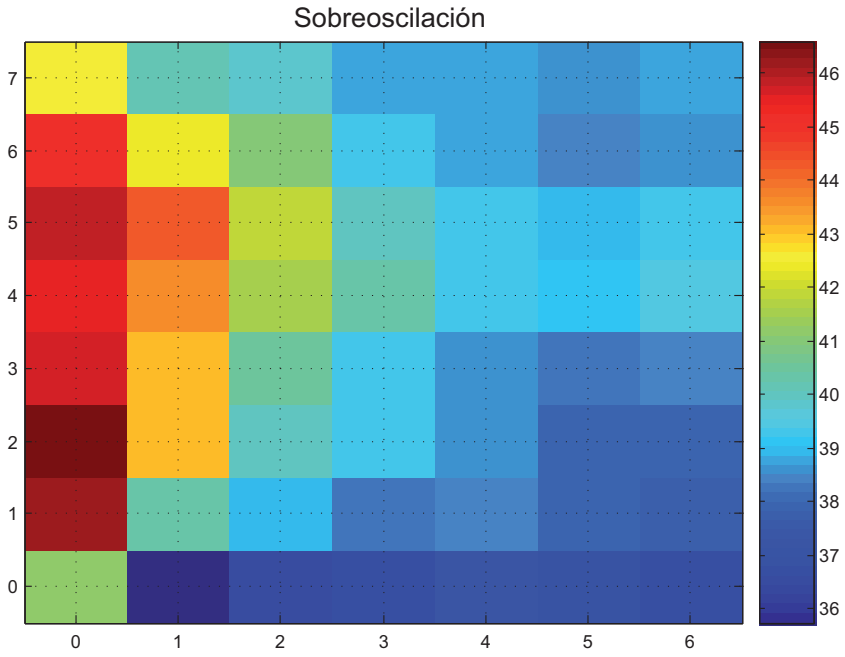


Figura 5.24. Mapa de sobreoscilación.

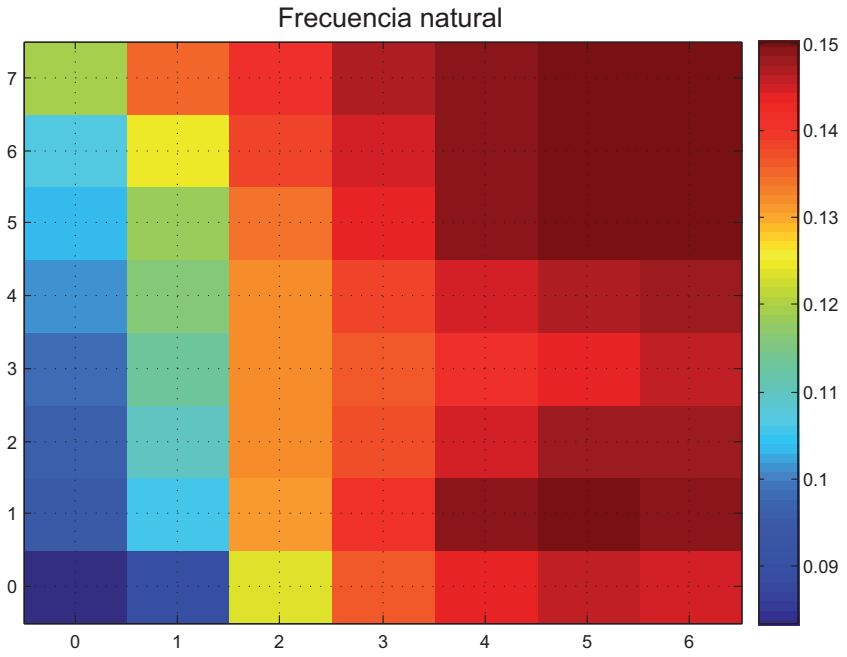


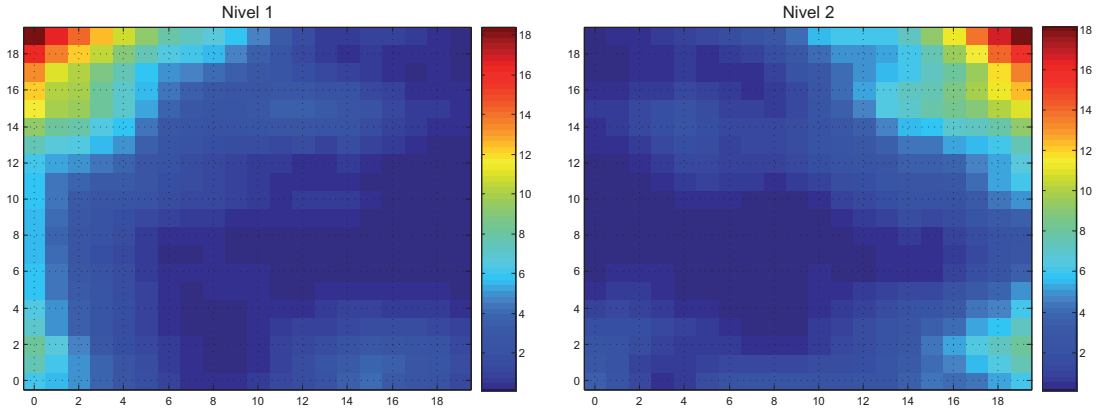
Figura 5.25. Mapa de frecuencia natural.

terísticas dinámicas en todo el espacio de funcionamiento. Por ello, estos mapas pueden ser útiles para confirmar, ampliar o descubrir informaciones acerca del comportamiento dinámico del sistema. Las correlaciones con variables del sistema resultan, una vez más, especialmente útiles.

5.3. Mapas de visualización en sistemas MIMO

5.3.1. Experimento MIMO1: Mapas de visualización para el modelo de 4 tanques

El conjunto de datos de entrada, previamente normalizado, se utiliza para un entrenamiento *batch* de 200 épocas de un SOM 20×20 con vecindad gaussiana decreciente. En este caso, puesto que el conjunto de datos de entrada tiene mayor número de elementos, se ha seleccionado una dimensión lo suficientemente grande como para no cuantizar demasiado los datos, pero no lo demasiado como para impedir su correcta visualización, especialmente de las flechas asociadas a las características vectoriales/direccionales. El error de cuantización medio en este caso es de $0,0298$.



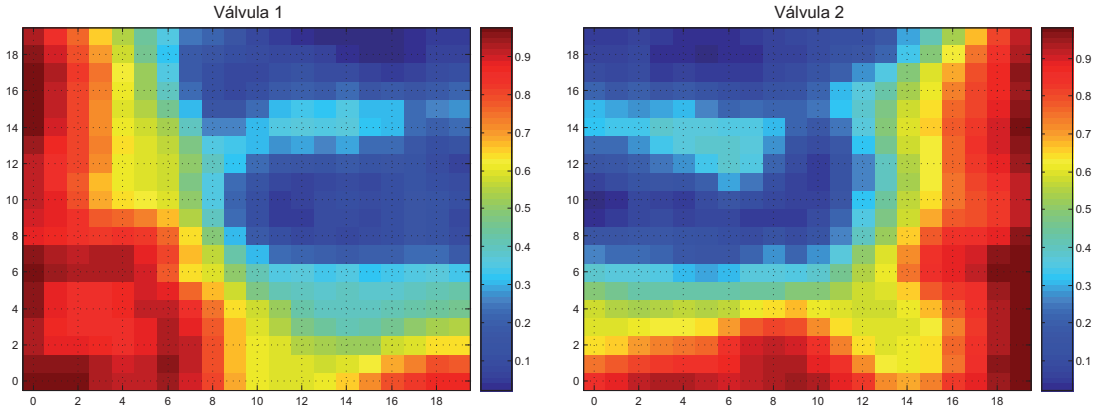
(a) Nivel 1. **(b)** Nivel 2.

Figura 5.26. Planos de componentes de nivel.

Tras el entrenamiento se procede a visualizar los mapas de dinámica, generados a partir de los vectores *codebook* debidamente desnormalizados. Los mapas de dinámica y los planos de componentes se interpretan poniendo énfasis en las conclusiones que puedan extraerse de los mismos, su relación con los otros mapas y el grado de coherencia con respecto al conocimiento previo. En algunos casos, los mapas se generan con un efecto contorno que facilita la visualización de las fronteras entre zonas, sin variar en ningún momento sus propiedades.

En primer lugar se presentan los planos de componentes correspondientes a las variables selectoras de la dinámica incluidos en el espacio de entrada: niveles (ver Fig. 5.26) y válvulas (ver Fig. 5.27). Los planos de componentes de estas últimas son especialmente importantes para comprender el comportamiento del proceso, cuya dinámica depende en gran medida de sus valores.

De hecho, de acuerdo al modelo original de cuatro tanques, la suma de las aperturas de las válvulas determina si el sistema es de fase mínima o no mínima. El sistema tiene dos ceros finitos para $\gamma_1, \gamma_2 \epsilon (0, 1)$, uno siempre en el semiplano izquierdo mientras que el otro puede ser negativo o positivo en función del valor de la suma de las aperturas de las válvulas. Concretamente, si $0 < \gamma_1 + \gamma_2 < 1$, el cero es positivo y el sistema es de fase no mínima, lo que provoca limitaciones en el rendimiento del sistema. Esta situación se puede comprender de una forma intuitiva, pues esa apertura de las válvulas indica que la mayor parte del líquido se vierte en los depósitos superiores, lo que supone una dificultad añadida. En los casos en que $1 < \gamma_1 + \gamma_2 < 2$, es decir, cuando el caudal a los tanques inferiores es superior, el cero se sitúa en el semiplano izquierdo y, por tanto, el sistema es de fase mínima.



(a) Válvula 1. **(b)** Válvula 2.

Figura 5.27. Planos de componentes de válvulas.

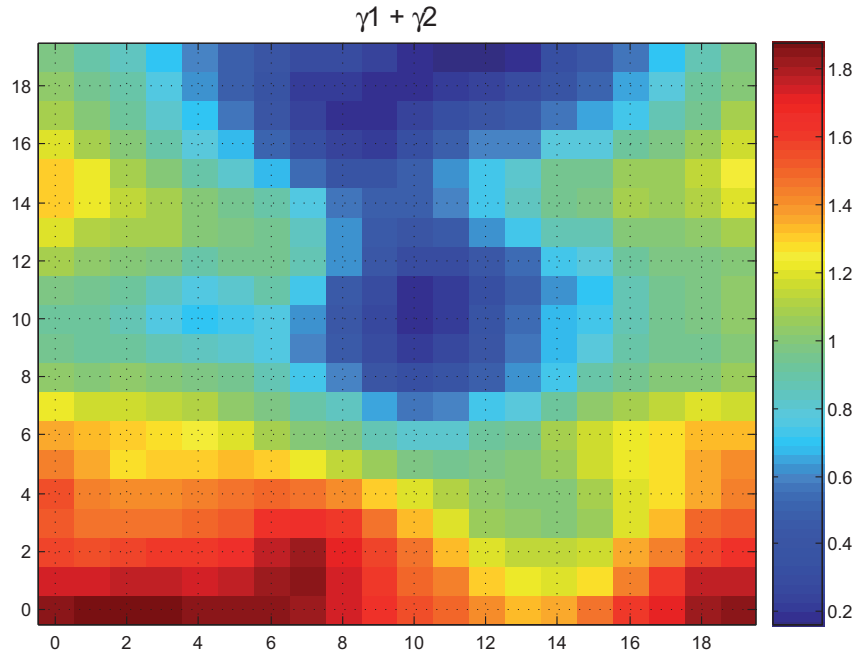


Figura 5.28. Mapa de $\gamma_1 + \gamma_2$

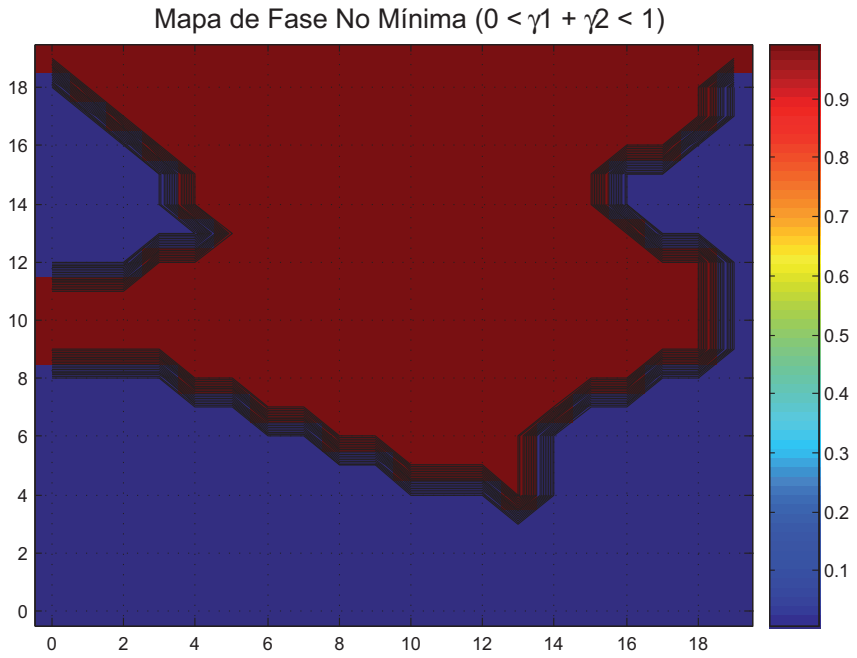
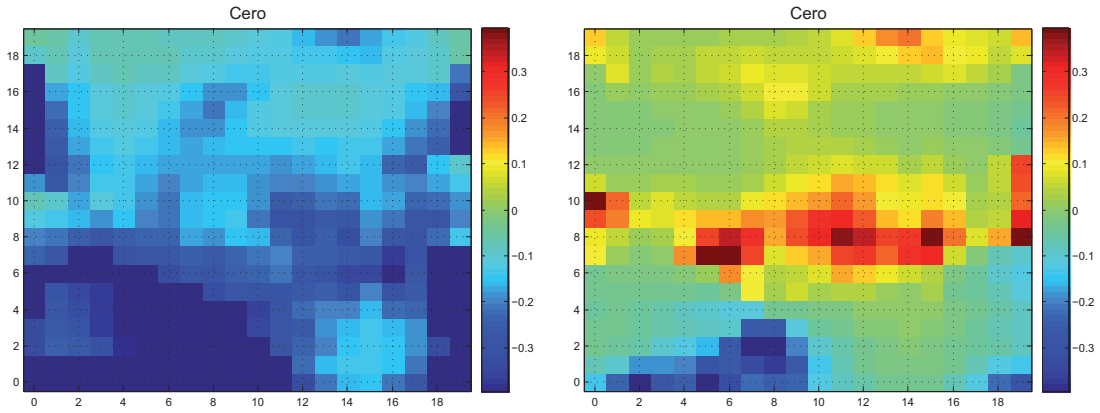


Figura 5.29. Mapa de fase no mínima de acuerdo a la apertura de las válvulas.



(a) Cero siempre negativo.

(b) Cero "móvil".

Figura 5.30. Mapa de ceros multivariable.

Si se presenta un plano de la suma de componentes $\gamma_1+\gamma_2$ (Fig. 5.28), se puede visualizar cómo sus valores se agrupan por zonas. Introduciendo un umbral en $\gamma_1+\gamma_2=1$, se separan los puntos de funcionamiento para los cuales el comportamiento es de fase no mínima y aquellos en los que es de fase mínima en regiones claramente diferenciadas (Fig. 5.29). Este mapa, junto con el conocimiento previo acerca de la relación entre las válvulas y los ceros que se acaba de exponer, puede utilizarse para evaluar la coherencia de los mapas de ceros multivariable propuestos para este experimento. Los ceros, calculados como las raíces del determinante de la matriz de transferencia en cada vector prototipo o punto de funcionamiento del mapa auto-organizado, dan lugar a dos mapas (Fig. 5.30). El primero de ellos muestra valores negativos de la característica en todos los puntos, mientras que el segundo presenta valores positivos en ciertas regiones. Estas regiones deben corresponderse con las mostradas en la Fig. 5.29, lo que efectivamente ocurre, como queda patente al visualizar $z>0$ (Fig. 5.31).

Una propiedad importante de los ceros multivariable es su dirección, la cual determina el efecto del mismo sobre las diferentes salidas. Por ello, resulta especialmente relevante conocer la dirección de salida del mismo en las regiones de fase no mínima. En este caso, al contar el sistema solamente con dos salidas, es posible visualizar, en forma de flechas la dirección del cero sobre su propio valor (o, en este caso, sobre las regiones que determina). La visualización del mismo (Fig. 5.32) permite distinguir cómo, en la zona en que el valor de γ_1 es bajo y el de γ_2 es alto, es decir, en los puntos de funcionamiento en que la válvula 1 envía gran parte del caudal al tanque superior mientras que el tanque inferior 2 recibe la mayor parte del caudal de la segunda bomba, la dirección está asociada principalmente con la primera salida. Se puede ver el efecto inverso en la zona de valores bajos de $\gamma 2$ y una influencia más repartida en las zonas intermedias. Estas situaciones pueden interpretarse como un efecto más pronunciado del cero de fase no mínima para aquellas direcciones que implican que gran parte del fluido se envía al tanque superior, lo que tiene sentido intuitivo.

Por otra parte, la visualización del mapa de ganancias relativas debe, por otra parte, permitir definir los mejores emparejamientos para el control descentralizado: u_1-y_1/u_2-y_2 o u_1-y_2/u_2-y_1, a los que en adelante nos referiremos como lazo directo o lazo cruzado. Otro objetivo de la visualización de este mapa es distinguir si existen zonas en las que el control sea difícil ($\lambda<0$ ó $\lambda>>0$). La representación del mapa de ganancias se facilita en el caso 2×2, ya que basta con observar el valor λ_{11} en cada punto de funcionamiento. Si observamos detenidamente el mapa RGA que se obtiene para este sistema (Fig. 5.33), podemos reconocer su coherencia con el mapa de fase no mínima, aunque podemos llegar a más conclusiones acerca del acoplamiento de las variables y la facilidad de control. En este mapa, se puede distinguir que los vectores de la zona de fase mínima tienen asociado unos valores

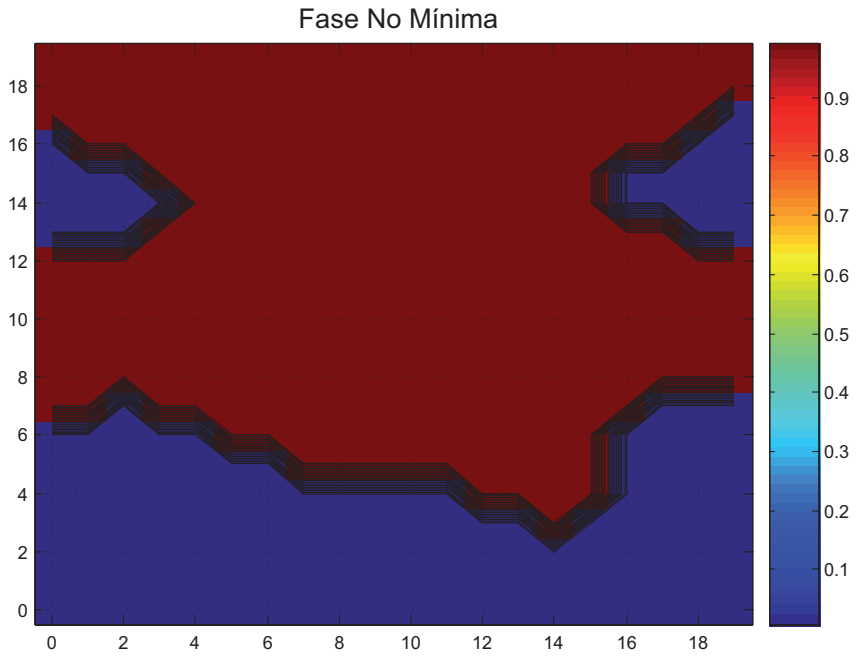


Figura 5.31. Mapa de fase no mínima de acuerdo al signo del cero.

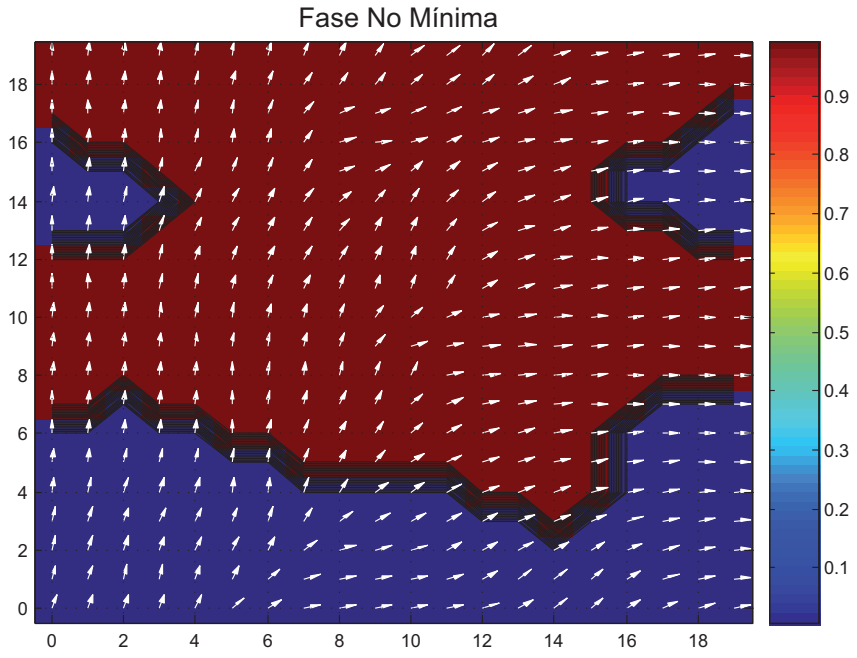


Figura 5.32. Mapa de fase no mínima y direcciones de salida del cero multivariable.

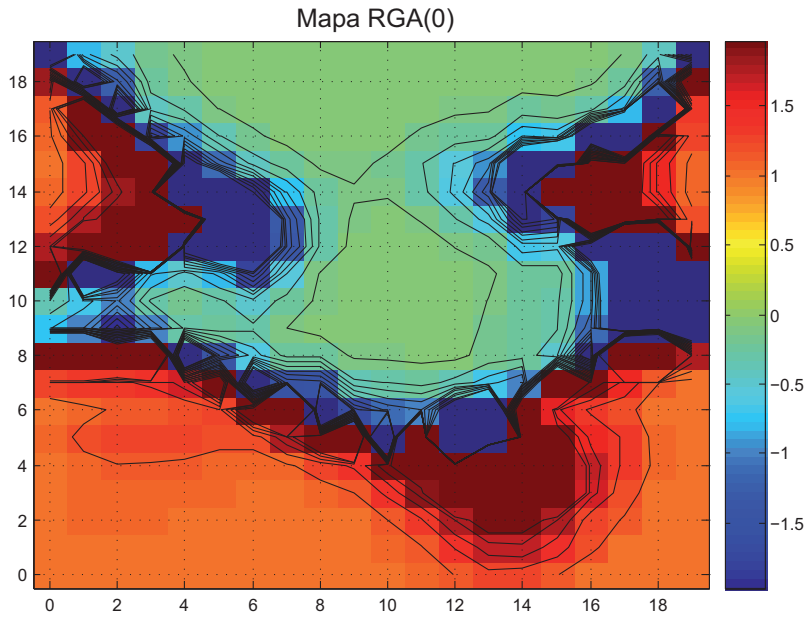


Figura 5.33. Mapa de array de ganancias relativas

$\lambda_1 1 \approx 1$. En estos casos el efecto de acoplamiento entre lazos es mínimo y el sistema puede controlarse de forma descentralizada seleccionando los lazos directos. Dentro de la región establecida como de fase no mínima por válvulas y ceros, aparecen diversos valores de λ. Una amplia zona de esta región presenta valores cercanos 0. En esos puntos el sistema aún puede ser controlado satisfactoriamente por medio de una estrategia descentralizada, pues el acoplamiento es mínimo. No obstante, para que esto sea posible es necesario cruzar los lazos. Las zonas en las que λ es negativo o presenta valores en torno a 2 presentan un comportamiento especialmente difícil de controlar por medio de una estrategia descentralizada, por lo que otro tipo de control sería preferible.

En su artículo, Johansson desarrolla una simple ecuación para calcular el valor de la ganancia relativa por medio de las aperturas de las válvulas

$$\lambda = \frac{\gamma_1 \gamma_2}{\gamma_1 + \gamma_2 - 1}, \tag{5.1}$$

lo que permite generar otro mapa (Fig. 5.34) que concuerda con el previamente calculado.

Se puede ampliar el conocimiento acerca del sistema por medio del análisis de los resultados de la descomposición en valores singulares de la matriz de ganancia en régimen permanente. La visualización conjunta de las direcciones singulares de entrada y de salida proporciona otro criterio para seleccionar el mejor empa-

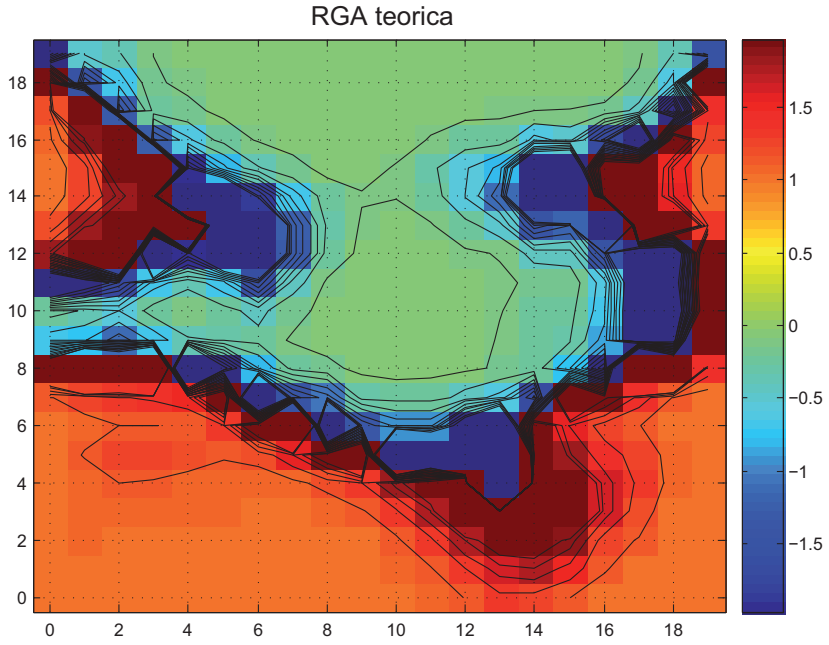


Figura 5.34. Mapa del array de ganancias relativas de acuerdo a la apertura de las válvulas

rejamiento entre variables en cada punto de funcionamiento. Asimismo, permite analizar la distribución de ganancia del sistema. La visualización de los mapas del mayor y menor valor singular (Fig. 5.35 y Fig. 5.36) permite comprobar que la mayor ganancia del sistema parece depender en mayor medida de las alturas de los niveles, pues es mínima en la región correspondiente a los niveles más bajos de los depósitos inferiores. Al contrario, la menor ganancia del sistema se ve claramente influenciada por los valores de las válvulas. De hecho, es evidente que su distribución en regiones se corresponde con las obtenidas para el mapa de ganancias relativas (Fig. 5.33) de tal modo que la ganancia es máxima en las zonas que permiten un control descentralizado satisfactorio.

Pero esta no es la única conclusión que se puede extraer a partir de estos mapas. Si se observan las direcciones de entrada (flechas negras) y de salida (flechas blancas) en la parte inferior de ambos mapas, que coincide con una región de fase mínima y $\lambda = 1$, se puede ver que ambas direcciones prácticamente coinciden en la mayor parte de los casos[3]. Así, cuando el vector entrada apunta en mayor

[3]Se recuerda que las direcciones asociadas a este mapa se calculan considerando únicamente la magnitud de cada componente, para hacer más fácil su interpretación. Por esa razón, podría parecer que en algunos casos los dos vectores de entrada (o los dos de salida) en cada mapa no son ortogonales, cuando en realidad lo son por definición.

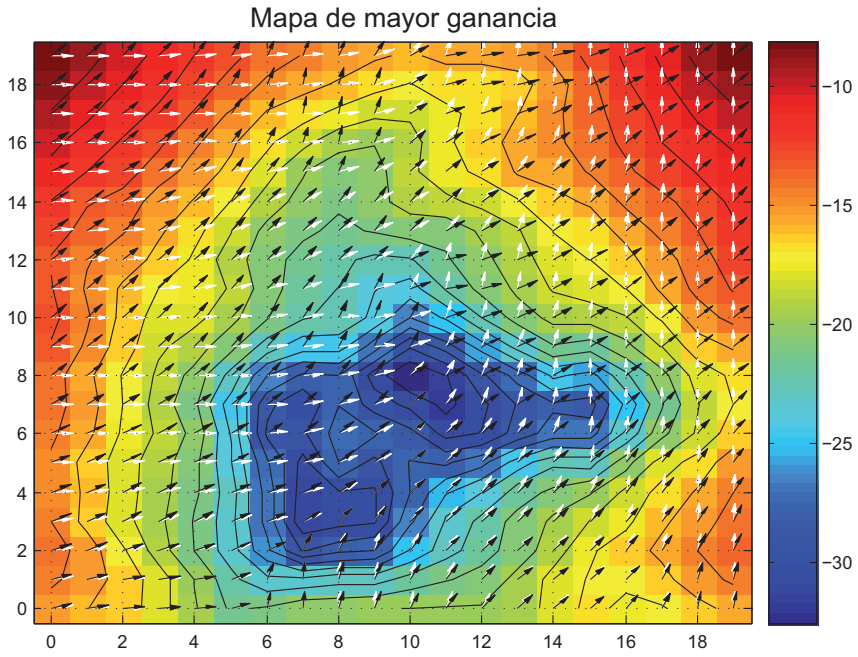


Figura 5.35. Mapa de mayor ganancia, $\sigma_1(0)$, y de sus direcciones de entrada y salida asociadas.

medida a la entrada u_1 (mayor valor de la primera componente), el vector de salida apunta a la salida y_1. Alternativamente, cuando apunta a u_2, el vector de salida lo hace a y_2. Esto sucede en la zona en la que los valores de válvula son altos y, por tanto, los tanques inferiores reciben la mayor parte del caudal y coincide con el emparejamiento sugerido por el array de ganancias relativas. Es decir, la direccionalidad de ese conjunto de puntos de funcionamiento es coherente con el conocimiento intuitivo y las medidas previamente calculadas. En el resto del mapa, tanto el de mayor como el de menor ganancia, las direcciones de entrada y salida son relativamente opuestas, de tal modo que cuando la dirección de entrada apunta a u_1, la de salida lo hace a y_2 y cuando lo hace a u_2, a y_1. Es decir, en esta región las entradas están más correlacionadas con las salidas cruzadas. Por tanto, este mapa de dinámica vuelve a coincidir, en líneas generales, con el conocimiento previo, si bien es cierto que algunos casos específicos proporcionan información confusa o contradictoria, lo cual es especialmente patente en las pequeñas zonas de fase mínima situadas en los laterales del mapa.

Los valores singulares que se acaban de presentar permiten calcular también el número de condición. Aunque este número es muy dependiente de la escala y, por tanto, su valor cuantitativo aporta poca información, cualitativamente es útil para distinguir los puntos de funcionamiento para los que el control es más

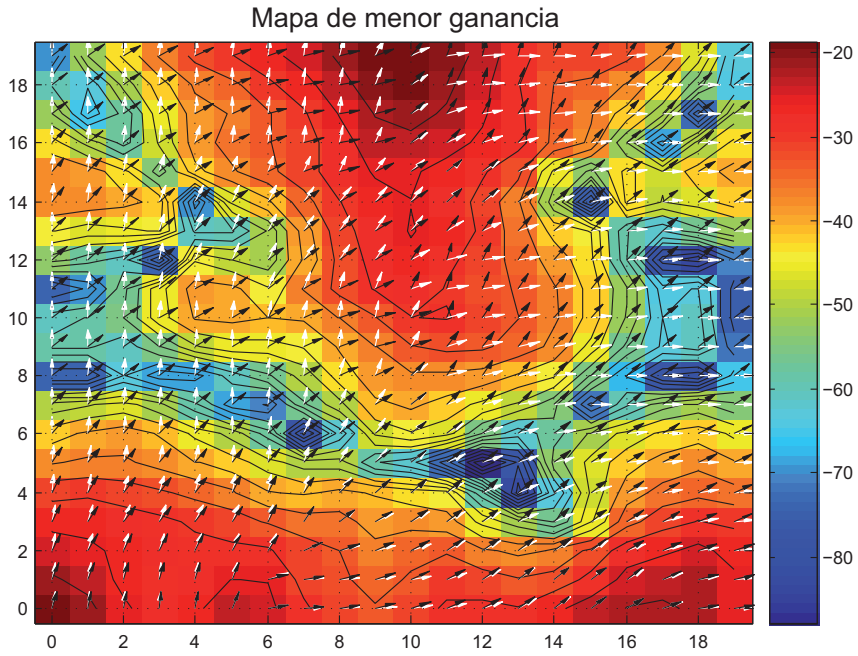


Figura 5.36. Mapa de menor ganancia, $\sigma_2(0)$, y de sus direcciones de entrada y salida asociadas.

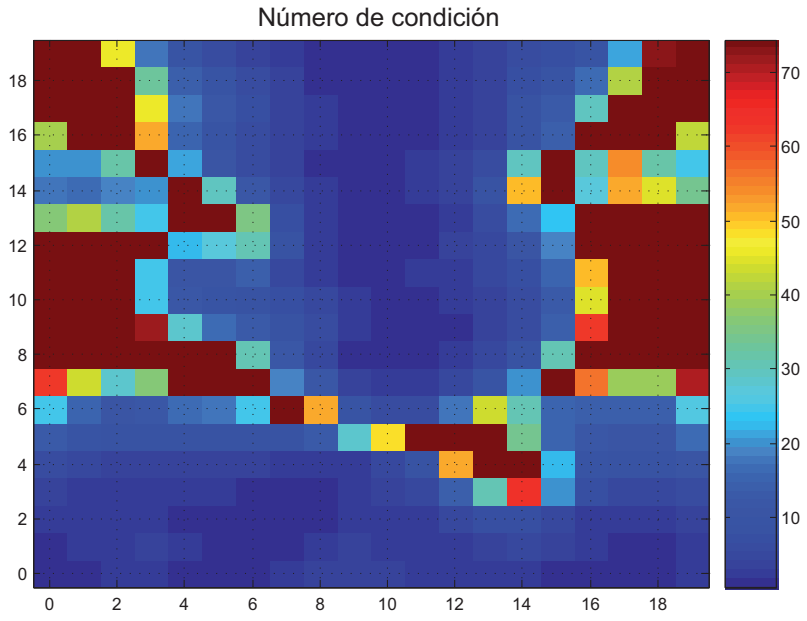


Figura 5.37. Mapa de número de condición

complicado. Su aplicación al sistema objeto de análisis (Fig. 5.37) distingue como comportamientos que dificultan el control los correspondientes a las zonas en las que la ganancia relativa (de nuevo, ver Fig. 5.33) es negativa o cercana a 2.

El resultado de este experimento es, por lo tanto, bastante satisfactorio, pues ha permitido la generación de mapas informativos que corroboran el conocimiento previo y son coherentes entre sí. Las relaciones y limitaciones puestas matemáticamente de relieve y aplicados a dos casos concretos por Johansson (2000), pueden extraerse de forma intuitiva para todo el rango de operación mediante el análisis visual de los mapas de dinámica propuestos.

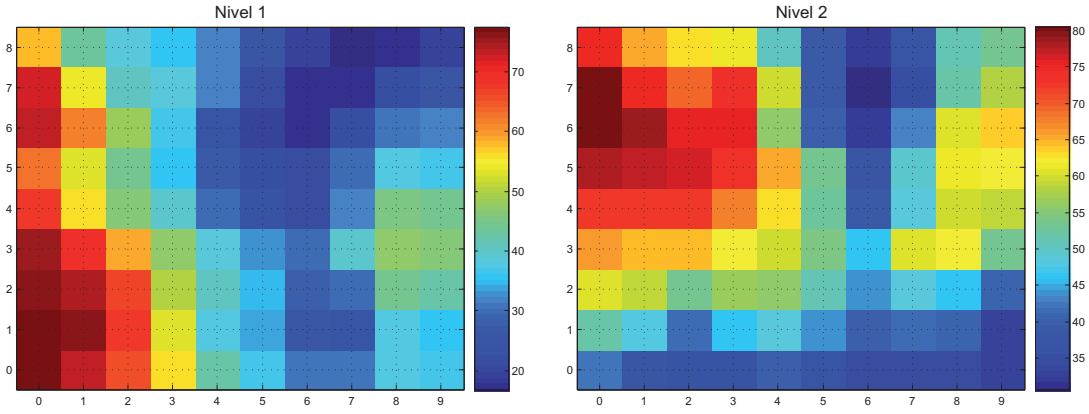
5.3.2. Experimento MIMO2: Mapas de visualización para un sistema real de 4 tanques

El conjunto de datos de entrada, previamente normalizado, se utiliza para un entrenamiento *batch* de 200 épocas de un SOM 9×10 con vecindad gaussiana decreciente. Las dimensiones del mapa han sido elegidas para minimizar la distorsión de los datos de entrada. No obstante, el error de cuantización medio del mapa es de $0,1411$, mayor que en los otros casos debido a una mayor dimensión de entrada. Los mapas de dinámica se generan a partir de los vectores *codebook* debidamente desnormalizados.

Este experimento nos da la posibilidad de comprobar la aplicabilidad de la estrategia propuesta frente a otros modelos paramétricos (en este caso matrices de transferencia discretas) y, sobre todo, frente a una situación en la que el conocimiento previo acerca del comportamiento del sistema es más incompleto y menos preciso, pues ha sido identificado a partir de muestras reales. De hecho, esa identificación no ha sido satisfactoria en todo el rango de operación por lo que algunos comportamientos podrían verse subrepresentados en los mapas de dinámica.

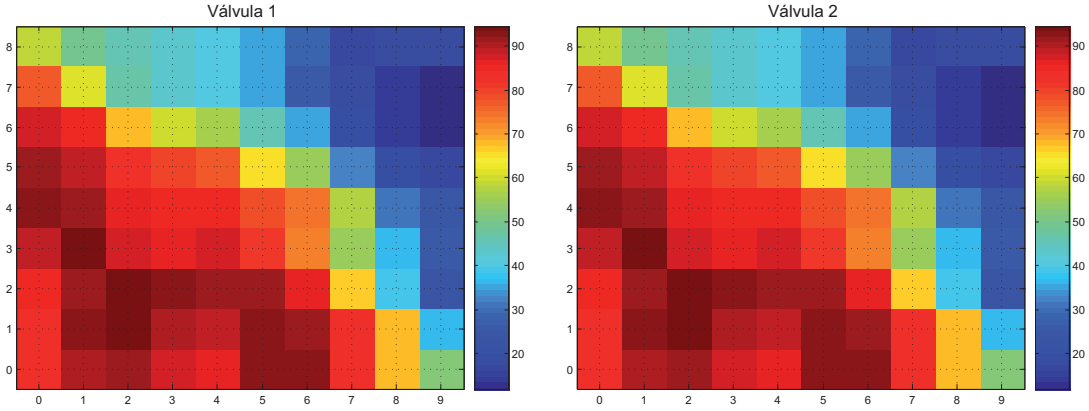
Los mapas de dinámica propuestos incluyen los mapas de ganancias relativas, los de valores singulares y el de número de condición, cuya aplicación al caso discreto es directa. Como en los experimentos anteriores, los mapas de dinámica y planos de componentes se interpretan poniendo énfasis en las conclusiones que puedan extraerse de los mismos, su relación con los otros mapas y el grado de coherencia con respecto al conocimiento previo.

Se comienza el análisis visual del sistema mostrando los planos de componentes de nivel (Fig. 5.38) y válvulas (Fig. 5.39). La suma de las aperturas de las válvulas (Fig. 5.40) determina dos regiones claramente diferenciadas: la zona inferior izquierda, en la que el caudal se dirige en mayor medida a los tanques inferiores, y la zona superior derecha, que representa los puntos de funcionamiento en que los tanques superiores reciben mayor caudal (y por tanto el control es más complejo).



(a) Nivel 1. **(b)** Nivel 2.

Figura 5.38. Plano del componente nivel.



(a) Válvula 1. **(b)** Válvula 2.

Figura 5.39. Planos de componentes de válvulas.

El resto de características, como el RGA o los valores y direcciones singulares, deberían depender en gran medida de este valor, pues se parte de la suposición de que las válvulas actúan como selectores de la dinámica del sistema.

Se interpretan, en primer lugar, los resultados relacionados con el array de ganancias relativas, calculado tanto de forma empírica a partir de los modelos identificados, $\Lambda = G(1) * G^{-T}(1)$, como teórica, a partir de su relación con las válvulas (ec. 5.1). Un simple análisis visual permite comprobar que ambos (Fig. 5.41 y Fig. 5.42) concuerdan en gran medida y muestran dos amplias regiones en las que λ vale aproximadamente 1 y 0. La zona de $\lambda \approx 1$, es decir, aquella en la que los

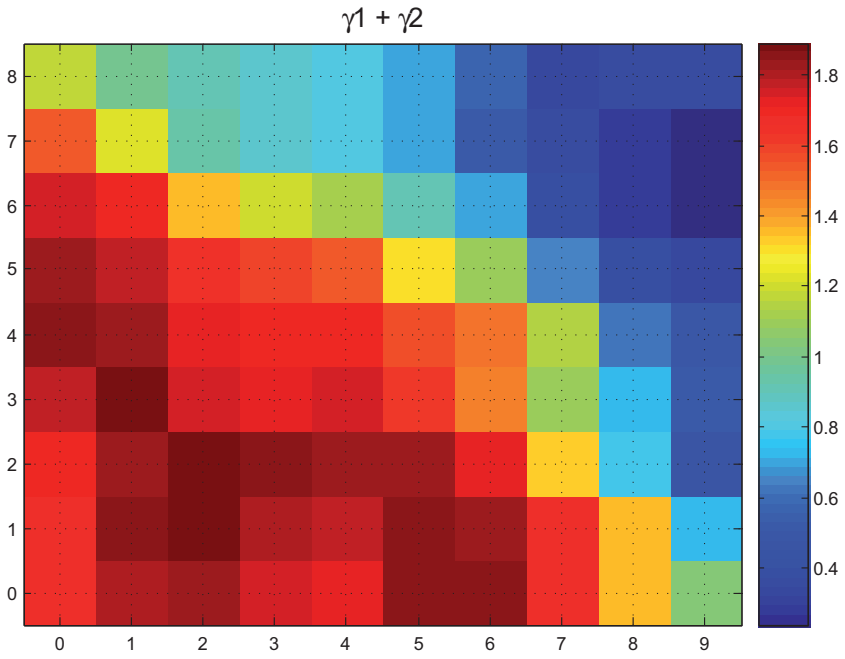


Figura 5.40. Mapa de $\gamma_1 + \gamma_2$

lazos monovariable directos son una buena elección para un control descentralizado, coincide con la zona en la que el valor de las válvulas es alto, es decir, donde el caudal se dirige principalmente a los tanques inferiores. Del mismo modo, para las aperturas de válvula en que el caudal se dirige a los tanques superiores, los mapas de RGA sugieren cruzar los lazos para poder controlar el sistema satisfactoriamente de forma descentralizada. Entre ambas regiones, existe una zona fronteriza constituida por puntos en los que el control es difícil. El mapa se simplifica con respecto al del experimento anterior, debido a que en este caso el valor de las válvulas es el mismo en cada punto de equilibrio.

Los mapas de valores y direcciones singulares para el régimen permanente (ver Fig. 5.43 y Fig 5.44) tienen una fácil interpretación en este caso. Tanto en el mapa de σ_1 como en el de σ_2, la región asociada a valores bajos de válvula (mayor caudal a los tanques inferiores) presenta mayor ganancia.

Por otra parte, las direcciones singulares muestran de forma clara la interacción entre entradas y salidas, corroborando los emparejamientos propuestos por el mapa de ganancias relativas. En el caso en que el caudal es dirigido en mayor medida a los tanques inferiores, las direcciones de entrada y de salida en ambos mapas coinciden, puesto que la mayor parte del líquido impulsado por las bombas u_1 y u_2 se vierte, respectivamente, en y_1 e y_2. En el caso contrario, el líquido procedente

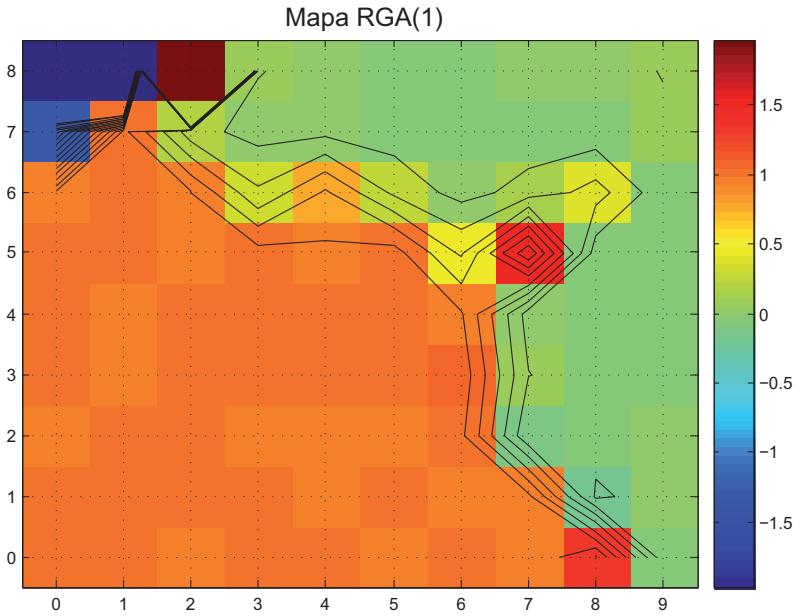


Figura 5.41. Mapa de array de ganancias relativas

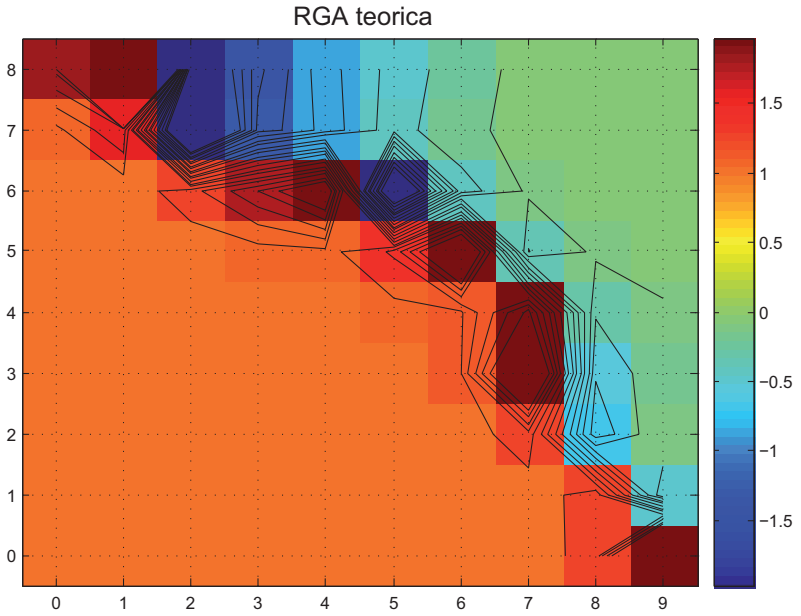


Figura 5.42. Mapa del array de ganancias relativas de acuerdo a la apertura de las válvulas

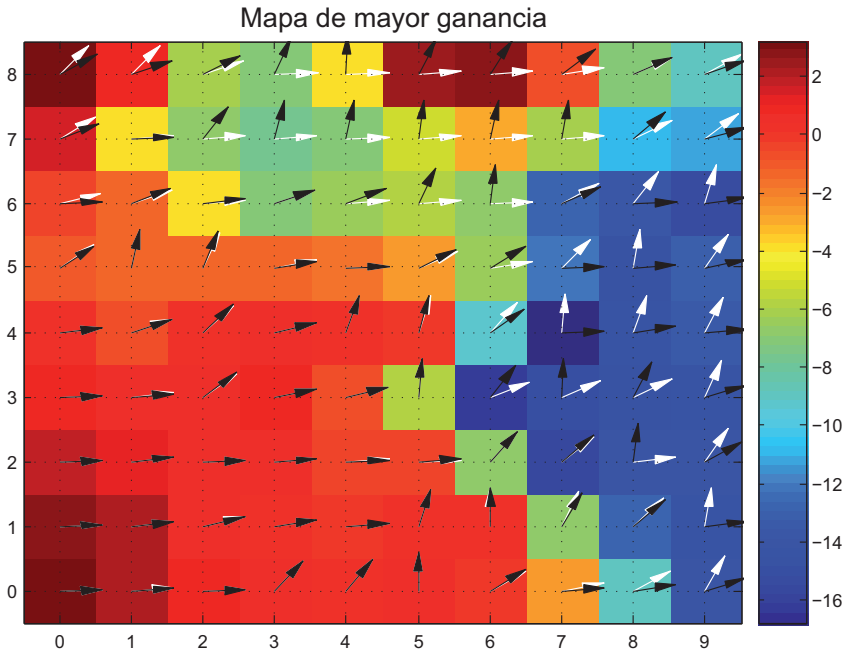


Figura 5.43. Mapa de mayor ganancia, $\sigma_1(0)$, y de sus direcciones de entrada y salida asociadas.

de u_1 se dirige principalmente al depósito y_4, que a su vez vierte en el depósito y_2, mientras que el impulsado por u_2 se dirige al tanque y_3, que vierte sobre el y_1. Por eso, las direcciones de entrada y salida son opuestas, porque existe una fuerte interacción cruzada.

Finalmente, el mapa de número de condición (ver Fig. 5.45) debe indicar de forma cualitativa en qué zonas es más complicado el control. El mapa revela un comportamiento razonable en todo el rango de operación salvo en una pequeña zona situada en el extremo superior izquierdo, en la que el RGA teórico y empírico proporcionan valores que, aun siendo contradictorios, indican dificultades en el control.

El método ha proporcionado, una vez más, buenos mapas de visualización para comprender los dos grandes tipos de comportamientos presentes en el sistema. En las zonas de operación intermedias, los mapas de dinámica han proporcionado una información visual más pobre, afectados por la fallida identificación en ciertos puntos y la disminución en el número de puntos de equilibrio considerados que causó. No obstante, las conclusiones extraídas siguen siendo informativas y coherentes.

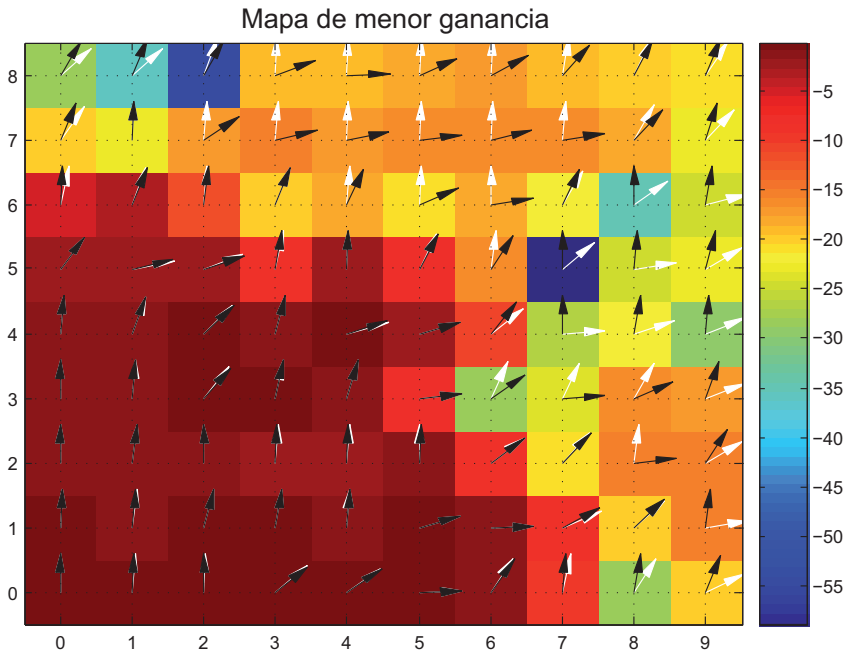


Figura 5.44. Mapa de menor ganancia, $\sigma_2(0)$, y de sus direcciones de entrada y salida asociadas.

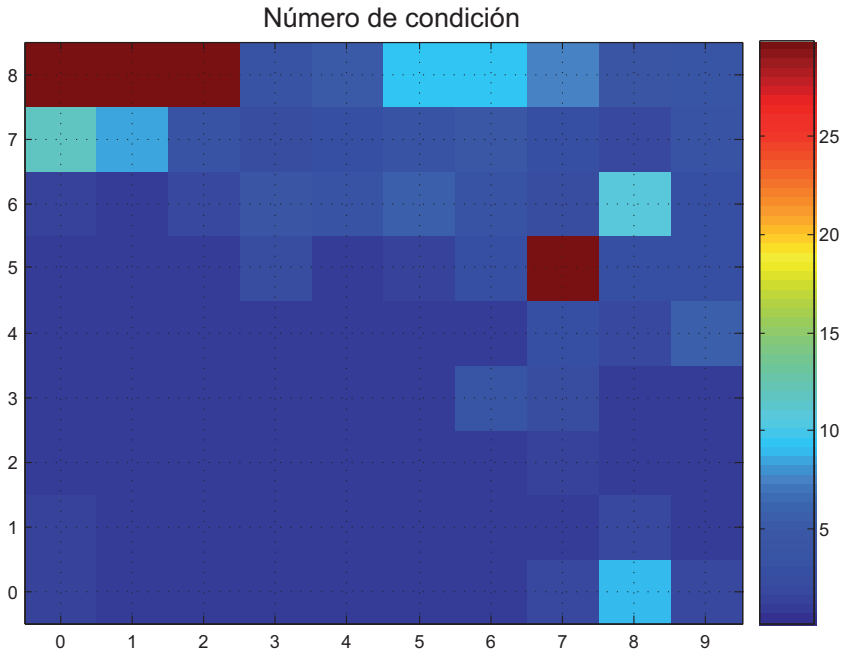


Figura 5.45. Mapa de número de condición

Capítulo 6

Conclusiones y líneas futuras

6.1. Conclusiones

Considerando el hecho de que los esfuerzos de investigación se enmarcan en el uso del algoritmo SOM para la extracción de conocimiento de la dinámica de procesos industriales a partir de los datos históricos, las conclusiones se pueden agrupar en dos vertientes: la comparación de algoritmos de modelado y la definición de mapas de visualización.

Con respecto a la comparación de modificaciones del SOM para procesamiento temporal en cuanto a su capacidad para modelar la dinámica, se pueden extraer las siguientes conclusiones:

- La comparación permite determinar pautas para la correcta selección de algoritmos y parámetros en el modelado de la dinámica. De igual manera, permite reconocer qué modificaciones del algoritmo son más útiles para este propósito, un aspecto clave para poder definir nuevos algoritmos orientados a este objetivo.

- De entre las modificaciones del SOM estudiadas, la difusión de la actividad del SOMTAD, el énfasis en secuencias difíciles del SOM-DL, la recursividad del método MSOM y la exclusión de la activación de SARDNET son las que ofrecen mejores resultados en cuanto a error de predicción, mientras que el RSOM presenta como principal desventaja la dificultad de ajuste de sus parámetros.

- Aunque todas las modificaciones preservan los invariantes satisfactoriamente, la red SOMTAD es la menos afectada por sobreajuste al objetivo de predicción.

- La inclusión de nuevos parámetros libres, muy importantes para su rendimiento final, dificulta, no obstante, la correcta aplicación de estos algoritmos, pues se requiere un proceso de selección previa de los mismos.
- Se concluye que el ajuste del modelo por medio de vectores prototipo proporciona resultados inferiores al ajuste por medio de los vectores de la región de Voronoi.

En cuanto a los mapas de visualización definidos para el análisis y la supervisión de la dinámica de procesos industriales, se pueden enumerar las siguientes conclusiones:

- Los mapas permiten descubrir, ampliar o confirmar conocimientos acerca del comportamiento del sistema a un nivel global, que puede abarcar todo el rango de operación.
- Debido a su consistente modo de visualización, se puede establecer una comparación visual entre características o con respecto a las variables que definen el punto de funcionamiento. Esto permite comprender las correlaciones presentes en el sistema.
- Los mapas también son consistentes con otros esfuerzos previos en el área de visualización de procesos industriales por medio del SOM y pueden ser utilizados conjuntamente.
- La visualización de mapas de la respuesta temporal puede dar lugar al descubrimiento de relaciones no explícitas de forma intuitiva y proporciona una visión general del sistema. Por ello, puede permitir al ingeniero distinguir qué zonas y rangos presentan una dinámica adecuada para sus propósitos de control o son de interés para un análisis más a fondo.
- La visualización, en todo el rango de operación, de características que se complementan en el análisis de sistemas multivariable, puede proporcionar al ingeniero una idea muy completa y precisa de la naturaleza de la interacción entre lazos y de las situaciones que dificultan el control.
- Como en cualquier técnica basada en datos, la calidad de la visualización o modelado depende de los mismos. Su utilización, por tanto, debe decidirse en función de la disponibilidad y coste de obtención de datos frente a la posibilidad y coste del desarrollo de modelos analíticos. No obstante, estas no deben entenderse como dos posibilidades confrontadas, pues la visualización de sistemas simulados en un amplio rango de funcionamiento también puede contribuir a extraer nuevas conclusiones acerca del comportamiento del sistema.

6.2. Aportaciones

Las principales contribuciones que presenta este trabajo son:

- La definición de mapas de visualización por medio del SOM de características relevantes de la respuesta de sistemas monovariable —mapas de ganancia, de tiempo de establecimiento, de tiempo de pico, de tiempo de subida, de sobreoscilación, de coeficiente de amortiguamiento y de frecuencia natural—.

- La definición de mapas de visualización de características relacionadas con la dificultad en el control, direccionalidad y acoplamiento entre variables en sistemas multivariable —mapas de ganancia relativa, de valores y direcciones singulares, de condición y de ceros multivariable—.

- Definición de una nueva regla de propagación de la mejora para el uso del algoritmo SOMTAD con un espacio de salida de dos dimensiones.

- Aplicación de los algoritmos SARDNET, SOMTAD y MSOM a la predicción de series temporales basada en el modelado local de la dinámica.

- Comparación, en base al error de predicción y los invariantes de la dinámica, de diversas variaciones temporales del SOM propuestas en la bibliografía y de las dos técnicas propuestas para el ajuste de los modelos.

6.3. Líneas futuras

A la luz de lo presentado en esta tesis, se podrían citar como futuras líneas de investigación las siguientes:

- Definición de nuevos mapas de visualización que permitan un análisis visual de otras características relevantes del sistema como, por ejemplo, la sensibilidad (valor de las funciones de sensibilidad a una determinada frecuencia o valor de pico de S), los polos multivariable y sus direcciones, el criterio de Niederlinski, etc.

- Estudio de los métodos de visualización aplicados a la supervisión en línea. Incluye la definición de mapas de visualización específicos para esta tarea y el seguimiento del comportamiento dinámico actualizado del proceso. Esta línea podría, asimismo, derivar en la investigación de métodos y algoritmos de modelado local adaptativos para sistemas no estacionarios.

- Elaboración de una metodología integral de análisis y supervisión de procesos mediante técnicas de visualización y reducción de la dimensión. Estudio del alcance y compatibilidad de las herramientas propuestas y desarrollo de un software para la aplicación conjunta de las técnicas basadas en SOM de visualización de características estáticas y dinámicas para el análisis y/o supervisión de procesos industriales.

- Aplicación conjunta del modelado de la dinámica y la visualización de la misma por medio del SOM de acuerdo a las conclusiones extraídas de este trabajo. Evaluación de los procedimientos y algoritmos disponibles con respecto a su idoneidad para visualización.

- Planteamiento de nuevas modificaciones espacio-temporales del SOM para el modelado dinámico que combinen la regulación del aprendizaje en función del error con mecanismos de memoria a corto plazo como recursividad o difusión de la actividad y resuelvan el problema de la selección de parámetros.

- Estudio de la aplicabilidad de la visualización de características dinámicas como ayuda para el diseño de controladores basados en el modelado local lineal de un sistema mediante el SOM.

Bibliografía

Abonyi, J., S. Nemeth, V. Csaba y P. A. J. Vesanto. "Process analysis and product quality estimation by self-organizing maps with an application to polyethylene production." *Computers in Industry*, 52(3), 221–234, 2003.

Ahola, J., E. Alhoniemi y O. Simula, "Monitoring industrial processes using the self-organizing map". En *SMCia/99 Proceedings of the 1999 IEEE Midnight–Sun Workshop on Soft Computing Methods in Industrial Applications*, 22–27, IEEE Service Center, Piscataway, NJ, 1999.

Alahakoon, D., S. K. Halgamuge y B. Srinivasan. "Dynamic self-organizing maps with controlled growth for knowledge discovery". *IEEE Transactions on Neural Networks*, 11(3), 601–614, Mayo 2000.

Albertos, P. y A. Sala. *Multivariable Control Systems: An Engineering Approach*. Springer, London, 2004.

Alhoniemi, E. "Analysis of pulping data using the Self-Organizing Map". *TAPPI Journal*, 83(7), 66, Julio 2000.

Alhoniemi, E., J. Hollmén, O. Simula y J. Vesanto. "Process monitoring and modeling using the self-organizing map". *Integrated Computer-Aided Engineering*, 6(1), 3–14, 1999.

Alpaydin, E. *Introduction to Machine Learning*. MIT Press, Cambridge, MA, 2004.

Arahal, M. R., M. Berenguel, E. F. Camacho y F. Pavón. "Selección de variables en la predicción de llamadas en un centro de atención telefónica." *Revista Iberoamericana de Automática e Informática Industrial*, 6(1), 94–104, 2009a.

Arahal, M. R., A. Reyes, I. Alvarado y F. Rodríguez. "Agrupaciones de modelos locales con descripción externa. aplicación a una planta de frío solar". *Revista Iberoamericana de Automática e Informática Industrial*, 6(1), 51–62, 2009b.

Bakker, R., J. C. Schouten, M.-O. Coppens, F. Takens, C. L. Giles y C. M. van den Bleek, "Robust Learning of Chaotic Attractors". En S. Solla, T. Leen y K.-R. Muller (editores), *Advances in Neural Information Processing Systems 12*, 879–885, MIT Press, 2000.

Barreto, G. A., "Time Series Prediction with the Self-organizing map: A Review." En P. Hitzler y B. Hammer (editores), *Perspectives on Neural-Symbolic Integration*, Spring-Verlag, 2007.

Barreto, G. A. y A. F. R. Araújo. "Time in self-organizing maps: An overview of models". *International Journal of Computer Research*, 10(2), 139–179, 2001.

Barreto, G. A. y A. F. R. Araujo. "Identification and control of dynamical systems using the self-organizing map". *IEEE Transactions on Neural Networks*, 15(5), 1244–1259, 2002.

Barreto, G. A., J. C. M. Mota, L. G. M. Souza y R. A. Frota, "Nonstationary time series prediction using local models based on competitive neural networks". En *IEA/AIE 2004: Proceedings of the 17th international conference on Innovations in applied artificial intelligence*, 1146–1155, Springer Springer Verlag Inc, 2004. ISBN 3-540-22007-0.

Bauer, H. U., M. Herrmann y T. Villmann. "Neural maps and topographic vector quantization". *Neural Networks*, 12(4), 659–676, 1999.

Belkin, M. y P. Niyogi. "Laplacian eigenmaps for dimensionality reduction and data representation". *Neural Computation*, 15, 1373–1396, 2003.

Bishop, C. M., M. Svensen y C. K. I. Williams. "GTM: The generative topographic mapping". *Neural Computation*, 10(1), 215–234, 1998.

Carreira-Perpiñán, M. Á., *Continuous Latent Variable Models for Dimensionality Reduction and Sequential Data Reconstruction*. Tesis Doctoral. Department of Computer Science, University of Sheffield, UK, 2001.

Carreira-Perpiñán, M. Á., "A review of dimension reduction techniques". Informe Técnico CS–96–09, Dept. of Computer Science, University of Sheffield, Diciembre 1996.

Casdagli, M. "Nonlinear prediction of chaotic time series". *Physica D*, 35, 335–356, 1989.

Chapman, P., J. Clinton, R. Kerber, T. Khabaza, T. Reinartz, C. Shearer y R. Wirth, *CRISP-DM 1.0: Step-by-Step Data Mining Guide*. SPSS, 2000.

Chappell, G. J. y J. G. Taylor. "The Temporal Kohonen Map". *Neural Networks*, 6, 441–445, 1993.

Cherkassky, V. y F. Mulier. *Learning from Data – Concepts, Theory and Methods*, 2ª edición. John Wiley & Sons, New York, 2007.

Cho, J., J. C. Principe, D. Erdogmus y M. A. Motter. "Modeling and inverse controller design for an unmaned aerial vehicle based on the self-organizing map." *IEEE Trans. Neural Networks*, 17(2), 445–460, 2006.

Claussen, J. C. "Winner-Relaxing Self-Organizing Maps". *Neural Comp.*, 17(5), 996–1009, 2005.

Cottrell, M., J.-C. Fort y G. Pagès. "Theoretical aspects of the SOM algorithm." *Neurocomputing*, 21(1-3), 119–138, 1998.

Crutchfield, J. P. y B. S. McNamara. "Equations of motion from a data series". *Complex Systems*, 1, 417–452, 1987.

Cuadrado Vega, A. A., *Supervisión de Procesos Complejos mediante Técnicas de Data Mining con Incorporación de Conocimiento Previo*. Tesis Doctoral. Universidad de Oviedo, Oviedo, España, 2002.

De, A. y N. Chatterjeee. "Recognition of impulse fault patterns in transformers using Kohonen's Self–Organizing Feature Map." *IEEE Transactions on Power Delivery*, 17(2), 489–494, 2002.

Demartines, P. y J. Hérault. "Curvilinear component analysis: a self organizing neural network for non linear mapping of data sets". *IEEE Transactions on Neural Networks*, 8, 148–154, 1997.

Díaz, I., A. A. Cuadrado, A. B. Díez, J. J. Fuertes, M. Domínguez y P. Reguera, "Visualization of dynamics using local dynamic modelling with Self Organizing Maps." En J. Marques de Sá, W. Duch y L. A. Alexandre (editores), *International Conference on Artificial Neural Networks 2007, Part I, Lecture Notes in Computer Science*, tomo 4668, 609–617, Springer-Verlag, 2007. ISBN 978-3-540-74689-8.

Díaz, I., A. A. Cuadrado, A. B. Díez, F. Obeso y J. A. Rodríguez. "Visual predictive maintenance tool based on SOM projection techniques." *Revue de Métallurgie*, 103(3), 307–315, 2003.

Díaz, I., A. A. Cuadrado, A. B. Díez y G. Ojea. "Modelado visual de procesos industriales." *Revista Iberoamericana de Automática e Informática Industrial*, 2(4), 101–112, 2005.

Díaz, I., A. B. Díez, A. A. Cuadrado y J. Enguita, "RBF approach for trajectory interpolation in Self-organizing map based condition monitoring". En *1999 7th IEEE International Conference on Emerging Technologies and Factory Automation. Proceedings ETFA '99.*, tomo 2, 1003–10, IEEE Service Center, Piscataway, NJ, 1999.

Díaz, I., M. Domínguez, A. A. Cuadrado y J. J. Fuertes. "A new approach to exploratory analysis of system dynamics using SOM. Applications to industrial processes." *Expert Systems with Applications.*, 34(4), 2953–2965, Mayo 2008.

Díaz, I. y J. Hollmén, "Residual generation and visualization for understanding novel process conditions." En *Proc. of International Joint Conference on Neural Networks (IJCNN 2002)*, tomo 3, 2070–2075, IEEE, Piscataway, 2002.

Dittenbach, M., D. Merkl y A. Rauber, "Growing hierarchical self-organizing map". En *Proceedings of the International Joint Conference on Neural Networks*, tomo 6, 15–19, Technische Universitat Wien, IEEE, Piscataway, NJ, 2000.

Domínguez, M., *Supervisión remota de procesos complejos vía Internet mediante Técnicas de Data Mining Visual. Aplicación a una planta piloto industrial.* Tesis Doctoral. Universidad de Oviedo, Oviedo, España, 2003.

Domínguez, M., J. J. Fuertes, P. Reguera, J. J. González y J. M. Ramón. "Maqueta industrial para docencia e investigación". *Revista Iberoamericana de Automática e Informática Industrial*, 1(2), 58–63, 2004.

Domínguez, M., J. J. Fuertes, P. Reguera, M. A. Prada y A. Morán, "Inter-university network of remote laboratories." En *17th IFAC World Congress*, Seoul, 2008.

Domínguez, M., P. Reguera y J. J. Fuertes. "Laboratorio Remoto para la enseñanza de la Automática en la Universidad de León (España)". *Revista Iberoamericana de Automática e Informática Industrial*, 2(2), 36–45, 2005.

Domínguez, M., P. Reguera, J. J. Fuertes, I. Díaz y A. A. Cuadrado. "Internet-based remote supervision of industrial processes using Self-organizing maps." *Engineering Applications of Artificial Intelligence*, 20(6), 757–765, Septiembre 2007.

Donoho, D. L. y C. Grimes, "Hessian eigenmaps: Locally linear embedding techniques for high-dimensional data". En *Proceedings of the National Academy of Sciences*, 100, 5591–5596, 2003.

Dormido, S., "Control learning: Present and future." En *IFAC Annual Control Reviews*, tomo 28, 115–136, New York, 2004.

Dubrawsky, I., C. T. Baumrucker, J. Caesar, M. Krishnamurthy, T. W. Shinder, B. Pinkard, E. Seagren *et al. Designing and Building Enterprise DMZs.* Elsevier Inc., 2006.

Erdogmus, D., J. Cho, J. Lan, M. Motter y J. C. Principe, "Adaptive local linear modelling and control of nonlinear dynamical systems". En A. Ruano (editor), *Intelligent Control Systems Using Computational Intelligence Techniques*, 1–36, IEE Control Series, London, UK, 2005.

Erwin, E., K. Obermayer y K. Schulten. "Self-organizing maps: Order, convergence properties and energy functions". *Biol. Cyb.*, 67, 47–55, 1992.

Euliano, N. R., *Temporal Self-Organization for Neural Networks.* Tesis Doctoral. University of Florida, Gainesville, FL, 1998.

Farmer, J. y J. Sidorowich. "Predicting chaotic time series". *Phys. Rev. Lett.*, 59, 845–8, 1987.

Fayyad, U. M., G. Piatetsky-Shapiro y P. Smyth. "From data mining to knowledge discovery in databases". *AI Magazine*, 17(3), 37–54, 1996.

Ferreira de Oliveira, M. C. y H. Levkowitz. "From visual data exploration to visual data mining: A survey". *IEEE Transactions on Visualization and Computer Graphics*, 9(3), 378–394, 2003.

Flexer, A. "On the use of self-organizing maps for clustering and visualization". *Intelligent-Data-Analysis*, 5, 373–84, 2001.

Frey, C. W. "Process diagnosis and monitoring of field bus based automation systems using self-organizing maps and watershed transformations". 620–625, Aug. 2008.

Fritzke, B. "Growing cell structures—a self-organizing network for unsupervised and supervised learning". *Neural Networks*, 7(9), 1441–1460, 1994.

Fuertes, J. J., *Modelado y Representación de la Dinámica de Sistemas Complejos mediante Técnicas de Data Mining Visual para la Supervisión Remota de Procesos Industriales vía Internet.* Tesis Doctoral. Universidad de Oviedo, Oviedo, España, 2006.

Fuertes, J. J., M. A. Prada, M. Domínguez, P. Reguera, I. Díaz y A. A. Cuadrado, "Modeling of Dynamics using Process State Projection on the Self Organizing Map." En J. Marques de Sá, W. Duch y L. A. Alexandre (editores), *International Conference on Artificial Neural Networks 2007, Part I, Lecture Notes in Computer Science*, tomo 4668, 589–598, Springer-Verlag, 2007. ISBN 978-3-540-74689-8.

Fuertes, J. J., P. Reguera, M. Domínguez, I. Díaz y A. A. Cuadrado, "Industrial Supervision System based on Visual Data Mining and Motion Trajectory Analysis." En *International Federation of Automatic Control 16th IFAC World Congress*, Prague, 2005.

Furao, S. y O. Hasegawa. "An incremental network for on-line unsupervised classification and topology learning". *Neural Networks*, 1(19), 90–106, 2006.

Gamma, E., R. Helm, R. Johnson y J. Vlissides. *Design Patterns: Elements of Reusable Object-Oriented Software*. Addison-Wesley, 1995. ISBN 0-201-63361-2.

Gencay, R. y W. D. Dechert. "An algorithm for the n Lyapunov exponents of an n-dimensional unknown dynamical system". *Physica D*, 59, 142–157, 1992.

Gershenfeld, N. A. y A. Weigend, "The future of time series: Learning and understanding". En A. S. Weigend y N. A. Gershenfeld (editores), *Time Series Prediction: Forecasting the Future and Understanding the Past*, 1–70, Addison Wesley, Reading, MA, 1994.

Gertler, J. J. "Fault detection and isolation using parity relations". *Control Engineering Practice*, 5(5), 653–661, 1997.

Graepel, T., M. Burger y K. Obermayer. "Self-organizing maps: generalizations and new optimization techniques". *Neurocomputing*, (21), 173–190, 1998.

Grassberger, P. y I. Procaccia. "Measuring the strangeness of strange attractors". *Physica*, 9, 189–208, 1983.

Gray, R. M. y D. L. Neuhoff. "Quantization". *IEEE Transactions on Information Theory*, 44(6), 1–63, 1998.

Grossman, R., S. Bailey, A. Ramu, B. Malhi, P. Hallstrom, I. Pulleyn y X. Qin. "The management and mining of multiple predictive models using the predictive modeling markup language." *Information & Software Technology*, 41(9), 589–595, 1999.

Guimarães, G., V. Sousa-Lobo y F. Moura-Pires. "A taxonomy of self-organizing maps for temporal sequence processing." *Intelligent Data Analysis*, (4), 269–290, 2003.

Hagenbuchner, M., A. Sperduti y A. C. Tsoi. "A self-organizing map for adaptive processing of structured data." *IEEE Transactions on Neural Networks*, 14(3), 491–505, 2003.

Hammer, B., A. Micheli, A. Sperduti y M. Strickert. "Recursive self-organizing network models." *Neural Networks*, 17, 1061–1085, 2004.

Hammer, B. y T. Villmann, "Mathematical aspects of neural networks". En M. Verleysen (editor), *European Symposium of Artificial Neural Networks 2003*, 59–72, 2003.

Hastie, T. y W. Stuetzle. "Principal curves". *J. of Am. Stat. Assoc.*, 84, 502–516, 1989.

Haykin, S. *Neural Networks: A Comprehensive Foundation*. Macmillan Co., New York, 1994. ISBN 0-13-273350-1.

Haykin, S. y J. Principe. "Making sense of a complex world [Chaotic events modeling]". *IEEE Signal Processing Magazine*, 15(3), 66–81, 1998.

He, X. y H. Asada, "A new method for identifying orders of input-output models for nonlinear dynamic systems". En *Proceedings of ACC93*, 2520–2523, 1993.

Hegger, R., H. Kantz y T. Schreiber. "Practical implementation of nonlinear time series methods: The TISEAN package". *Chaos*, 9(2), 413–435, 1999.

Hénon, M. "A two-dimensional mapping with a strange attractor". *Communications in Mathematical Physics*, 50, 69–77, 1976.

Heskes, T., "Energy functions for self-organizing maps". En E. Oja y S. Kaski (editores), *Kohonen Maps*, 303–316, Elsevier, Amsterdam, 1999.

Heskes, T. "Self-organizing maps, vector quantization, and mixture modeling". *IEEE Transactions on Neural Networks*, 12(6), 1299–1305, 2001.

Himberg, J., "Enhancing the SOM-based data visualization by linking different data projections". En *Proceedings of 1st International Symposium IDEAL'98, Intelligent Data Engineering and Learning–Perspectives on Financial Engineering and Data Mining*, 427–434, Springer, Hong Kong, 1998.

Hinton, G. E. y S. Roweis, "Stochastic neighbor embedding." En T. S. Becker y K. Obermayer (editores), *Advances in Neural Information Processing Systems*, tomo 15, 833–840, MIT Press, Cambridge, MA, 2002.

Hollmén, J., *Process Modeling Using the Self-Organizing Map*. Tesis de Master. Helsinki University of Technology, Espoo, Finland, 1996.

Hollmén, J. y O. Simula, "Prediction models and sensitivity analysis of industrial production process parameters by using the self-organizing map." En *Proceedings of IEEE Nordic Signal Processing Symposium(NORSIG96)*, 79–82, 1996.

Hynna, K. I. y M. Kaipainen. "Activation-based recursive self-organizing maps: A general formulation and empirical results". *Neural Processing Letters*, 24, 119–136, 2006.

Hyvärinen, A., J. Karhunen y E. Oja. *Independent Component Analysis*. John Wiley & Sons, 2001.

Isermann, R. "Model-based fault-detection and diagnosis - status and applications." *Annual Reviews in Control*, 29(1), 71–85, 2005.

Isermann, R. y P. Ballé. "Trends in the application of model-based fault detection and diagnosis of technical processes". *Control Engineering Practice*, 5(5), 709–719, 1997.

Jaeger, H. "Harnessing nonlinearity: Predicting chaotic systems and saving energy in wireless communication". *Science*, 304, 78–80, 2004.

James, D. L. y R. Miikkulainen, "SARDNET: A self-organizing feature map for sequences". En G. Tesauro, D. S. Touretzky y T. K. Leen (editores), *Advances in Neural Information Processing Systems 7, [NIPS Conference, Denver, Colorado, USA, 1994]*, 577–584, MIT Press, 1995.

Jamsa-Jounela, S.-L., M. Vermasvuori, S. Haavisto y P. Enden. "A process monitoring system based on the Kohonen self-organizing maps." *Control Engineering Practice*, 11(1), 83–92, 2003.

Johansson, K. H. "The Quadruple–Tank Process: A Multivariable Laboratory Process with an Adjustable Zero." *IEEE Transaction on Control Systems Technology*, 8(3), 456–465, 2000.

Kangas, J., *On the Analysis of Pattern Sequences by Self-Organizing Maps*. Tesis Doctoral. Helsinki University of Technology, Espoo, Finland, 1994.

Kantz, H. "A robust method to estimate the maximal Lyapunov exponent of a time series." *Phys. Lett. A*, 185(77), 1994.

Kantz, H. y T. Schreiber. *Nonlinear Time Series Analysis*, 2ª edición. Cambridge University Press, 2003.

Kaski, S., *Data Exploration Using Self-Organizing Maps*. Tesis Doctoral. Helsinki University of Technology, Espoo, Finland, 1997.

Kaski, S., J. Kangas y T. Kohonen. "Bibliography of self-organizing map (SOM) papers: 1981–1997". *Neural Computing Surveys*, 1(3 & 4), 1–176, 1998.

Kasslin, M., J. Kangas y O. Simula, "Process state monitoring using self-organizing maps". En I. Aleksander y J. Taylor (editores), *Artificial Neural Networks, 2*, tomo II, 1531–1534, North-Holland, Amsterdam, Netherlands, 1992.

Keim, D. A. "Visual exploration of large data sets." *Communications ACM*, 44(8), 38–44, 2001.

Keim, D. A. "Information visualization and visual data mining." *IEEE Trans. Vis. Comput. Graph.*, 8(1), 1–8, 2002.

Kohonen, T. "Self-organizing formation of topologically correct feature maps". *Biol. Cyb.*, 43(1), 59–69, 1982.

Kohonen, T., "The self-organizing map." En *Proceedings of the IEEE*, tomo 78, 1464–1480, 1990.

Kohonen, T., "Generalizations of the self-organizing map". En *Proc. IJCNN-93, International Joint Conference on Neural Networks, Nagoya*, tomo 1, 457–462, IEEE Service Center, 1993.

Kohonen, T. *Self-Organizing Maps*, 3ª edición. Springer-Verlag New York, Inc., Secaucus, NJ, USA, 2001. ISBN 3–5406–7921–9.

Kohonen, T., E. Oja, O. Simula, A. Visa y J. Kangas. "Engineering applications of the self-organizing map". *Proceedings of the IEEE*, 84(10), 1358–84, 1996.

Koskela, T., M. Varsta, J. Heikkonen y K. Kaski. "Temporal sequence processing using Recurrent SOM". *Proc. Second International Conference on Knowledge-Based Intelligent Electronic Systems*, 1, 290–297, Abril 1998.

Kramer, M. "Nonlinear principal component analysis using autoassociative neural networks". *AIChE Journal*, 37, 233–243, 1991.

Laine, J., "Analysis and monitoring of continuous casting mould with the self-organizing map". En *Proceedings of the ECSC workshop on Application of artificial neural network systems in the steel industry*, Bruselas, Bélgica, 1998.

Lampinen, J. y T. Kostiainen, "Self-organizing map in data-analysis notes on overfitting and overinterpretation." En *Proceedings of ESANN 2000*, 239–244, 2000.

Lampinen, J. y T. Kostiainen, "Generative probability density model in the self-organizing map." En U. Seiffert y L. Jain (editores), *Self-organizing neural networks: Recent advances and applications*, 75–94, 2002.

Lau, K. W., H. Yin y S. Hubbard. "Kernel self-organising maps for classification". *Neurocomputing*, 69, 2033–2040, 2006.

Lendasse, A., J. Lee, V. Wertz y M. Verleysen. "Forecasting electricity consumption using nonlinear projection and self-organizing maps". *Neurocomputing*, 48(1), 299–311(13), Octubre 2002.

Linde, Y., A. Buzo y R. Gray. "An algorithm for vector quantizer design". *IEEE Transactions on Communications*, COM-28, 84–95, 1980.

Linsker, R. "How to generate ordered maps by maximizing the mutual information between input and output signals". *Neural Computation*, (1), 402–411, 1989.

Ljung, L. *System Identification: Theory for the User, Second Edition*. Prentice-Hall, Englewood Cliffs, NJ, 1999.

Ljung, L. "Identification for control: simple process models". *Proceedings of the 41st IEEE Conference on Decision and Control, 2002*, 4, 4652–4657, Dec. 2002.

Machón, I. y H. López. "End-point detection of the aerobic phase in a biological reactor using SOM and clustering algorithms." *Engineering Applications of Artificial Intelligence*, 19(1), 19–28, 2006.

Mackey, M. y L. Glass. "Oscillation and chaos in a physiological control system". *Science*, 197(287), 1977.

MacQueen, J., "Some methods for classification and analysis of multivariate observations". En L. LeCam y J. Neyman (editores), *Proceedings of the Fifth Berkeley Symposium on Mathematical Statistics and Probability. Volume I: Statistics*, 281–297, University of California Press, Berkeley and Los Angeles, CA, 1967.

Martinetz, T. M., S. G. Berkovich y K. J. Schulten. "'Neural-gas' network for vector quantization and its application to time-series prediction". *IEEE Trans. on Neural Networks*, 4(4), 558–569, 1993.

McNames, J., *Innovations in Local Modeling for Time Series Prediction.* Tesis Doctoral. Stanford University, Palo Alto, CA, 1999.

McNames, J. "Local averaging optimization for chaotic time series prediction". *Neurocomputing*, 48(1–4), 279–297, 2002.

Mulier, F. y V. Cherkassky. "Self-organization as an iterative kernel smoothing process". *Neural Computation*, 7(6), 1165 – 1177, 1995.

Narendra, K. S. y K. Parthasarathy. "Identification and control of dynamical systems using neural networks". *IEEE Trans. Neural Networks*, 1(1), 4–27, 1990.

Nickerson, J. V., J. E. Corter, S. K. Esche y C. Chassapis. "A model for evaluating the effectiveness of remote engineering laboratories and simulations in education". *Computers and Education*, 49(3), 708–725, 2007.

Nørgaard, M., "Neural network based system identification toolbox, version 2". Informe Técnico 00-E-891, Department of Automation, Technical University of Denmark, Diciembre 2000.

Ogata, K. *Modern Control Engineering*, 4ª edición. Prentice Hall, 2001. ISBN 0-13-060907-2.

Oja, M., S. Kaski y T. Kohonen. "Bibliography of self-organizing map (SOM) papers: 1998–2001 addendum". *Neural Computing Surveys*, 3, 1–156, 2003.

Patton, R. J. y J. Chen. "Observer-based fault detection and isolation: Robustness and applications." *Control Engineering Practice*, 5(5), 671–682, 1997.

Pölzlbauer, G., M. Dittenbach y A. Rauber. "Advanced visualization of Self-Organizing Maps with vector fields". *Neural Networks*, (19), 911–922, 2006.

Postolache, O. A., P. M. B. Silva-Girao, J. M. Dias-Pereira y H. M. Geirinhas-Ramos. "Self-organizing maps application in a remote water quality monitoring systems." *IEEE Transactions on Instrumentation and Measurement*, 54(1), 322–329, 2005.

Principe, J., N. Euliano y S. Garani. "Principles and networks for self-organization in space-time." *Neural Networks*, (15), 1069–1083, 2002.

Principe, J., L. Wang y M. Motter. "Local dynamic modeling with self-organizing maps and applications to nonlinear system identification and control". *Proc. IEEE*, 86(11), 2240–2258, 1998.

Principe, J. C., H. Hsu y J. Kuo. "Analysis of short term memories for neural networks". *Advances in Neural Information Processing Systems*, 6, 1011–1018, 1993.

Principe, J. C., A. Rathie y J. M. Kuo. "Prediction of chaotic time series with neural networks and the issue of dynamic modeling". *Int. J. of Bifurcation and Chaos*, 2(4), 989–996, 1992.

Rao, S., S. Han y J. Principe, "Information theoretic vector quantization with fixed point updates". En *IJCNN 2007. International Joint Conference on Neural Networks.*, 1020–1024, 2007.

Rauber, A. y D. Merkl, "Automatic labeling of self-organizing maps: Making a treasure-map reveal its secrets". En *Proc. Pacific Asia Conf. on Knowledge Discovery and Data Mining*, 228–237, Springer Verlag, 1999.

Ritter, H., "Self-organizing maps in non-euclidean spaces". En E. Oja y S. Kaski (editores), *Kohonen Maps*, 97–108, 1999.

Ritter, H., T. Martinetz y K. Schulten. *Neural Computation and Self-Organizing Maps: An Introduction*. Addison-Wesley, Reading, MA, 1992.

Rosenstein, M., J. Collins y C. D. Luca. "A practical method for calculating largest Lyapunov exponents from small data sets". *Physica D*, 65, 117–134, 1993.

Roweis, S. T. y L. K. Saul. "Nonlinear dimensionality reduction by locally linear embedding". *Science*, 290, 2323–2326, 2000.

Sammon, Jr., J. W. "A non-linear mapping for data structure analysis". *IEEE Transactions on Computers*, 18, 401–409, 1969.

Schölkopf, B. y A. J. Smola (editores). *Learning with Kernels*. MIT Press, Cambridge, MA, 2001.

Schölkopf, B., A. J. Smola y K. Müller, "Kernel principal component analysis". En B. Schölkopf, C. J. C. Burges y A. J. Smola (editores), *Advances in Kernel Methods*, 327–352, MIT Press, 1998.

Simula, O. y J. Kangas, "Process monitoring and visualisation using Self-Organizing Maps". En *Neural Networks for Chemical Engineers*, 377–390, Elsevier Science B.V., 1995.

Simula, O., J. Vesanto, E. Alhoniemi y J. Hollmén, "Analysis and modeling of complex systems using the self-organizing map." En N. Kasabov y R. Kozma (editores), *Neuro-Fuzzy Techniques for Intelligent Information Systems.*, 3–22, Physica-Verlag, 1999.

Skogestad, S. y I. Postlethwaite. *Multivariable Feedback Control – Analysis and Design*. John Wiley & Sons, 1996.

Small, M. *Applied Nonlinear Time Series Analysis. Applications in physics, physiology and finance. Nonlinear Science Series A vol. 52*. World Scientific, 2005.

Strickert, M. y B. Hammer. "Merge SOM for temporal data". *Neurocomputing*, 64, 39–71, 2005.

Takens, F., "Detecting strange attractors in turbulence". En D. A. Rand y L.-S. Young (editores), *Dynamical Systems and Turbulence*, tomo 898 de *Lecture Notes in Mathematics*, 366–381, Springer, 1981.

Tenenbaum, J. B., V. de Silva y J. C. Langford. "A global geometric framework for nonlinear dimensionality reduction". *Science*, 290, 2319–2323, 2000.

Terra, M. H. y R. Tinós, "Fault detection and isolation for robotic systems using a multilayer perceptron and a radial basis function network". En *1998 IEEE International Conference on Systems, Man, and Cybernetics*, tomo 2, 1880–1885, IEEE, San Diego, CA, USA, 1998.

Toderick, L., T. Mohammed y M. Tabrizi. "A reservation and equipment management system for secure hands-on remote labs for information technology students". *Frontiers in Education, 2005. FIE '05. Proceedings 35th Annual Conference*, S3F–13–S3F–18, Oct. 2005.

Tryba, V. y K. Goser, "Self-Organizing Feature Maps for process control in chemistry". En T. Kohonen, K. Mäkisara, O. Simula y J. Kangas (editores), *Artificial Neural Networks*, 847–852, North-Holland, Amsterdam, Netherlands, 1991.

Tryba, V., S. Metzen y K. Goser, "Designing basic integrated circuits by self-organizing feature maps". En *Neuro-Nîmes '89. Int. Workshop on Neural Networks and their Applications*, 225–235, ARC; SEE, EC2, Nanterre, France, 1989.

Tukey, J. *Exploratory Data Analysis*. Addison-Wesley, Reading, MA, 1977.

Ultsch, A., "Self organized feature maps for monitoring and knowledge aquisition of a chemical process". En S. Gielen y B. Kappen (editores), *Proc. ICANN 93, International Conference on Artificial Neural Networks*, 864–867, Springer, Londres, Reino Unido, 1993.

Ultsch, A., "Maps for the visualization of high-dimensional data spaces". En *Proc. Workshop on Self Organizing Maps, WSOM'03*, 225–230, 2003a.

Ultsch, A., "U-Matrix: A tool to visualize clusters in high dimensional data". Informe Técnico 36, Dept. of Mathematics of Computer Science, University of Marburg, 2003b.

Ultsch, A., "UC: Self-organized clustering with emergent feature maps". En *LWA 2005, Lernen Wissensentdeckung Adaptivität*, 240–244, German Research Center for Artificial Intelligence (DFKI), Saarland University, Saarbrücken, Germany, 2005.

Ultsch, A. y H. P. Siemon, "Kohonen's Self Organizing Feature Maps for Exploratory Data Analysis". En *INNC Paris 90*, 305–308, Universitat Dortmund, 1990.

Van Hulle, M. M. "Joint Entropy Maximization in Kernel-Based Topographic Maps". *Neural Comp.*, 14(8), 1887–1906, 2002.

Vapola, M., O. Simula, T. Kohonen y P. Meriläinen, "Representation and identification of fault conditions of an anaesthesia system by means of the Self-Organizing Map". En M. Marinaro y P. G. Morasso (editores), *Proc. ICANN'94, International Conference on Artificial Neural Networks*, tomo I, 350–353, Springer, Londres, Reino Unido, 1994.

Venkatasubramanian, V., R. Rengaswamy y S. Kavuri. "A review of process fault detection and diagnosis part II: Qualitative models and search strategies." *Computers and Chemical Engineering*, 27(3), 313–326, Abril 2003a.

Venkatasubramanian, V., R. Rengaswamy, S. Kavuri y K. Yin. "A review of process fault detection and diagnosis part III: Process history based methods." *Computers and Chemical Engineering*, 27(3), 327–346, Abril 2003b.

Venna, J., *Dimensionality reduction for visual exploration of similarity structures.* Tesis Doctoral. Helsinki University of Technology, Espoo, Finland, 2007.

Venna, J. y S. Kaski. "Local multidimensional scaling." *Neural Networks*, 19(6-7), 889–899, Agosto 2006.

Vesanto, J., "Using the SOM and local models in time-series prediction". En *Proceedings of Workshop on Self-Organizing Maps (WSOM'97)*, 209–214, 1997.

Vesanto, J. "SOM-based data visualization methods." *Intelligent Data Analysis*, 3(2), 111–126, 1999.

Vesanto, J., "Neural network tool for data mining: SOM toolbox". En *Proc. Symposium on Tool Environments and Development Methods for Intelligent Systems (TOOLMET2000)*, 184–196, 2000.

Vesanto, J., *Data Exploration Process Based on the Self-Organizing Map.* Tesis Doctoral. Helsinki University of Technology, Espoo, Finland, 2002.

Vesanto, J. y E. Alhoniemi. "Clustering of the self-organizing map". *IEEE Transactions on Neural Networks*, 11(3), 586–600, Mayo 2000.

Vesanto, J., E. Alhoniemi, J. Himberg, K. Kiviluoto y J. Parviainen. "Self-organizing map for data mining in MATLAB: The SOM toolbox." *Simulation News Europe*, (25), 54, 1999.

Villmann, T. y J. C. Claussen. "Magnification control in self-organizing maps and neural gas." *Neural Computation*, 18(2), p446 – 469, Febrero 2006.

Villmann, T., R. Der, M. Herrmann y T. M. Martinetz. "Topology preservation in self-organizing feature maps: exact definition and measurement". *IEEE Transactions on Neural Networks*, 8(2), 256–266, 1997.

Voegtlin, T. "Recursive self-organizing maps". *Neural Networks*, 15(8), 979–991, 2002.

Walter, J., C. Nölker y H. Ritter, "The PSOM algorithm and applications". En *Proc. Symposion Neural Computation 2000 (Berlin)*, 758–764, 2000.

Walter, J., H. Ritter y K. Schulten, "Non-linear prediction with self-organizing maps". En *Proc. IJCNN-90, International Joint Conference on Neural Networks, San Diego*, tomo 1, 589–594, IEEE Service Center, Piscataway, NJ, 1990.

Webb, A. *Statistical Pattern Recognition.* Oxford University Press Inc., 1999.

Wiemer, J. C. "The Time-Organized Map Algorithm: Extending the Self-Organizing Map to Spatiotemporal Signals." *Neural Computation*, 15(5), 1143 – 1171, 2003.

Wolf, A., J. B. Swift, H. L. Swinney y J. A. Vastano. "Determining Lyapunov exponents from a time series". *Physica D*, 16, 285–317, 1985.

Xu, R. y D. Wunsch. "Survey of clustering algorithms". *Neural Networks, IEEE Transactions on*, 16(3), 645–678, Mayo 2005.

Yin, H. "Data visualisation and manifold mapping using ViSOM". *Neural Networks*, 15, 1005–1016, 2002.

Yin, H. "On multidimensional scaling and the embedding of self-organising maps". *Neural Networks*, 21(2-3), 160–169, 2008.

Ypma, A. y R. P. Duin, "Novelty detection using self-organizing maps." En N. Kasabov, R. Kozma, K. Ko, R. O'Shea, Coghill y T. Gedeon (editores), *Proceedings of International Conference on Neural Information Processing*, tomo 2, 1322–1325, Springer, 1997.

Yu, D., J. Gomm y D. Williams. "Sensor fault diagnosis in a chemical process via RBF neural networks". *Control Engineering Practice*, 7(1), 49–55, 1999.

Yu, D., M. Small, R. G. Harrison y C. Diks. "Efficient implementation of the Gaussian kernel algorithm in estimating invariants and noise level from noisy time series data". *Physical Review E*, 61(1), 3750–3756, 2000.